

**INSTITUTO POLITECNICO NACIONAL**  
**ESCUELA SUPERIOR DE COMPUTO**



# **GISDF: Sistema de Información Turística para el Distrito Federal**

## **T R A B A J O T E R M I N A L**

presentan

PAVEL BELIZARIO CORDOVA ANGEL  
JESUS IVAN FAJARDO GARCIA  
FELIPE FIGUEROA DEL PRADO  
VAN LOPEZ TOLEDO

**Director:** M en C. Carlos de la Cruz Sosa

**Asesores:** Dr. Alfredo Viniegra Islas, M en C. José María Ramos, M en C. Andrés Ortigoza Campos.

México, D.F. A 24 de Mayo de 2001

# ADVERTENCIA

“Este informe contiene información desarrollada por la Escuela Superior de Computo del Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convengan”

La aplicación no convenida exime a la escuela de su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte podrá obtenerse en la Subdirección Académica de la Escuela Superior de Computo del Instituto Politécnico Nacional, sita en Avenida Juan de Dios Batís s/n, teléfono: 729-6000 extensión 52012

# INDICE

<b>Capítulo</b>	<b>Tema</b>	<b>Página</b>
	<b>Advertencia</b>	2
	<b>Indice</b>	3
	<b>Agradecimientos</b>	5
	<b>Resumen</b>	10
	<b>Objetivo</b>	11
	<b>Introducción</b>	12
<b>1</b>	<b>S I G</b>	14
<b>1.1</b>	<b>Evolución Histórica del SIG</b>	14
<b>1.2</b>	<b>Definición de SIG</b>	16
<b>1.3</b>	<b>Características generales y limitaciones del SIG</b>	18
1.3.1	Características del SIG	18
1.3.2	Limitaciones del SIG	20
<b>1.4</b>	<b>Funcionamiento del SIG</b>	21
1.4.1	Entrada	21
1.4.2	Manipulación de Datos	21
1.4.3	Organización de Datos	22
<b>1.5</b>	<b>Arquitectura y Tipos de SIG</b>	24
1.5.1	Modelo Vectorial	24
1.5.2	Modelo Raster	25
<b>1.6</b>	<b>Tipos de Datos Geográficos</b>	27
<b>2</b>	<b>Diseño de GISDF</b>	30
<b>2.1</b>	<b>Descripción General del Sistema</b>	30
<b>2.2</b>	<b>Diagrama General del Sistema</b>	32
<b>2.3</b>	<b>Diagrama a Bloques del Sistema</b>	33
<b>2.4</b>	<b>Modulo de Consultas</b>	35
<b>2.5</b>	<b>Modulo de Conversión de datos</b>	38
<b>2.6</b>	<b>Diseño de la BD</b>	40
<b>3</b>	<b>Implementación de GISDF</b>	43
<b>3.1</b>	<b>Entrada de datos</b>	43
<b>3.2</b>	<b>Consultas a Base de Datos y a Mapas</b>	43
<b>3.3</b>	<b>Conversión de Datos</b>	51
<b>4</b>	<b>Manual de Usuario</b>	54

<b>4.1</b>	<b>Manual de Usuario (pagina)</b>	<b>54</b>
<b>4.2</b>	<b>Manual de Usuario (administración de Base de Datos)</b>	<b>61</b>
	<b>ANEXOS</b>	<b>67</b>
<b>1</b>	<b>Digitalización</b>	<b>68</b>
<b>2</b>	<b>Imágenes</b>	<b>72</b>
<b>3</b>	<b>Mapas</b>	<b>77</b>
<b>4</b>	<b>SQL Server</b>	<b>82</b>
<b>5</b>	<b>JDBC, Servlets</b>	<b>89</b>
<b>6</b>	<b>Mapas digitalizados</b>	<b>102</b>
<b>7</b>	<b>Referencias y Bibliografía</b>	<b>106</b>

## AGRADECIMIENTOS

# *Pánel Belizacio Córdova Angel*

*Dedicataria.*

*A mi mamá*

*A mis hermanos*

*A mi sobrina Danaé*

*A mis maestros*

## *Jesús Iván Fajardo García.*

*A Dios por darme la vida, darme fortaleza y acompañarme en todo momento.*

*A mis Padres, por haberme dado los valores necesarios para poder superarme; por su apoyo en todo momento y por su comprensión.*

*A ti "mamá" que supiste como orientarme y espero haberme convertido en una persona de la que te puedas sentir orgullosa; te doy las gracias por haberme apoyado durante tanto tiempo, por "partirmela" cuando fue necesario y por estar conmigo siempre que lo necesite.*

*A "andreas" te dedico esto que espero, me haya convertido en un buen ejemplo para ti, gracias por aguantarme y por ser mi mejor amigo*

*A Mis cuates y la familia, que me apoyaron y me aguantaron mi "carácter de perro".*

## *Felipe Figueroa del Prado*

*Le doy gracias a Dios porque sin el no hubiera sido posible esto.*

*A mis padres Felipe y Marcela por su apoyo incondicional para terminar mis estudios, siempre estuvieron ahí para apoyarme en todo.*

*A mis hermanos que estuvieron en todo momento para ayudarme en cuanto se pudo.*

*A mis familiares que se preocuparon de una u otra forma por mí, les agradezco su atención.*



## *Van López Toledo*

*A mi PADRE, quien a sabido ser mi guía en esta vida y me ha dado la fuerza suficiente de concluir esta meta en mi vida. Prof. Pedro López Escobar.*

*A mi MADRE, una gran mujer, por haberme traído al mundo y ser una fiel amiga y consejera en mi vida. Gracias por estar conmigo en todo momento. Profa. Luz María Toledo Castillejos*

*A Frini, por apoyarme de principio a fin en todo momento de mi carrera. Gracias por estar conmigo siempre.*

*A Thanya, por sus consejos y apoyo. Gracias por que se que puedo contar contigo siempre.*

*A Leida, porque siempre me a brindado su cariño y amor. Simplemente gracias.*

*A mis Tios "Cheche y Pancho", por su infinito apoyo en mi vida personal, de estudiante y como profesionista.*

*A mi familia, por apoyarme en todo momento, y sobre todo por ser eso una FAMILIA*

*A mis amigos por enseñarme el valor de la amistad*

*A toda esa gente que en un momento de mi vida estuvo conmigo y me sirvió de apoyo e inspiración para terminar esta etapa de mi vida.*

**Agradecemos a todos los que estuvieron involucrados en la realización de este Trabajo Terminal, A "Chema", "Dr. Vinie", Miguel y Marco. Que nos dieron posada en el CIC, nos orientaron.**

## Resumen.

El sistema de información turística está presentado a través de una página WEB, sin la necesidad de la descarga de ninguna aplicación por parte del usuario cliente; la parte principal del sistema es la ubicación espacial de los sitios turísticos de mayor interés con el adicional de incluir imágenes para los sitios más representativos así como imágenes panorámicas. El sistema contiene algunas de las características más importantes de un sistema de información geográfica, constituyendo así tan solo una aplicación de un GIS, es decir, el despliegue de mapas a petición del usuario; además relacionando la información espacial con las características de cada sitio.

La base esta implementada en un manejador de base de datos comercial como lo es SQL Server 7.x. Para las consultas desde la WEB se implemento por medio de Servlets, utilizando tecnología Java de tal manera que permite que la consulta al sistema puede ser con diferentes plataformas con la condición de que dichas plataformas tengan JRE (Java Run-time Enviroment); en estas plataformas están contempladas PC, UNIX y MAC.

El funcionamiento principal del sistema, esta basado en que una ves que el usurario ha descargada la página principal del sistema, tendrá que realizar una consulta o una petición especifica de información hacia el servidor donde esta incluida la base de datos. La información de los sitios turísticos está repartida en diferentes categorías, todo esto para que las consultas de la información a la base de datos puedan ser mas fácilmente realizadas por el usuario.

El despliegue de los mapas esta acompañado de botones, que implementan las funciones de manipulación de mapas; como son: zoom in, zoom out, centrado, cambio de escala, ancho del mapa, etc.

Es necesario mencionar que el formato en el que son almacenados los mapas son incompatibles con cualquier navegador estándar en el mercado, es decir, los navegadores solo permiten el despliegue de imágenes GIF o JPG; pero no permiten el despliegue de imágenes generadas por los formatos de los mapas en este caso el formato MapInfo. TAB, que es con el cual esta implementado el sistema.

# Objetivo

*Crear un sistema que ofrezca información actual de la localización espacial de sitios de interés turístico del Distrito Federal*

## Objetivos Particulares

- Llevar a la WEB un mapa de formato vectorial.
- Relacionar una base de datos con el mapa vectorial a través de las diferentes tecnologías WEB.
- Mostrar imágenes reales de los diversos sitios en la medida de lo posible.
- Actualización en tiempo real de la página al momento de modificar algún registro de la base de datos.

# INTRODUCCIÓN

Históricamente, la utilización de mapas como instrumentos para la toma de decisiones se sitúa en una época muy temprana. Un mapa es en definitiva un modelo gráfico del territorio en el que se representan determinadas características (morfológicas, temáticas, etc.) de acuerdo a una simbología. Y en la actualidad se disponen de técnicas muy elaboradas para su confección.

Las técnicas de cartografía digital, o cartografía asistida por ordenador, orientadas a la mecanización de la producción de mapas, si bien suponen una mejora considerable en el proceso de elaboración no suplen las limitaciones prácticas que presentan los mapas, respecto a la diversidad de temas y características que pueden representarse simultáneamente y el análisis que pueda hacerse en los mapas.

Es precisamente la capacidad de análisis de información espacial compleja el objetivo que se pretende con los Sistemas de Información Geográfica (SIG). Los Sistemas de Información Geográfica son sistemas informáticos capaces de realizar una gestión completa de datos geográficos asociados a mapas reales. Convirtiéndose los SIG en una herramienta muy potente para el manejo de grandes volúmenes de información.

Lo más característico de un SIG es su capacidad de analizar y generar nueva información mediante la manipulación y reelaboración de un conjunto previo de datos. Un SIG, de otro modo es bastante más que un sistema de diseño asistido por computadora.

Pero la mayoría de los software SIG comerciales existentes, sino es que todos tienen un alto costo en el mercado; además de que muchos de ellos tienen requerimientos de hardware en específico; sin contar que casi la totalidad de los SIG requieren de personal capacitado para su correcta operación. Representando así; un alto costo para quien desee implementar uno de estos SIG. Y que casi ninguno tiene como salida la presentación de los resultados vía Internet en un navegador común.

Por lo que, con la realización de este Trabajo Terminal se fijo como objetivo principal la implementación de un software, aplicación de un SIG que permitiera la recuperación de información de un mapa relacionado con una base de datos, permitiendo la localización de sitios en un mapa del DF así como mostrando la información relevante de cada uno de los sitios, y lo más importante que la salida de datos de este software sea a través de un navegador convencional WEB, es decir se implemento tan solo una aplicación de recuperación de información de. Dado que el campo de aplicación de un SIG es demasiado amplio. La aplicación implementada fue solamente enfocada a sitios turísticos del Distrito Federal; pero sentando las bases para que pueda no solo ser implementado para el DF y localización de sitios turísticos; si no para el sistema pueda ser adaptado a algún otro campo de aplicación del SIG.

# Capítulo 1. SIG

## 1.1 Evolución Histórica del SIG

---

El desarrollo de los Sistemas de Información Geográfica comienza en los años 60, Manteniéndose como un sistema de información original y sencillo en su concepción, a la vez que revolucionario: una simple herramienta a modo de híbrido entre una aplicación de dibujo y una base de datos, ambos de lo más tradicional. Las primeras realizaciones de este tipo de sistemas se registran en la segunda mitad de los años sesenta, impulsados principalmente por organizaciones con responsabilidades en la gestión de recursos con implantación territorial, ordenación del territorio, recursos naturales, censo, defensa, etc. Un ejemplo de estos sistemas es el Canadian Geographical Information System (CGIS) y que en Gran Bretaña se desarrolló la Unidad Experimental de Cartografía.

Una serie de circunstancias, entre las que la capacidad y coste de la tecnología digital disponible juegan un marcado papel, hacen que el desarrollo sea relativamente lento hasta la década de los ochenta en la que, especialmente al final de la misma, donde existe gran cantidad de oferta de productos comerciales y áreas de aplicación de esta tecnología.

Durante los años 70 aparecieron los primeros sistemas gráficos de este tipo, los cuales tenían como características principales:

- Concebidos para la delineación automática de mapas.
- Permitían realizar el mantenimiento de la cartografía existente.
- Eran sistemas experimentales desarrollados fundamentalmente en entornos universitarios.
- Seguían manteniéndose como campo de investigación

Estas aplicaciones estaban basadas en bases de datos muy sencillas que mantenían de forma separada, aunque relacionada, la información gráfica y la alfanumérica. Durante los años 80 se produjo una evolución de estos sistemas adoptando como características:

- Aparición de los primeros sistemas gráficos integrados, es decir la integración de gráficos de los mapas, con información alfanumérica almacenada en bases de datos.
- Explotación de herramientas CAD para el mantenimiento de la cartografía.
- Facilidad de mantenimiento gráfico de la información.

En 1982 se desarrollo el ARC/INFO, bajo la tutela de los ingenieros del Instituto de Investigaciones de Sistemas Ambientales (ESRI) en Red Lands, California, fundado en 1969. Otra institución pionera en el desarrollo de esto sistemas durante la década de los 80 fue el Laboratorio de Gráficos de la Universidad de Harvard.

Como ya se había mencionado en párrafos anteriores el desarrollo histórico de los SIG va en relación directa con el costo de los equipos de cómputo, ya que los principios matemáticos del tratamiento de matrices, vectores, componentes principales y análisis multivariado, fueron descritos por los matemáticos desde el siglo XVIII, pero es ahora con los medios actuales, cuando se hacen más rápidos y aplicables.

Por último podemos mencionar que en la década final del siglo XX casi todas las instituciones geográficas de prestigio han adquirido o desarrollado un SIG para su gestión, toma de decisiones o investigaciones.

En la década de los 80 instituciones como la Unión Geográfica Internacional (UGI) comenzó a organizar conferencias latinoamericanas sobre Sistemas de Información Geográfica; las cuales sirven como un marco propicio para favorecer las transferencias tecnológicas y el apoyo a las instituciones que se inician en estos métodos. Al igual que en Europa y Norteamérica se realizan los congresos AUTOCARTO donde numerosos profesores e investigadores de alto nivel muestran sus resultados más relevantes.

A lo largo de los años la tecnología de los SIG no solo ha incursionando en temas referentes a datos espaciales en el ámbito rural, también se han desarrollado aplicaciones de interés en los ambientes urbanos

Por último cabe mencionar que la evolución de los Sistemas de Información Geográfica durante los últimos años ha tenido gran auge o crecimiento en países como Canadá y USA; así como España Alemania y Rusia; contando ya estos países con cierta experiencia en la elaboración de SIG. Pero en países Latinoamericanos, es relativamente joven la elaboración de estos sistemas, contando ya la mayoría de las universidades y centros de investigación más importantes con departamentos de Geoprocesamiento o alguno relacionado como lo es el Laboratorio de Geoprocesamiento dentro del Instituto Politécnico Nacional (IPN) en el Centro de Investigación en Computo (CIC. Así como los departamentos relacionados de la Universidad Autónoma Metropolitana (UAM) y en la Universidad Nacional Autónoma de México.

## 1.2 Definición del SIG

---

Los SIG pueden definirse atendiendo: A su carácter de sistema de información, A su capacidad para la gestión de datos gráficos o A su concepción como base de datos, debido a esto no podemos dar una definición específica de un GIS. Entre las definiciones más extendidas podemos citar:

- Sistema informático diseñado con el fin de soportar la captura, análisis, manipulación y consulta de información referenciada espacialmente para la resolución de problemas de gestión.
- Cualquier sistema de información que permita: Capturar, almacenar y mostrar información basada en su localización espacial. Analizar los datos relacionados espacialmente como ayuda en la toma de decisiones. Selección e intercambio de información con modelos analíticos capaces de establecer diferentes alternativas.
- Tipo especializado de base de datos caracterizado por su capacidad de manejar datos geográficos que pueden ser representados gráficamente.
- Un conjunto de equipos informáticos, de programas, de datos geográficos y técnicos organizados para recoger, almacenar, actualizar, manipular, analizar y presentar eficientemente todas las formas de información georeferenciada.
- Un sistema de hardware, software y procedimientos diseñados para soportar la captura, gestión, manipulación, análisis, modelado y visualización de datos espacialmente-referenciados para resolver problemas complejos de planeamiento y gestión.

Desde un punto de vista práctico un Sistema de Información Geográfica es un sistema informático capaz de realizar una gestión completa de datos geográficos referenciados. Es decir que estos datos geográficos o mapas tienen unas coordenadas geográficas reales asociadas, las cuales nos permiten manejar y hacer análisis con datos reales como longitudes, perímetros o áreas. Todos estos datos alfanuméricos asociados a los mapas más los que queramos añadirle los gestiona una base de datos integrada con el SIG.

Es decir que, lo más característico de un SIG es su capacidad de analizar y generar nueva información mediante la manipulación y reelaboración de un conjunto previo de datos. Un SIG, de otro modo es bastante más que un sistema de diseño asistido por ordenador (CAD/CAM, *Computer Aided Design/Computer Aided Manufacturing*).



Dicho todo lo anterior consideramos que el profesor David Rhind (1989) ha pronunciado la mejor definición de un SIG que es:

*“Es un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados, para la solución de los problemas complejos del manejo y planeamiento territorial.”*

Muchos programas de computadora, como paquetes estadísticos o paquetes de dibujo, pueden dar sostén a datos geográficos o espaciales simples; pero esto no necesariamente los hace un SIG. Un verdadero SIG relaciona los datos espaciales con información particular de cada mapa. Por ejemplo para la localización de un sitio en un mapa; las líneas localizadas en el mapa no dicen mucho al usuario, así con un SIG el usuario podría hacer una consulta a un mapa y que la consulta proyectase no solo la localización del sitio en el mapa; si no información adicional de ese sitio; Como lugares cercanos, vías de comunicación para llegar a ese punto, tipo de terreno etc, construyendo nuevos datos de cierta forma a partir de los ya almacenados.

En pocas palabras un SIG no solo almacena mapas o imágenes; también almacena mas datos relacionados a los mapas en una base de datos. Estando muy relacionado el concepto de base da datos a un SIG; estableciendo con ello la diferencia entre un SIG y un programa cualquiera de diseño, o un programa que solo captura o modifica los mapas, que solo pueden producir una buena salida de datos grafica. Para que pueda un sistema ser llamado SIG tendría que tener la habilidad de asociar la información que figura en un mapa y crear nuevas relaciones que determinen información importante al usuario, información que pudiera utilizarse como por ejemplo en algunos de los siguientes campos:

- Construcción
- Impacto Ambiental
- Mejor localización de un lugar
- Marketing

Siendo estas tan solo algunas pocas aplicaciones que pudiera tener un SIG, las cuales se mencionaran con detenimiento más adelante.

## 1.3 Características generales y Limitaciones del SIG

---

### 1.3.1 Características del SIG

El SIG nos permite analizar los mapas estructurados en combinación con bases de datos asociadas. Se pueden interrogar para seleccionar los datos de interés, ver los resultados interactivamente eligiendo la simbología en función de los atributos asociados y producir cartografía de calidad. También se pueden preparar aplicaciones a medida, como un plan de control de incendios, de evaluación de impactos ambientales, un modelo que prevea la evolución de un incendio o de una inundación, aplicaciones verticales como un sistema de gestión municipal o una aplicación para una empresa eléctrica, etc.

Muchos programas de aplicaciones trabajan con operadores espaciales, sin embargo no se consideran un SIG pues no son capaces de realizar búsquedas y análisis espaciales, condición indispensable para que se considere como un SIG.

Los siguientes problemas constituyen preguntas típicas que tienen que resolver los SIG: Determinación del cierre óptimo de un embalse, Calcular el trazado menos costoso de una vía de comunicación, Evaluar los territorios más propicios para el cultivo. Problemas que jamás podrían ser resueltos por aplicaciones del tipo CAD/CAM, paquetes estadísticos y otras aplicaciones; que aunque apoyan a los SIG pero solo en calidad de herramientas del trabajo esencial.

Un sistema que sea llamado SIG deberá por lo menos responder a las siguientes preguntas y realizar las siguientes tareas:

<b>Preguntas</b>	<b>Tipos de Tareas</b>
¿Que es?	Inventario y Monitoreo
Donde es... Verdadero / Falso	Inventario
Que ha cambiado desde?	Inventario
Que patrón espacial existe?	Análisis Espacial
Que pasaría si?	Modelado

En la actualidad los SIG almacenan, analizan y realizan mapas de datos espaciales de todo tipo. También se tienen Sistemas de Información Geográfica Integrados, es decir que reciben de manera directa los datos provenientes de sensores remotos. Ejemplos de estos sistemas tenemos a ERDAS y DRAGON pero encontrando un problema estos, ya que requieren de la utilización de arquitecturas de computadoras superiores a las microcomputadoras, aunque esto representa un alto costo. En los próximos años se espera

un crecimiento importante en el uso de Sistemas de Información Geográfica Integrados. En general los SIG desarrollados hasta el momento se componen de módulos de Entrada y Almacenamiento, Análisis.

Características mínimas de un SIG:

### **Mantenimiento de los Datos**

Otra forma de ver la funcionalidad del GIS es la de disponer de información actualizada en todo momento, dependiendo del tipo de aplicación GIS de que se trate. Para ello se deberá realizar un mantenimiento de la información residente en la base de datos a lo largo de la explotación del mismo. Efectuando un mantenimiento de la información alfanumérica como de la información gráfica.

### **Georeferenciación**

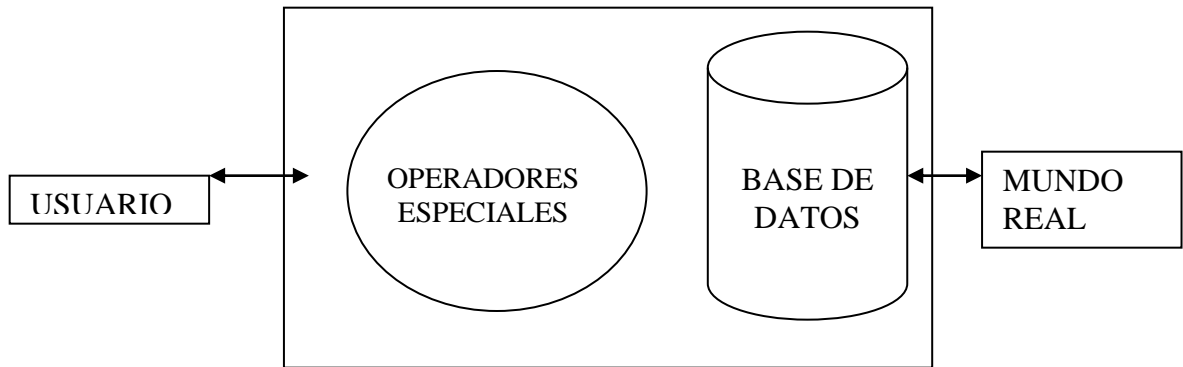
Encontrando mas características que hacen funcionales a un GIS encontramos la Georeferenciación de los datos la cual se define como aquel proceso mediante el cual se identifica una posición en la superficie terrestre. Existen dos tipos de georeferenciación:

- Georeferenciación Directa: Se basa en el uso de un sistema de coordenadas establecido para un determinado sistema de proyección.
- Georeferenciación Indirecta: Su fundamento es asociar al elemento que se representa una clave o índice, normalmente con significado administrativo (dirección, código postal, etc.), que puede ser usada para la determinación de una posición, naturalmente con una precisión no siempre equivalente a la obtenida con Georeferenciación directa.

### **Modelado Cartográfico**

Creación de nuevos mapas a partir de mapas existentes: Combinado atributos del terreno como pendiente, vegetación, tipo de suelo, etc. Mediante un modelo matemático se pueden crear nuevas variables, como un índice de erosionabilidad, de riesgo de incendios, etc.

En 1987 Guevara planteó que un SIG <sup>es</sup> ~~es~~ como una interfase entre el mundo real y el usuario, como se describe en la siguiente figura:



### 1.3.2 Limitaciones del SIG

Un SIG es un gran sistema informático cuya implantación en una organización es siempre gradual y costosa. Se requiere siempre la adecuación del sistema al trabajo requerido, mediante programación (frecuentemente realizada por el autor del SIG) y recopilación de los datos necesarios (suministrados por otras organizaciones o introducidos por el cliente), o incluso por la compañía que suministra o encargada de la programación siendo esta una gran labor; en la adquisición de datos; así como su integración con la información requerida en un área específica de aplicación.

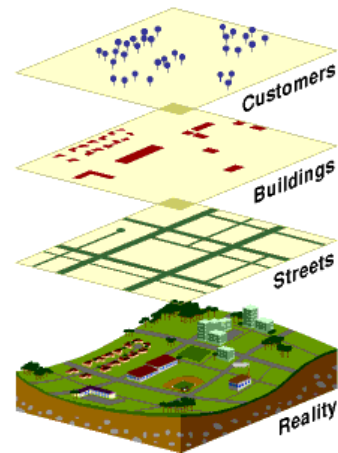
Los Sistemas de Información Geográfica representan un campo de desarrollo, donde se interceptan muchas disciplinas, entre ellas, Cartografía, Computación, Fotogrametría, Teledetección, Estadística y otras disciplinas relacionadas con el manejo y análisis de datos territorialmente codificados.

Para su explotación es necesaria la concurrencia de programadores junto con los profesionales del área de estudio en cuestión. Autores de conceptos primarios del SIG, sostenían que: no son pues, herramientas de usuario final, es decir, para profesionales no conocedores del SIG: un SIG es una herramienta propia de una organización, no una herramienta personal. Pero ahora se está tratando de llevar esta tecnología a todo tipo de usuarios sean o no conocedores de la tecnología del GIS; aumentando así la dificultad del sistema, en cuanto a su programación, integración de los datos; para que sea entendible lo más fácilmente por el usuario.

## 1.4 Funcionamiento del SIG

---

Un SIG almacena información a cerca del mundo como una colección de capas temáticas, que pueden ser ligadas entre ellas por los datos geográficos. Pero este concepto rápido y versátil, ha sido invaluable para resolver problemas del mundo real. Un SIG deberá realiza todas o algunas de las siguientes tareas:



### 1.4.1 Entrada

Antes de que los datos geográficos puedan ser usados en un SIG, los datos tienen que ser convertidos a un formato digital conveniente, el proceso de convertir los datos de los mapas en archivos de computadora se llama digitalización. Aunque este paso de la digitalización puede ser omitido debido a la existencia de empresas que pueden proporcionar los datos ya digitalizados.

Las tecnologías modernas de SIG pueden también automatizar el proceso para proyectos muy grandes usando scanner; pero trabajos pequeños pueden requerir de la digitalización manual (usando una tableta digitalizadora). Ahora existen muchos tipos de datos geográficos en formatos compatibles de SIG. Estos datos pueden ser obtenidos de proveedores y cargados directamente en un SIG.

### 1.4.2 Manipulación de Datos

Es la forma como los datos requeridos por un proyecto de SIG se necesitaran ser transformados o manipulados en alguna manera para hacerlos compatibles con la aplicación a desarrollar. Por ejemplo la información geográfica esta disponible en diferentes escalas (detallando calles, algunas solo colonias, códigos postales en un nivel de región). Antes de que esta información pueda ser integrada, deberá ser transformada a la misma escala o nivel. Esta podría ser una transformación temporal para propósitos de despliegue de la información en imágenes; o para un análisis permanente que sea requerido. La tecnología SIG ofrece en la actualidad múltiples herramientas para manipular los datos espaciales y así mismo eliminar datos innecesarios.

### 1.4.3 Organización de Datos

Para proyectos pequeños será suficiente con almacenar la información geográfica como simples archivos. Pero cuando el volumen de datos se incrementa y llega a ser demasiado grande; además del número de usuarios de los datos también llega a ser mayor a unos cuantos. Es recomendable el uso de un manejador de base de datos (DBMS) que ayude al almacenamiento, organización y manejo de los datos.

Con los diferentes tipos de DBMS que existen, para un SIG el más útil son los que utilizan el modelo relacional; Así los datos se almacenan como una colección de tablas. Los campos comunes en las diferentes tablas son usados para ligarlas entre ellas. Este tipo de diseño ha sido el más usado en cuanto a los GIS debido a la flexibilidad que presentan.

Attributes of California Counties				
FIPS	County	Pop. 1990	Pop. 2000	Pop. 2010
0001	Alameda	1559	1	Pacific
0003	Alpine	5384	3	Pacific
0005	Amador	1490	5	Pacific
0007	Butte	1953	7	Pacific
0009	Calaveras	1488	8	Pacific
0011	Colusa	1159	11	Pacific
0013	Colusa	1802	13	Pacific
0015	Contra Costa	1472	13	Pacific
0017	Contra Costa	156	15	Pacific
0019	Del Norte	1225	17	Pacific
0021	El Dorado	1783	18	Pacific

income.dbf		
FIPS	County	Pop. 2000
0001	Alameda	12488
0003	Alpine	11039
0005	Amador	9395
0007	Butte	9647
0009	Calaveras	9554
0011	Colusa	8791
0013	Contra Costa	14863
0015	Contra Costa	14593
0017	Del Norte	7514
0019	El Dorado	10817
0021	El Dorado	9236

Un aspecto importante a considerar en el desarrollo de todo SIG es la posibilidad de contar con un módulo de programas y sistemas que permitan el análisis de la información almacenada de modo que esta no sea solo una colección de datos de carácter descriptivo sino que, por el contrario, producto de toda una serie de manejos y análisis variados, resulte en respuestas concretas a problemas planteados.

### Consulta y Análisis

Una vez que ya se tiene un SIG funcional conteniendo su información geográfica; ahora se puede comenzar a hacer preguntas dependiendo del tipo de aplicación de que se trate, como:

- ¿A quien pertenece la granja de aquella esquina?
- ¿Qué tan lejos están dos lugares, uno del otro?
- ¿Dónde esta la tierra situada para uso industrial?

Y preguntas aun más específicas como:

- ¿Dónde están todos los sitios más propicios para construir nuevas casas?
- ¿Cuál es el suelo dominante de los bosques de roble?

- ¿Si construyo una carretera aquí, como se vera afectado el tráfico?

Una consulta realizada por un SIG puede ser un acceso individual o directo a los datos, un acceso selectivo mediante una variedad de especificaciones o condiciones de tipo geométrico, simbólico o lógico, una reclasificación de atributos o una modelación o cualquier procedimiento analítico determinado por el usuario. Varios SIG denominan a esa parte funcional del sistema como subsistema de recuperación y análisis o simplemente subsistema de búsquedas.

Los SIG pueden proveer consultas incluso con un solo click; hacer análisis mas sofisticados para los directores y analistas de la información, por ejemplo. Los SIG modernos proveen muchas herramientas de análisis pero dos son importantes: Análisis de proximidad, Sobre posición de capas La primera se refiere a la consulta de que tan cerca esta un lugar de otro, y el segundo se refiere a la integración de diferentes datos geográficos para dar origen a nuevos datos geográficos.

## 1.5 Arquitectura y Tipos de SIG

---

Solo podría hacerse una división o clasificación de las Arquitecturas o Tipos de SIG, utilizando las estructuras de datos por medio de las cuales pueden ser expresados con una precisa referencia espacial, provocando diferentes interrelaciones entre las unidades elementales de información,  $x$  &  $y$  que se usan en un SIG.

El diseño de estas interrelaciones es lo que da las diferentes opciones entre las Arquitecturas o tipos de SIG, cuya elección entre una de ellas es trascendental pues condiciona el futuro manejo de la información. Por otro lado algunos de los tipos son los usados por los modelos digitales del terreno, que en la actualidad y en su mayoría son hechos en 2d y 3d. Se podría decir que históricamente los tipos de SIG se han dividido básicamente en dos grupos en función de la concepción de la representación de los datos los cuales son: **Vectorial** y **Raster**.

### 1.5.1 Modelo Vectorial

Los **vectoriales** están basados en entidades (básicamente puntos y líneas) definidas por sus coordenadas. En los modelos vectoriales los atributos del terreno se representan mediante líneas (vectores) y puntos acotados. Los puntos se definen mediante un par de valores de coordenadas y las líneas, por tanto, por medio de un vector de pares de coordenadas. En la práctica, de los modelos vectoriales se ha reducido entre otros a dos potenciales métodos de los cuales se puede desarrollar un SIG:

#### **Isopletas o Contornos:**

La estructura básica es el vector, compuesto por un conjunto de pares de coordenadas  $(x,y)$  que describe la trayectoria de líneas isométricas coincidiendo por tanto con curvas de nivel o isohipsas del mapa topográfico convencional. El número de elementos de cada vector es variable y la reducción de este a un único elemento permite incorporar cotas puntuales sin introducir incoherencias estructurales.

#### **Redes de triángulos irregulares (TIN).**

Una estructura de datos cada vez más usada es la que se compone de un conjunto de triángulos irregulares y que se identifica por las siglas de su denominación inglesa *triangulated irregular network* (TIN). Estos están internamente organizados en función de su vecindad mediante un conjunto de información bastante posible un manejo relativamente ágil y eficaz frente a alternativas menos estructuradas.



### **1.5.2 Modelo Raster.**

En los de tipo **raster**, los datos se interpretan como el valor medio de unidades elementales de superficie no nula, que representan el terreno con una distribución regular, sin solapamiento y con recubrimiento total del área representada. Las matrices que son una característica más del formato Raster, pueden ser consideradas como un conjunto de puntos acotados sin asumir necesariamente que el atributo, de estos se asigna a una celda cuadrada de altitud uniforme.

#### **Matrices regulares**

La estructura matricial es el resultado de superponer una retícula sobre el terreno y extraer la altitud media de cada celda (aunque habitualmente se utiliza un valor puntual), asociado a cada nodo de la retícula o de punto medio de la celda, con lo que esencialmente se construye un modelo vectorial de puntos. La retícula puede adoptar formas variantes pero la más utilizada es una red regular de malla cuadrada de filas y columnas. En esta estructura la localización espacial de cada dato está implícitamente ligada por su situación en la matriz, una vez definidos su origen y el intervalo entre filas y columnas. Las matrices de altitudes suelen ser generadas por interpolación a partir de un modelo previo de contornos o por métodos fotogramétricos.

#### **Matrices de resolución variable**

El interés de las matrices de resolución variable, reside en la posibilidad de solucionar el principal problema de las matrices regulares (su resolución espacial prefijada), manteniendo en principio sus principales ventajas: la sencillez conceptual y operacional. En este tipo de matrices los elementos pueden ser, datos elementales (como en las matrices regulares), submatrices con un nivel de resolución diferente. La estructura final es un árbol jerárquico y dinámico de submatrices con una profundidad en principio arbitraria y cuya resolución espacial se duplica en cada nivel.

A continuación se incluye una tabla donde se resumen los principales Tipos o Arquitecturas de GIS:

**Vectoriales**

**Contornos**

**Perfiles**

**Triángulos**

**Secuencial:** Las líneas se almacenan como

una cadena de cotas

**analítica:** Las líneas se almacenan como

segmentos de Bezier, polinómicos, etc.

Cadenas paralelas de cotas en línea con altitud variable

Red de Triángulos Irregulares

**Raster**

**Matrices**

**Polígonos**

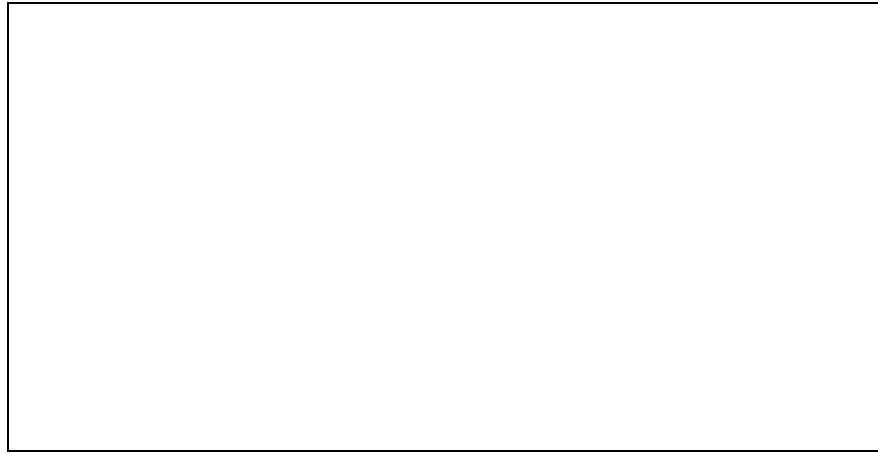
**Regulares:** Cotas sobre una malla

cuadrada de filas y columnas equidistantes

**Escalables:** Cotas sobre submatrices

jerárquicas y de resolución variable

Cotas asignadas a teselas poligonales regulares (triángulos o hexágonos)



## 1.6 Tipos de Datos Geográficos

---

Los datos que la ciencia utiliza son los que se refieren a hechos objetivos y en el caso específico de la geografía, históricamente han estado relacionados con la descripción de los espacios estudiados. Entonces los datos geográficos son entidades espacio-temporales que describen o cuantifican la distribución, el estado y los vínculos de los distintos fenómenos naturales y sociales. Los datos geográficos se expresan gráficamente en mapas, y se representan por signos convencionales especiales denominados signos cartográficos.

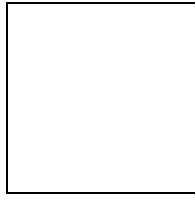
Los SIG son diseñados por lo tanto en especial para tratar simultáneamente con datos espaciales e información descriptiva de datos no gráficos (información estadística) relacionada a dichos datos espaciales. En un SIG los datos se describen por sus atributos temáticos, por su localización geográfica y su configuración espacial. Las bases de datos de un SIG se consideran como su espina dorsal y de su estructura dependerá la validez del resultado final de la consulta a un SIG. Es decir de la calidad de los datos introducidos a la Base de Datos del SIG dependerá la calidad de los resultados.

La clasificación de los datos geográficos de acuerdo a su topología tomando en cuenta solo el número de sus dimensiones es:

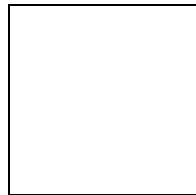
- 1) **Datos Puntuales:** Representan fenómenos que ocurren en localizaciones concretas del espacio geográfico cuya extensión sea nula o insignificante. Con un cambio de escala significativo este tipo de datos no son estables.



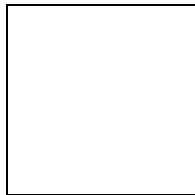
- 2) **Datos Lineales:** Son entidades representadas por líneas en los mapas, ejemplo: carreteras.



- 3) **Datos Superficiales:** Recogen información arealmente, ejemplo: Tipos de suelos, masas forestales



- 4) **Datos Volumétricos:** Para la representación de objetos tridimensionales, aunque la mayoría de los sistemas vigentes no pueden representar objetos tridimensionales de forma consistente; entonces se representa el volumen por la superficie que lo representa.



# Capítulo 2. Diseño de GISDF

## 2.1 Descripción General del Sistema

---

El sistema de información turística es presentado a través de una página WEB, sin la necesidad de la descarga e instalación de ninguna aplicación por parte del usuario o cliente; la parte fundamental del sistema es la ubicación espacial de los sitios turísticos de mayor interés para el área del Distrito Federal con el adicional de incluir imágenes, de los sitios más representativos; así como imágenes panorámicas de algunos de esos sitios.

El sistema esta basado en las características esenciales de un Sistema de Información Geográfica, constituyendo así una aplicación de un SIG. Es decir el sistema relaciona la información espacial contenida en los mapas con información específica de cada sitio turístico contenida en una base de datos relacional.

El funcionamiento principal del sistema, se esta basado en la idea de que el usuario una vez accesado a la pagina principal del sistema, tendrá que realizar una consulta o una petición específica de información hacia el servidor donde esta instalada la base de datos. Internamente en el servidor se realizará una petición a un applet para que realice la búsqueda de la información solicitada por el usuario en el mapa y se la presente al usuario. La información de los sitios turísticos será repartida en diferentes categorías, todo esto para que las consultas de información a la base de datos puedan ser más fácilmente realizadas por el usuario.

### **Base de Datos**

La base de datos esta implementada con un manejador de base de datos que permite la relación de información multimedia, datos espaciales y atributos característicos de cada sitio turístico. El manejador de Base de Datos utilizado fue SQL Server 7.0 debido a que ofrece un rendimiento estable como servidor de base de datos; añadiendo a esto los recursos de hardware que solicita son mínimos en comparación a otros manejadores de base de datos relacionales que también permiten la inclusión de datos espaciales.

### **Consultas a la Base de Datos y al Mapa**

Como ya se había mencionado anteriormente el sistema es accesible desde una pagina web desde la cual se pueden realizar consultas al sistema de búsqueda de sitios turísticos y si así lo desea el usuario la ubicación espacial del sitio. Entonces las consultas desde la pagina al sistema están implementadas por medio de servlets y applets; ambos utilizando tecnología JAVA lo cual permite que el sistema pueda ser instalado y consultado en diversas plataformas con la única condición que dichas plataformas tengan instalado el JVM (Java

Virtual Machine); como ejemplo de las diversas plataformas tenemos: Windows / Unix / Mac / Solaris.

## **Mapas**

En cuanto a la información georeferenciada es decir los mapas están almacenados en 4 diferentes archivos correspondiendo cada uno a una parte del Distrito Federal. Los archivos son guardados en el formato de archivos de MapInfo, información específica sobre este formato es incluida en los anexos de este mismo documento. En forma general cada archivo constituye una capa diferente en el mapa por sí solo.

La información contenida en la base de datos y los mapas de este sistema incluye sitios turísticos como son:

- Escuelas y Centros Universitarios
- Cines
- Teatros
- Bares
- Antros
- Casas de Cultura y Museos
- Hoteles
- Restaurantes
- Deportivos
- Tiendas o plazas comerciales
- Servicios
- Monumentos históricos.

## **Consultas:**

Existen diferentes tipos de consultas

- Consultas específicas en base a las categorías antes mencionadas.
- Consultas por zona geográfica
- Consultas por una palabra que ingrese el usuario para la consultas

Debido a que las consultas son de diferente tipo y pueden ser de diferentes categorías los resultados de las búsquedas se presentan en código HTML dentro de la misma página WEB del sistema. Con ligas de cada uno de los sitios arrojados por la búsqueda para que se pueda ser presentada la búsqueda del sitio en el mapa.

El despliegue de los mapas se acompañará de botones, que implementarán las funciones de manipulación de mapas; como son: Zoom In , Zoom Out, cambio de escala, ancho del mapa, etc.

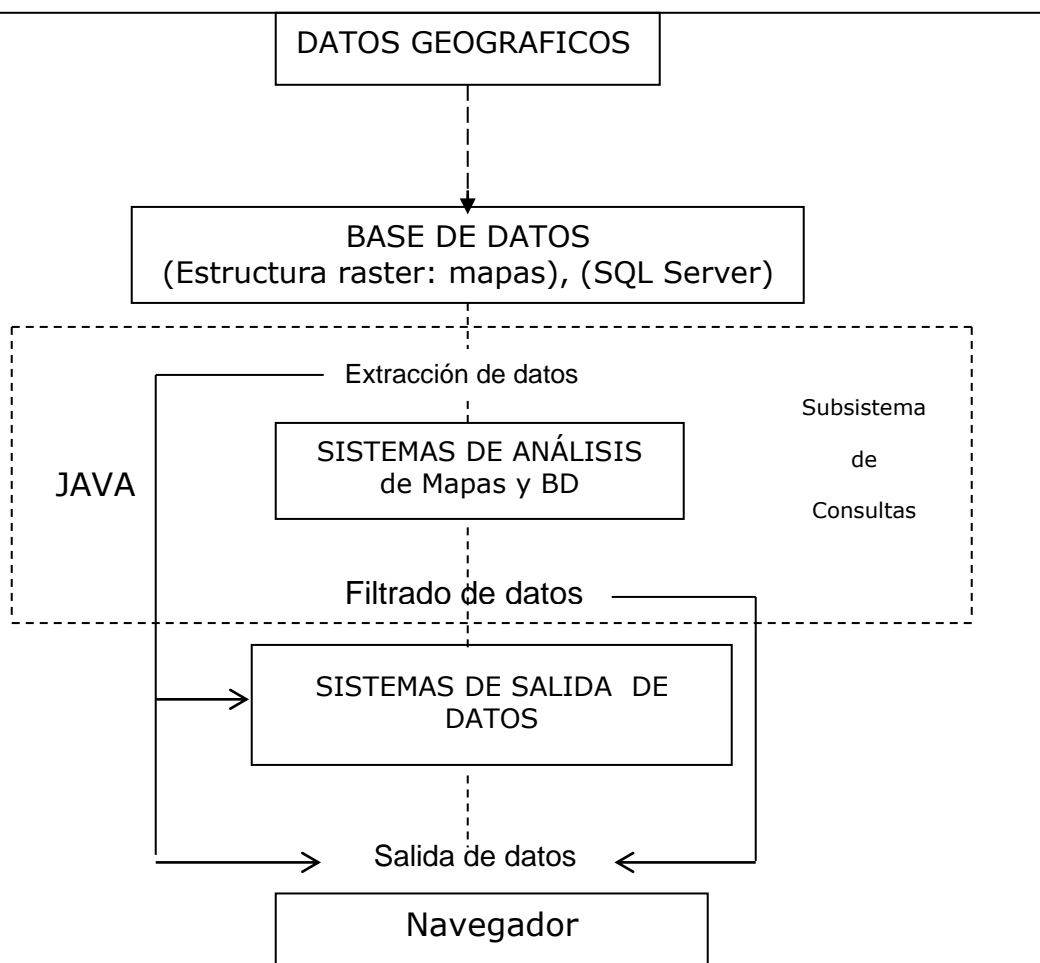
Es necesario mencionar que el formato en el que son almacenados los mapas son incompatibles con cualquiera de los navegadores comerciales. Es decir los navegadores comerciales solo permiten el despliegue de imágenes GIF, JPG, BMP, TIFF, PNG; pero no permiten el despliegue de imágenes generadas por los formatos de los mapas como el formato de MapInfo que será utilizado en este sistema.



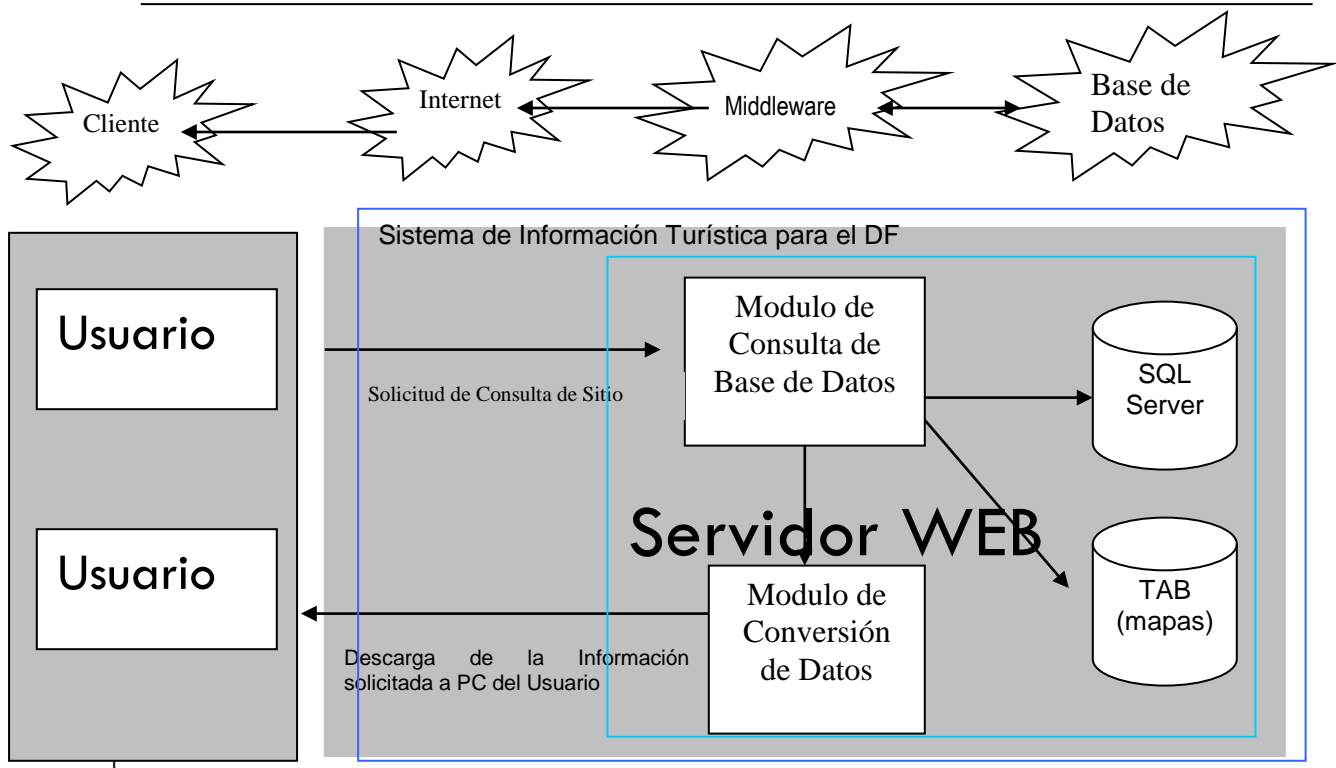
### Consultas a Mapas:

Entonces para el despliegue de estas imágenes, se hace necesario la conversión a un formato que pueda ser abierto por cualquier navegador de las diferentes plataformas. Esto se realizará haciendo un proceso similar al de una consulta, se extrae la información solicitada por el usuario y esta información se convertirá a un formato que para este sistema, podrá ser o GIF y JPG, para lo cual se usara MapXtreme for Java; para la consulta a mapas y la conversión de imágenes del formato de MapInfo a GIF o JPEG.

## 2.1 Diagrama General del Sistema



## 2.2 Diagrama a Bloques del Sistema



### Análisis del Diagrama a Bloques del Sistema:

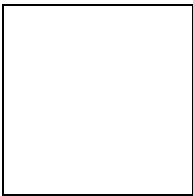
1. Analizando el sistema que vamos a desarrollar: Sistema de Información Turística para el DF, esta representado por el diagrama anterior desde un punto de vista muy general. Como ya se había mencionado anteriormente en este documento, el sistema presenta las siguientes características:
2. El Usuario desde un navegador WEB, independiente de la plataforma, pero compatible con JAVA. Descarga la pagina de GISDF en su PC
3. El Usuario elabora una consulta mediante un sencillo sistema de menú's integrados en la misma pagina.
4. El Usuario selecciona alguno de los sitios mostrados como resultado de la consulta anterior.
5. Se elabora una nueva consulta, en base al punto anterior.

6. El proceso de las Consultas pasa a través de un modulo que captura las consultas
7. El modulo de Consultas, se encarga de la conexión con el servidor y la base de datos, la extracción de la información, de ahí pasa al Modulo de Conversión de datos.
8. El Modulo de Conversión de Datos se encarga, de una vez que se recibió la información del Modulo de Consultas, es necesario la transformación de los datos aun formato que puedan ser presentados en un navegador WEB.
9. La transformación de datos hecha por el Modulo de Consultas, integra los datos y ya pueden ser presentados al Usuario.
10. Se descarga la información integrada al navegador del Cliente, presentado así los resultados de la Consulta elaborada por el Cliente.

Este sistema puede ser, presentado de una manera muy fácil con tan solo la integración de dos módulos en el sistema.- Así se hace una representación sencilla del sistema. Pero más adelante se presentan con más detalle y explicación estos módulos. Por otro cada uno de los módulos presentados en este diagrama presenta su grado de dificultad, obedeciendo a la integración de datos de diferente naturaleza.

Es decir primero que nada la extracción de la información de la base de datos, donde estará almacenada la información de los Sitios Turísticos, después se encuentra el problema de la extracción de la información geográfica especifica solicitada por el usuario en la consulta elaborada y su posterior conversión para que pueda ser observada desde un navegador, la información de los sitios turísticos muchas veces ira acompañada de Imágenes, Sonido, Audio, Imágenes Multimedia. Por ultimo la integración de toda esta información en Código HTML para elaborar algo así como un reporte donde se encuentra la integración de toda la información para un sitio turístico en especifico.

## 2.3 Modulo de Consultas



### Análisis del modulo de Consultas:

Este diagrama representa el funcionamiento del sistema o modulo de consultas, tanto a la Base de Datos de los Sitios Turísticos, como; también contempla la consulta de la información geográfica al mapa.

**Obtener Consulta del Cliente.**- Este proceso se refiere, o tiene contemplado que a partir de una Pagina WEB el usuario pueda realizar o elaborar una consulta; es decir, en la pagina se desplegaran las diferentes opciones de búsqueda y las diferentes opciones de consulta que se podrán hacer en el sistema para que pueda realizarse la consulta de información de un sitio en específico

Entre las opciones de búsqueda encontramos:

- Por Zona Geográfica, pudiendo encontrar diferentes zonas en el área del Distrito Federal.
- Por Delegación Política

- Por Tipo de Sitio, es decir podrá ser: Hoteles, Restaurantes, Bares, Cantinas, Cafés y Cafeterías, Centros Nocturnos, Discotecas, Parques Bosques y Parques Recreativos.
- Por Medio de Transporte
- Por Nombre, Si el usuario desea buscar algún sitio específico el usuario podrá buscarlo tan solo por su nombre.
- Por Tipo de servicio

Por último, se menciona que de ser necesario, se incluirán más opciones de búsqueda del sitio, de manera que el sistema sea lo más eficiente y fácil de utilizar por incluso el usuario menos experimentado.

### **Contactar al Servidor WEB**

Este es un proceso que casi se realiza de forma automática, es necesario contactar al servidor WEB; pero debido a que el sistema será implementado con servlets de tecnología JAVA es necesario que para poder procesar las consultas tanto a la Base de Datos como al mapa, se contacte al servidor en el puerto adecuado para ello, o designado previamente por el administrador del sistema para el soporte de estos servlets.

### **Conectar a BD y ejecutar consulta**

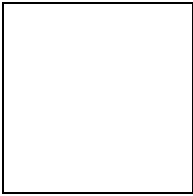
Este proceso es uno de los de mayor importancia para este módulo ya que se conecta a la Base de Datos de un determinado RDBMS y procesa la consulta solicitada por el usuario y de la cual ya se ha hablado como se obtiene, anteriormente. Dejando al final de este proceso o como salida, la consulta de la base de datos, obteniendo información del sitio, como es algunas características, información específica como su dirección, teléfono, desde este mismo proceso se obtendrán la información multimedia que pudiera contener cada sitio turístico. Al final se entrega toda esta información a otro módulo que pueda convertir e integrar los datos para que puedan ser observadas en una página WEB.

### **Conectar a Mapa y ejecutar consulta**

Aquí se realiza casi el mismo proceso que el anterior, se accesa al mapa que contiene la información de la localización geográfica de cada sitio, así como las imágenes de mapa de los sitios, pero el resultado o como salida de este proceso se tiene la imagen del sitio, información de la localización geográfica; pero este tipo de imagen obtenida en este proceso; aun no es compatible para su despliegue en una página de código HTML que pueda ser observada desde un navegador. Para esto se entrega la información a otro módulo que pueda hacer una conversión de los datos.

## 2.4 Modulo de Conversión de Datos

---



### **Análisis del modulo de conversión de datos.**

Este modulo esta diseñado para que reciba la información de la Base de Datos de la información del mapa, que pueda transformar en código valido que pueda ser interpretado en un navegador y por otro lado integra la información del mapa, con el código HTML antes mencionado.

#### **Transformación de la BD.**

Una ves, que logro hacerse la consulta en el módulo anterior, en este modulo se hace que la información obtenida por la Base de Datos ahora se tiene que insertar con código HTML, todo esto para que se pueda desplegar en un navegador, sin ningún problema; por otro lado en este proceso debe también crearse links hacia archivos multimedia o que la información multimedia se integra directamente con el Código HTML.

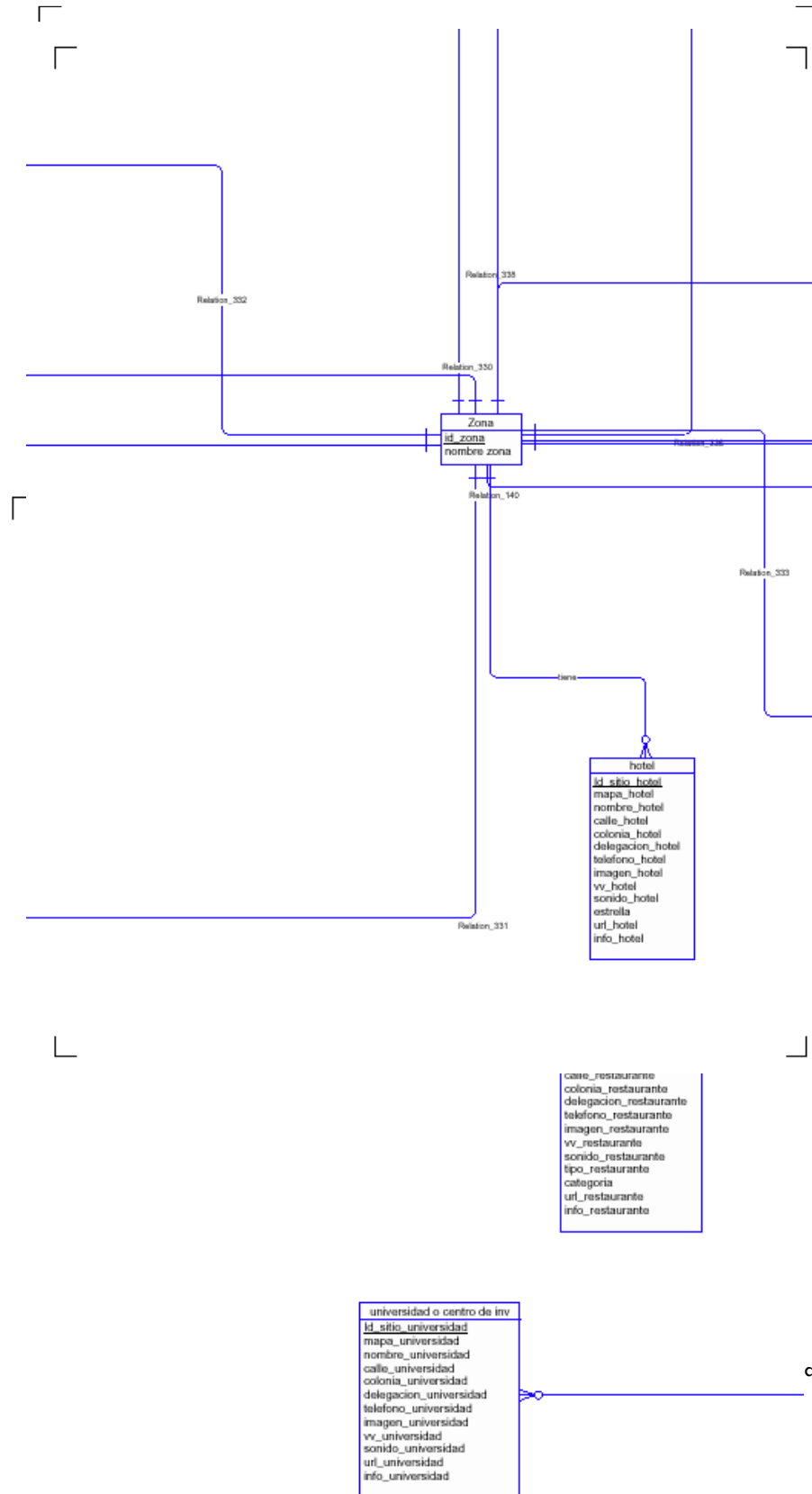
#### **Transformación del mapa a Imagen.**

En este proceso se recibe la información procesada anteriormente por la consulta hecha al mapa, pero esta no tiene un formato compatible con los navegadores. Por lo que se tiene que hacer una transformación a un formato que pueda se pueda integrar en una pagina WEB para que se observe, sin la necesidad de la descarga de un plug in.

#### **Integración de la Información con la Imagen.**

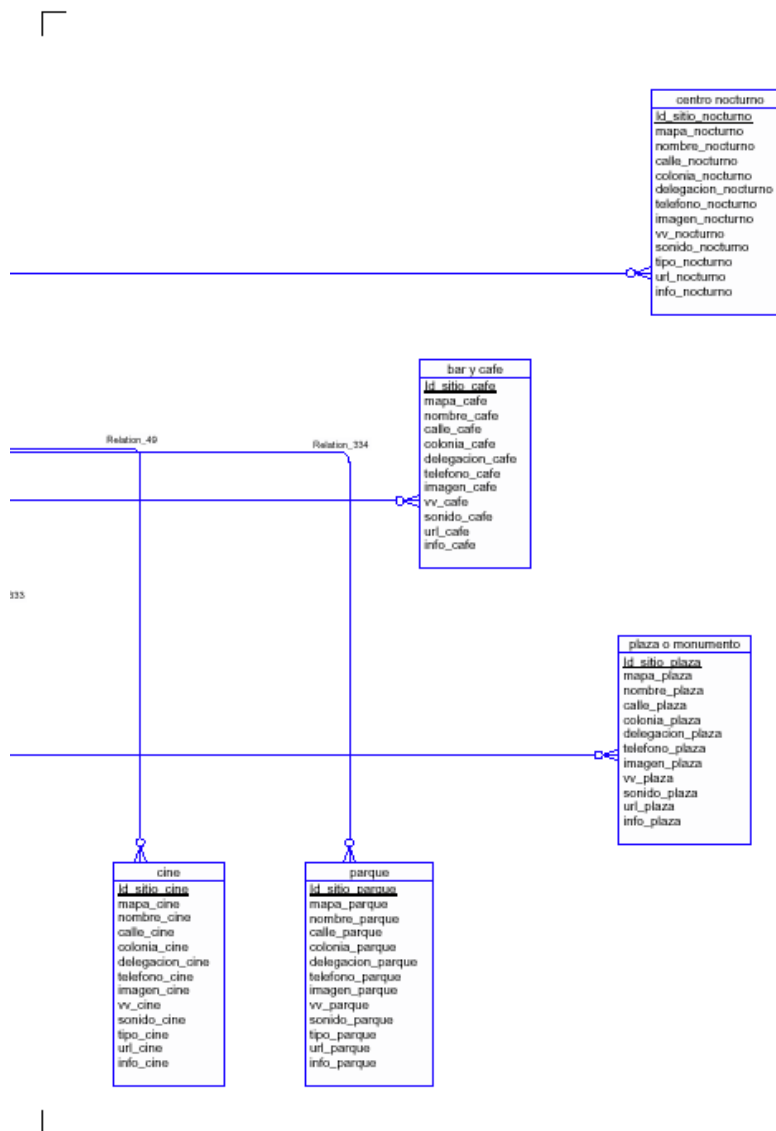
Este proceso permite que se integre la información ya transformada por los procesos anteriores en una sola pagina de tal manera que pueda ser más fácilmente apreciada por el usuario.

## 2.5 Diseño de la Base de Datos









# Capitulo 3. Implementación de GISDF

## 3.1 Entrada de datos

Como se muestra en el Diagrama General en la sección 2.8; el funcionamiento del sistema comienza con la entrada de los datos, es decir con la entrada de datos a la base de datos y al mapa. La parte de entrada de datos en lo que se refiere al mapa; se realiza introduciendo los datos por medio de una tableta digitalizadora, siendo esta una de las formas más elementales de entrada de datos. Como fuente de información se utilizó un mapa del Distrito Federal ya existente. Para la realización de esta etapa del sistema se utilizó AutoCAD 14.

El AutoCAD 14 utiliza el método vectorial para el almacenamiento de la información geográfica, para que sea posible el posterior análisis de estos mapas. Se guardan en formato vectorial debido a la sencillez que representa el tratamiento de la información de esta forma, pero por otro lado representa ciertos riesgos de imprecisión en la adquisición de los datos.

Utilizando AutoCAD 14 fueron digitalizados los mapas en 4 partes o haciendo una división del mapa del Distrito Federal en 4 archivos diferentes; correspondiendo cada uno de ellos a una región diferente del Distrito Federal. Una vez obtenidos estos archivos fue necesario hacer una conversión del formato de AutoCAD al formato de archivos utilizado por MapInfo, con el fin de que puedan ser manipulados para su visualización por un navegador WEB

## 3.2 Consultas a Base de Datos y a Mapas

---

Este modulo es donde se realizan las consultas y donde se obtiene la información referente a los sitios turisticos.

El modulo de consultas se ejecuta del lado del servidor totalmente siendo la entrada de este las peticiones del cliente.

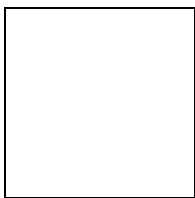
Para realizar la implementación de este modulo se utilizo:

SQL Server 7.x Standar  
JDBC-ODBC  
Servlets  
Windows 2000 AS (como sistema operativo)

### Implementación

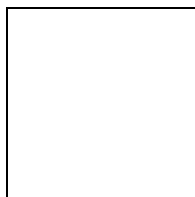
En el esquema utilizado la aplicación Java (API Servlets) es invocado por el usuario con una petición especifica, el servlet se ejecuta en el servidor haciendo una conexión a través del puente JDBC-ODBC con el Manejador de Base de Datos, de aqui en adelante el driver ODBC ejecuta el query en la Base de Datos GIS y regresa los resultados del query.

El servlet regresa una página dinamica desplegando el resultado de la petición.

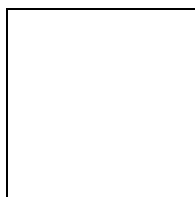


### **Creación de la Base de Datos GIS**

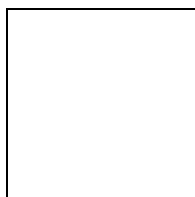
Una vez instalado apropiadamente el SQL Server 7.x, procedemos a la creación de la base de datos GIS. Desde el Administrador Corporativo creamos una nueva base de datos con el nombre de GIS



Se procede a crear las tablas con el Script que se proporciona en este CD. Desde el Analizador de Consultas se ejecuta el Script GIS.SQL en la base de datos GIS.



El resultado de la ejecución se aprecia la creación de las tablas y las relaciones de la Base GIS.

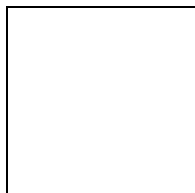


Para poder utilizar las ventajas que nos ofrece el puente JDBC-ODBC se tiene que configurar el Origen de Datos en el Servidor.

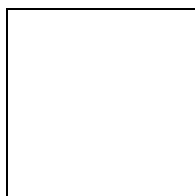
### **Configuración ODBC**

Desde el Panel de Control, Herramientas Administrativas

#### **Origen de Datos (ODBC)**



En el boton agregar



Seleccionamos el origen de datos, en este caso será SQL Server

Se le asigna un nombre de conexión y el nombre del servidor



Asignar con autenticación SQL el Login y el Password



Se elige el archivo de base de datos

De este modo queda configurado el origen de datos ODBC que interactuara con el JDBC posteriormente.

Una vez instalado el Java Web Server 2.0 se procede a la configuración para la ejecución de los servlets.

### 3.3 Conversión de Datos

---

Como ya se había mencionado antes este modulo fue diseñado para que reciba la información de las consultas hechas a la Base de Datos por él modulo anterior, que se interprete esa información para que pueda ser visualizada en un navegador y al mismo tiempo haga la integración del mapa, con los datos mencionados anteriormente.

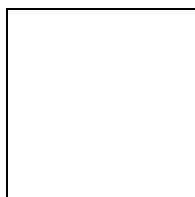
#### **Transformación de código HTML**

Ya que pudo realizarse la conexión a la Base de Datos y la posterior consulta por medio de un servlet instalado en el servidor se toman los datos de la consulta y ahora por el mismo servlet son insertados estos datos dentro de código HTML. Este código es cargado en el navegador del Cliente. Es decir se trabaja con paginas dinámicas que son actualizadas y creadas según las peticiones de información del usuario. Dentro de este código también son generados e insertados los parámetros de consulta que requiere el Applet para que una vez que el usuario decida hacer la petición de la ubicación espacial del sitio en el mapa y así pueda visualizar el mapa.

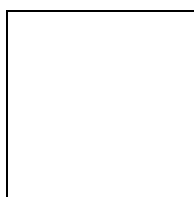
#### **Transformación del mapa a Imagen**

Una vez elaborada la consulta por el servlet y que le pasa los parámetros al applet; este internamente contacta al servidor donde se encuentran los mapas cargados, contacta al mapxtreme servlet . El mapa como ya se menciona esta organizado en diferentes capas. Los parámetros que son pasados al applet por el servlet son:

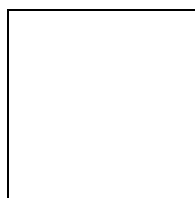
- El nombre de la capa donde se realizara la búsqueda
- El nombre o cadena del atributo a buscar.
- El nombre de la columna donde se deben de buscar el atributo anterior.



Se realiza la consulta por el applet entonces el applet manda llamar funciones del mapxtreme servlet para que efectue una renderización de los datos; obteniéndose una imagen que es presentada dentro de la ventana del applet; Este applet también viene acompañado con herramientas de Zoom In, Zoom Out, Cambiar el origen del mapa que se esta visualizando, cambio de escala y de tamaño de la ventana tal como se muestra a continuación.



Previo a lo anterior tuvo que haberse instalado el JavaWebServer 2.0 por lo menos y sobre el JavaWebServer instalar el MapXtremeServlet para que responda a las solicitudes de diferentes aplicaciones como se muestra a continuación:



El JavaWebServer debe haberse iniciado con los siguientes parámetros :

```
Unidad:\JavaWebServer2.0\bin\httpdnojre -javahome D:\JDK1.3.0_02 -cp  
.;C:\MXJ30\SERVER
```

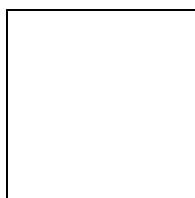
Para que pueda encontrar unos archivos de configuración el MapXtreme Servlet al momento que sea llamado.

# Capítulo 4. Manual de Usuario

## 4.1 Manual de Usuario (pagina)

---

Debido a que nuestro sistema esta montado en una pagina WEB y se trato de que fuera lo más intuitiva posible; entonces no hay necesidad de configuraciones complejas por parte del usuario. Al contrario la operación de nuestra pagina es bastante sencilla.

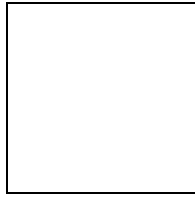


Esta es la pantalla que muestra al inicio la aplicación GIS DF Sistema de Información Turística para el DF.

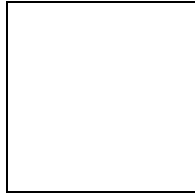


Al presionar sobre la ceja de Arte nos desplegara otra pagina, con una imagen alusiva al tema. Al hacer clic sobre el link arte este despliega en otras subcategorias como Teatros, Museos, Galerías, Plazas, Monumentos.

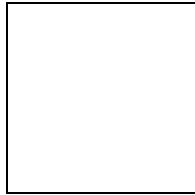
Al hacer clic sobre uno de estos links, por ejemplo en Museos se desplegara una nueva pagina al igual que lo mencionado anteriormente, con una imagen de un museo destacado, dándonos ahora la opción de listar todos los sitios que halle debido a la división en zonas (Norte Sur, Centro, Este y Oeste). También se puede buscar por nombre, solo especificando el tipo de sitio que se este buscando. Esta imagen se muestra abajo.



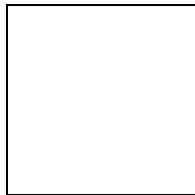
este es el resultado de una búsqueda de museos, seleccionando zona centro, esto es lo que muestra al usuario:



En la caja de Atracciones, también se encuentra aparte de la herramienta de búsqueda, un link de Atracciones, el cual contiene las opciones *museos*, *monumentos* y *parques*.

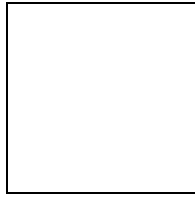


En la caja de Antros, al igual que los demás se encuentra la herramienta de búsqueda por zonas o por el nombre deseado específicamente.

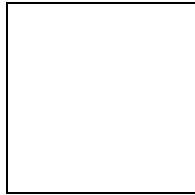


En la caja de Hoteles podemos encontrar la herramienta de búsqueda por zonas o por el nombre deseado específicamente.

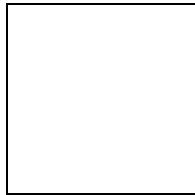




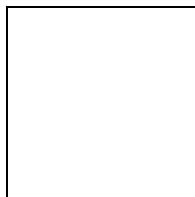
En la ceja de Cines, al igual que los demás se encuentra la herramienta de búsqueda por zonas o por el nombre deseado específicamente.



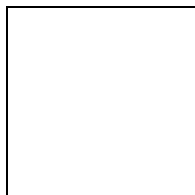
A continuación se muestra la pantalla de salida de una búsqueda de cine:



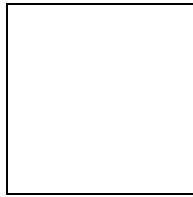
En la ceja de Restaurantes, al igual que los demás se encuentra la herramienta de búsqueda por zonas o por el nombre deseado específicamente.



En la ceja de Tiendas, así como las otras cejas tiene una herramienta de búsqueda por zonas o por el nombre deseado específicamente.



En la caja de Recreación, de la misma forma que las otras cejas, esta también presenta la herramienta de búsqueda por zonas o por el nombre deseado específicamente.

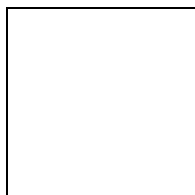


## 4.2 Manual de Usuario (Administración de la Base de Datos)

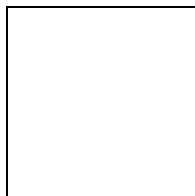
---

Para darle mantenimiento a la base de datos se necesito desarrollar una aplicación, esta fue hecha en Visual Basic 5 a través de ODBC para lograr la conexión con la Base de Datos que se encuentra en SQL Server 7.x.

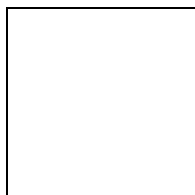
Ejecutar el archivo BASEGIS.EXE, en seguida desplegara la siguiente pantalla, la cual consta de los menús ARCHIVO, CAPTURA, VENTANA y AYUDA.



pantalla principal de la aplicación para mantenimiento de la Base de Datos

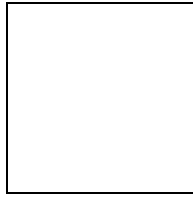


En el menú *Archivo* se encuentra la opción *Salir*, esta opción nos permite salir de la aplicación una vez que se haya terminado de dar el mantenimiento correspondiente a la Base de Datos.



En el menú *Captura* se encuentran las distintas opciones de los lugares que contiene la Base de Datos, tales como *Hoteles*, *Cines*, *Restaurantes*, *Museos* y *Galerías*, *Deportivos*, *Antros*, *Tiendas* o *Plazas Comerciales*, *Servicios*, *Escuelas*, *Teatros*, *Parques* y *Plazas* o *Monumentos*.

Al elegir la opción *Hoteles* se desplegara la siguiente pantalla:



En esta pantalla se solicita el *Id. de inicio de sesión* y su *contraseña*, esto esta restringido ya que no cualquier persona tiene acceso a la Base de Datos, solo el Administrador. La opción de *Usar conexión de confianza*, se utiliza cuando se tienen servidores SQL vinculados con una relación de confianza existente entre si.

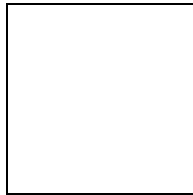
El botón *Opciones >>* despliega la caja de dialogo con mas opciones, estas son *Base de datos*, *Idioma*, *Nombre de la aplicación* y *Id. de estación de trabajo*.

En el campo *Base de datos*, se elige la base de datos que se le desea dar mantenimiento.

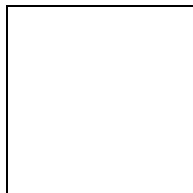
En el campo *Idioma*, se selecciona el Idioma en el que va a regresar los errores de la consulta.

En el campo *Nombre de la aplicación*, se escribe el nombre de la aplicación a la que se requiere acceder.

En el campo *Id. de estación de trabajo*, se selecciona la estación de trabajo a la que se desea realizar la conexión.



inmediatamente después de esta pantalla, se muestra el formulario para dar de alta un sitio (en este caso un Hotel)



En el campo *Id\_sitio\_hotel* se escribe el numero al que hará referencia, correspondiente al mapa. Este numero es único para cada sitio.

En el campo *Id\_zona* se escribe el numero de zona al que pertenece el sitio. Las zonas están separadas de la manera siguiente:

- 1 zona norte
- 2 zona este
- 3 zona centro
- 4 zona oeste
- 5 zona sur

En el campo *mapa\_hotel*, se escribe el numero del mapa al que pertenece, es decir, que debido al gran tamaño del DF, se tuvo que dividir este en cuatro regiones en forma horizontal. También con el fin de un mejor manejo y control de los mapas.

En el campo *nombre\_hotel*, se escribe el nombre del hotel al que se este haciendo referencia.

En el campo *calle\_hotel*, se escribe el nombre de la calle en que se encuentra ubicado este.

En el campo *colonia\_hotel*, la colonia a la que pertenece este.

En el campo *delegacion\_hotel*, se introduce la delegación de la que forma parte este.

En el campo *telefono\_hotel*, se introduce el numero telefónico del sitio turístico.

En el campo *imagen\_hotel*, se hace referencia a la ubicación del archivo de la imagen.

En el campo *vv\_hotel*, se hace referencia a la ubicación de la visita virtual.

En el campo *sonido\_hotel*, se hace referencia a la ubicación en donde se encuentra el archivo de sonido que se reproducirá cuando se elija este sitio turístico.

En el campo *estrella*, se especifica el numero de estrellas que posee el sitio, esto de acuerdo a sus instalaciones y servicios prestados.

En el campo *url\_hotel*, se escribe la dirección de Internet del sitio turístico.

En el campo *info\_hotel*, se escribe información rapida y sobresaliente del sitio turístico.

Los botones que aparecen junto con el formulario del sitio turístico son



# ANEXOS

# ANEXO 1

## Digitalización

### Tipos de Digitalización

Manual o introducción de las coordenadas a través de la tableta digitalizadora. Es el más habitualmente empleado para la generación de mapas en formato vectorial. Un digitalizador o tableta digitalizadora es “un periférico electrónico o electromagnético consistente en una tableta sobre la cual el mapa puede ser colocado”, a lo que añadiríamos que, además de poder ser colocado sobre él, permite la digitalización de coordenadas. Existen diferentes opciones:

- **Punto a punto:** Se introducen a través del cursor de la tableta los puntos más significativos del elemento a digitalizar.
- **Mode stream:** Consistente en repasar con el cursor de la tableta las líneas a Digitalizar, sin necesidad de introducir manualmente los puntos sobre los que se desea que aparezcan vértices.

Básicamente existen tres modalidades de digitalización stream:

- a) **Distancia:** El ordenador registra un punto cada vez que el cursor de la tableta ha recorrido una determinada distancia. Presenta el problema de la reiteración de información en el caso de digitalizar largas líneas rectas.
  - b) **Tiempo:** El ordenador registra un punto por cada periodo de tiempo transcurrido definido previamente. Presenta el problema del exceso o defecto de información en función de la velocidad del digitalizador.
  - c) **Algoritmo:** A través de un algoritmo, el ordenador registra puntos en función de la sinuosidad de la línea digitalizada. Así, en las líneas rectas no almacena puntos, mientras, que en las curvas sí.
- **Introducción de coordenadas:** Con algunos programas de digitalización poco desarrollados el único medio de digitalizar consiste en introducir a través del teclado la lista de coordenadas de los puntos que componen los elementos del mapa. Presenta el problema de la facilidad de cometer errores.
  - **Automático:** Por medio de un escáner o barredor óptico. Presenta la ventaja de la rapidez, pero el documento generado (en formato raster) ha de ser retocado, ya que no se trata de un mapa, sino de un archivo en donde el usuario debe generar las

relaciones topológicas entre elementos. Además, el espacio de almacenamiento que requiere es muy superior al formato vectorial.

### Elementos a digitalizar

#### Puntuales:

descritos por un par de coordenadas. Su tamaño es siempre el mismo. No contienen valor de superficie ni de longitud.



Figura. Puntos

#### Lineales:

de longitud variable. Están definidos por la sucesión de puntos que los componen. Contienen valor de longitud, pero no de superficie.

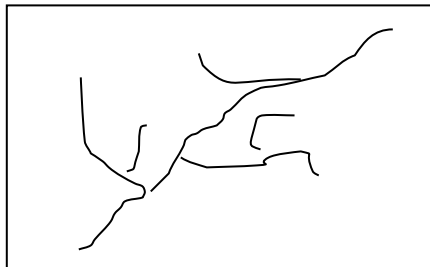


Figura: Líneas

#### Superficiales:

definidos por las líneas que los delimitan. Contienen valor de superficie.

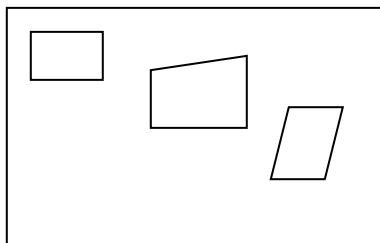


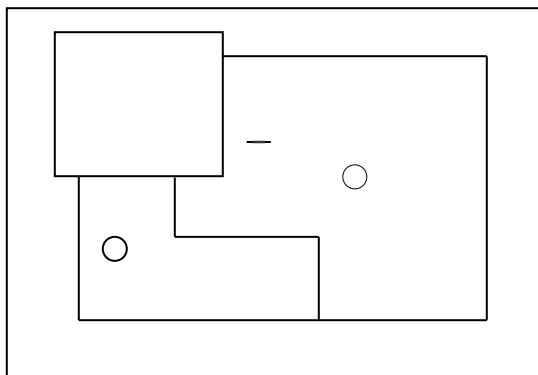
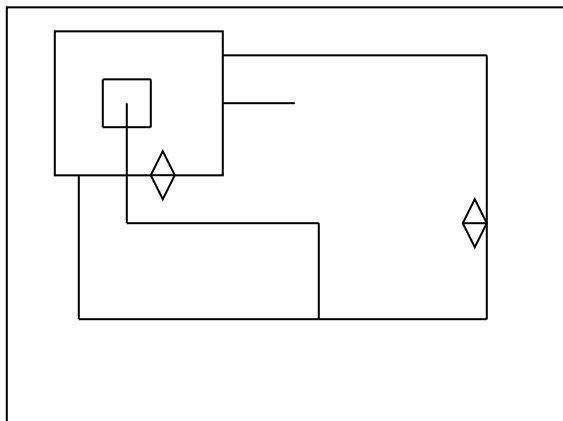
Figura. Polígonos



### Corrección de errores:

Antes de corregir los errores, conviene conocer exactamente cuántos hay y de que tipo son en cada cobertura, ya que dependiendo de su número será procedente corregirlos manualmente o automáticamente. En el caso de coberturas de líneas o de polígonos es necesario suprimir la presencia de nodos colgantes (dangle nodes) que indiquen la existencia de arcos no conectados correctamente o de polígonos abiertos.

**Tipos de errores:** De entre los dos tipos diferentes de *errores de nodo* (dangle y pseudo), hay que prestar especial atención a los primeros, ya que indican la presencia de líneas no conectadas correctamente. Los de tipo pseudo reflejan simplemente la existencia de un nodo que conecta dos arcos diferentes y no tres, como sería lo habitual; es decir, indican la existencia de un solo arco, pero dividido en dos partes con un identificador de usuario, que puede ser distinto para cada una de ellas. Los errores pseudo aparecen habitualmente representados por un pequeño rombo, mientras que los errores dangle aparecen representados por un cuadrado, ambos de color rojo.



### Figura. Errores de etiqueta

En cuanto a los errores de etiqueta, en el caso de las coberturas de polígonos y de cara a la generación de la topología, imprescindible la presencia de una, y sólo una, etiqueta (label) en cada uno de ellos. En caso contrario, no podemos asociar correctamente valores temáticos al mismo, será considerado, por tanto, como un error.

## ANEXO 2

### IMÁGENES

Las imágenes son uno de los elementos que dan más atractivo a las páginas Web. Es por eso que tienen tantos atributos modificadores de la posición, tamaño, etc. Para mostrar gráficos o imágenes en la pantalla, una computadora debe tener un sistema de salida de vídeo. El sistema de salida de vídeo típicamente consiste de un monitor y una tarjeta de vídeo. Las capacidades de salida de vídeo son importantes si se desean ver gráficas de alta resolución en la pantalla.

Una imagen digital es una fotografía convertida a una forma numérica de tal manera que puede ser almacenada y utilizada en una computadora. Las imágenes digitales están construidas de miles de píxeles, que es el término utilizado para identificar los elementos (puntos) de la fotografía. Un píxel es un punto pequeño, la unidad más pequeña de medida de la pantalla. Cuantos más puntos sean, mayor será el detalle de la imagen mostrada. El número de píxeles que pueden ser mostrados es la resolución de la pantalla, medido en píxeles de ancho y píxeles de alto

#### **Graphic Interchange Format (GIF)**

Mapa de imágenes dividido en cuadrículas. Alta compresión. Ideal para imágenes con colores planos. Son la base para imágenes fijas, transparentes y animadas.

Es el formato estándar para imágenes que fue desarrollado por CompuServe para que fuera un método independiente de hardware para el almacenamiento de imágenes. GIF permite gráficas de alta resolución y alta calidad que pueden visualizarse en una gran variedad de sistemas de vídeo. GIF está dirigido a sistemas de vídeo *económicos*, dado que solo puede almacenar 8 bits por píxel (256 o menos colores) y casi todas las PC's pueden mostrar más de 256 colores simultáneamente. GIF es bueno para imágenes con solo unos cuantos colores distintos, como dibujos de líneas, caricaturas simples o capturas de pantalla de aplicaciones

#### GIF Animados

Un GIF Animado es creado utilizando el formato GIF89a. Contiene un conjunto de marcos, por ejemplo. Una secuencia de imágenes GIF que pueden desplegarse una detrás de otra, o en un ciclo, así que se puede ver las imágenes en movimiento. Un GIF animado tiene la misma extensión que un GIF sin movimiento (GIF). Lo que hace al GIF animado especial es que no se tiene que instalar otras aplicaciones en el Navegador de Internet para visualizarlos. Actualmente la mayoría de los navegadores de Internet soportan los GIF animados.

## GIF Transparentes

Un GIF transparente es una imagen GIF con un fondo transparente. Cuando se ve una imagen transparente en el navegador de Internet, se puede observar el fondo del navegador a través del fondo de la imagen, en cualquier color que se elija como el color base del navegador o la imagen que se utilice como fondo. El formato de GIF Transparente permite al navegador de Internet desplegar imágenes de muchas formas, no solo cuadradas o rectangulares que se puedan crear en la mayoría de los editores de gráficos. Un GIF transparente tiene la misma extensión (.GIF) de una imagen en formato GIF normal.

El formato GIF se emplea con imágenes de 256 colores, por lo que es más apropiado para iconos y dibujos. El método de compresión que usa hace que no pierda calidad, de modo que al descomprimir la imagen queda como el original sin comprimir. Hay algunas variantes de este formato:

Los Gif animados son una sucesión de imágenes, pero que se comportan a todos los efectos como una sola. Los Gif transparentes, son aquellos en los que un color o más, de los 256 posibles, no se visualiza. Es muy recomendable que los ficheros de imágenes sean pequeños. Un icono Gif no debe ocupar más de 3 o 4 Kb y una foto en JPG menos de 20 o 30 Kb.

## GIF entrelazados

Normalmente, un fichero GIF contiene los datos de cada línea de la imagen de una manera ordenada, de tal manera que al ser cargada por el navegador aparecerá dibujada línea a línea desde arriba hasta abajo.

Se puede cambiar este comportamiento si se ha guardado la imagen como un GIF entrelazado (interlaced GIF). En este caso, las líneas quedan guardadas no de una manera consecutiva, sino en saltos de cuatro en cuatro, y al llegar al final recomienza desde el principio con otra secuencia diferente, también de cuatro en cuatro, así hasta completar la imagen.

Por ejemplo, una imagen de 20 líneas sería guardada con estas secuencias:

líneas 1	5	9	13	17
----------	---	---	----	----

líneas 2	6	10	14	18
líneas 3	7	11	15	19
líneas 4	8	12	16	20

El propósito de esto es que al ser cargada esta imagen por un navegador que implemente esta característica (Netscape, Explorador de Microsoft, etc), se verá la imagen completa desde el principio, comenzando con una definición muy grosera que luego se va afinando poco a poco.

El tiempo de carga de una imagen entrelazada y de la misma no entrelazada es el mismo, pero en el primer caso nos hacemos rápidamente una idea del tipo de imagen que es, lo cual puede ser muy conveniente a veces.

### **Joint Photographic Expert Group (JPEG)**

Sistema de compresión de imágenes basado en vectores. Especializado en imágenes de alta resolución y contraste, como fotografías.

El método de compresión utilizado por el formato GIF es muy conveniente para comprimir áreas monocolor, como las que puede haber por ejemplo en un diagrama sencillo. Pero no es tan conveniente para cosas más complicadas, como por ejemplo fotografías de objetos de la vida real. Además estos objetos no presentan un aspecto nada favorable si están reducidos a los 256 colores de este formato.

Para realizar este tipo de tareas se creó el formato JPEG. Almacena las imágenes con 16.7 millones de colores, que es mucho más de lo que el ojo humano puede distinguir, y hace uso de complicados algoritmos matemáticos para comprimir el tamaño de los ficheros hasta la décima parte de su tamaño original. Esto quiere decir que una imagen complicada que se haya comprimido con este formato tendrá, comparando con el formato GIF, una calidad mayor y un tamaño de fichero menor.

Su único inconveniente es que hay navegadores que no son capaces de manejar directamente estos ficheros y que tienen que traspasarlos a un programa auxiliar para ejecutarlos. Esto quiere decir, que si se quiere tener en cuenta a esos navegadores, no se deben poner imágenes con este formato directamente en la página. Un recurso muy utilizado para solventar esto es el de poner thumbnails en formato GIF que enlacen con las imágenes en tamaño natural y formato JPEG.

## Imágenes Panorámicas

En cierta forma de representar entornos 3D (mediante fotografías o generadas por ordenador) en una pantalla de ordenador (2D). Este tipo de imágenes nos muestran una visión esférica completa (360° x 180°) del mundo en un área rectangular de la pantalla del ordenador, dando la sensación de que se esta mirando por una ventana, la cual se puede mover en todas las direcciones.

Necesitamos dos elementos: el hardware para obtener las imágenes (ej. una cámara fotográfica digital) y los programas para procesarlas. Generar este tipo de imágenes, utilizando el software adecuado, es sencillo, lo difícil es *capturar* las imágenes. Normalmente los programas que generan la imagen panorámica lo hacen a partir de una serie de imágenes, tomadas con una cámara digital. Otra posibilidad es extraer las imágenes de una secuencia de vídeo.

Hay bastantes programas para generar imágenes panorámicas (dinámicas) a partir de un conjunto de imágenes estáticas (fotografías o imágenes generadas por ordenador) a continuación se presentan las características de algunos de ellos.

### **LIVEPICTURE PHOTOVISTA**

Es como el QuickTime VR, pero el software esta disponible para Power Macintosh y Windows 95/NT. Hay un visor para Windows y un Plug-In para el navegador de Internet

## ANEXO 3

### M A P A S

Los mapas están organizados en archivos que contienen registros y mapas. Los datos pueden ser especificados en el formato MapInfo que a continuación se especifica. Los archivos.tab están organizados en un grupo de archivos para construir una capa que constituye un mapa. Las tablas MapInfo consisten de los siguientes archivos:

#### **XXXXX.tab**

Este archivo describe la estructura de la tabla de MapInfo. Es un pequeño archivo de texto describiendo el formato de el archivo que contiene los datos.

#### **XXXXX.dat (\*.mdb, \*.dbf, \*.txt, \*.xls, \*.wks):**

Estos archivos contienen los datos tabulares (base de datos).

#### **XXXXX.map**

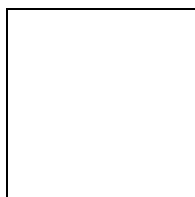
Este archivo describe los objetos gráficos (no existirá este archivo si la tabla no contiene objetos de mapa).

#### **XXXXX.id:**

Este archivo es una referencia cruzada que liga los datos con los objetos (no existirá si la tabla no contiene objetos de mapa).

#### **XXXXX.ind:**

Este es un archivo de índice. El archivo de índice permite hacer búsquedas de objetos de mapa(no existirá si la tabla no contiene objetos de mapa)



### **Tablas de MapInfo y capas de MapXtreme**

Cada Tabla mapeable de MapInfo puede ser desplegada en una aplicación realizada en MapXtreme como una capa en un mapa. Por ejemplo, usted puede desplegar una capa de clientes, una de calles y una capa de delegaciones.

Se debe de pensar en estas capas como transparencias donde cada capa contiene diferentes partes de el mapa. Las capas son apiladas una arriba de la otra y permite ver todos los aspectos del mapa al mismo tiempo. Las capas son dibujadas del fondo hacia arriba.

## MapInfo Geosets

Una colección de archivos.tab es llamada un geoset. Los Geoset son similares en concepto a un espacio de trabajo. En lugar de abrir individualmente los archivos.tab, usted puede abrir el geoset y desplegar todas las capas con sus características particulares.

Los Geosets están limitados en su funcionalidad en MapXtreme de Java, de cualquier forma. Ellos específicamente se refieren a un archivo \*.tab que no puede ser salvado en una base de datos remota.

## Definiciones de Mapa

Las definiciones de mapa no son nuevas en MapXtreme de Java, pero su formato si es nuevo. En versiones previas; las definiciones de mapa eran almacenados como objetos Java serializados en un formato binario. Cambiando el formato y dejándolo basado en un archivo de texto XML, las definiciones de mapa son ahora más compatibles y editables.

Cuando es cargada una definición de mapa, MapXtreme automáticamente lee los meta datos en una definición de mapa y fija las propiedades de visualización del mapa. Las características de una definición de mapa incluyen: proyección, zoom default, propiedades de etiquetado, y la ubicación donde la tabla esta abierta.

Las definiciones de mapa son creadas usando el Map Definition Manager que viene con MapXtreme de Java. Definiciones de mapa de versiones anteriores de MapXtreme no trabajaran con esta versión. Porque usted tiene mas flexibilidad y control cuando se trabaja con definiciones de mapa que cuando trabaja con Geoset; usted puede convertir los Geosets en Definiciones de Mapa.

## Características de Mapa

Mencionamos antes que los Mapas en MapXtreme son hechos de capas de objetos del mapa. Estos objetos del Mapa son accesados en MapXtreme a través de la característica del objeto. Estas son 3 características de tipo básico:

- **Regiones:** objetos cerrados que cubren un área dada. Estos incluyen polígonos, elipses y rectángulos. Las regiones incluyen los limites del continente, limites del código postal, territorios vendidos entre otros.
- **Objetos puntuales:** representa una locación sencilla de datos. Algunos ejemplos incluyen locaciones de los clientes, restaurantes y locales de estacionamiento.
- **Objetos lineales:** abrir objetos que cubren una distancia dada. Estas incluyen poli líneas y arcos. Por ejemplo las calles, ríos, líneas de poder.

Puedes tener cada tipo de objeto en capas separadas (lo más común), o puedes combinar objetos en la misma capa. MapXtreme deja modificar y visualizar los objetos para hacer mapas de acuerdo a tus necesidades.



### **Renditions o Renditions**

Todas las características en un mapa de MapXtreme Java se visualiza con ciertas características visibles, como el color o tipo de símbolo. Se puede tener el control completo sobre como se observan las características del mapa. Esta versión de MapXtreme Java incluye muchas propiedades nuevas de visualizar para tomar mejor ventaja de las capacidades de representación del API de Java2D, entre pintar los símbolos para líneas y regiones, el rayado y líneas paralelas y símbolos vectoriales.

Las interpretaciones pueden establecerse programaticamente o a través del administrador de definiciones de Mapas de la capa de control de dialogo.

### **Etiquetado**

Relacionado a las interpretaciones en que son desplegadas las características, el etiquetado es una característica poderosa en su derecho de realzar su mapa e impartir los mensajes apropiados para el visor. El etiquetado es mas que solo adicionar texto para describir la característica del mapa. Con MapXtreme Java tu puedes controlar el tipo de letra, tamaño, color, posición y uso creativo de efectos como el contorno, para crear etiquetas distintivas para cada capa de su mapa.

## **MapXtreme for Java**

Una característica poderosa de asociar la familia de MapInfo Corporation's con el software es la capacidad de habilitar y analizar el mapa para aprender mas de lo que tu podrías si tu únicamente viste los datos en formato de columna. Desplegando los datos de tu mapa puedes hacer comparaciones visuales de tus datos con cuales sirve mejor para las decisiones de negocios.

MapXtreme Java provee un numero de formas para analizar el mapa, usando las herramientas del mapa para seleccionar las características, creando mapas temáticos que visualicen uniones entre las características, buscando datos subyacentes de acuerdo a tu propio criterio.

Muchas de estas guías de desarrolladores proveen información de como obtener tus datos de afuera.

`/*_/*_/*_/*_/*_/*_/*_`

### **Que es MapXtreme Java ?**

MapXtreme Java es una aplicación asociada a desarrollar herramientas para organizaciones las que reconocen que la visualización de los datos y mapeo pueden ayudarlos a tomar mejores decisiones de negocios y administrar activos y operaciones mas efectivamente. Las aplicaciones corriendo en el servidor de red administrado ofrece grandes escalas de economía incluyendo bajos costos de hardware y administrativos mientras dramáticamente perfeccionando el funcionamiento de la aplicación, confiabilidad y seguridad. Compañías que una vez encontraron prohibitivos los costos del mapeo, pueden ahora desplegar aplicaciones a un bajo costo por usuario mas que nunca antes. Las aplicaciones construidas con MapXtreme Java son apropiadas ambas para intranets corporativas y al publico de Internet.

### **Características de MapXtreme Java**

MapXtreme Java ha sido rediseñado usando la arquitectura Servlet. Esto permite enfocar a MapXtreme para las peticiones, mientras el servidor web / contenedor de Servlets maneje otras ediciones al lado de servidor como por ejemplo al balancear la carga, seguridad y tolerancia de incidente. Adicionalmente, el modelo de servidor usa HTTP, el estándar para comunicaciones a través de Internet. Otras características de MapXtreme Java incluyen:

Sofisticadas capacidades de asociar como una selección de comportamiento, asociar el tema y análisis, y etiquetado avanzado y representado.

Acceso a datos fuentes a través de JDBC que te permite mantener tus datos espaciales seguros en una RDBMS, aun usando su potencial completo para crear mapas.

Conveniencias para una fácil instalación y configuración, desarrollo y despliegue, como JavaBeans, aplicaciones de ejemplo, han incluido la Maquina Virtual Java 2. El Servlet contenedor Tomcat, y la aplicación servidor HAHTsite and IDE.

### **Beneficios**

MapXtreme una multi-plataforma, alta calidad, alto funcionamiento, una solución fácil de usar para tus necesidades de mapeo. Los beneficios de MapXtreme son muchos:

#### **\* Multi-plataforma**

Porque la seguridad, confiabilidad, y funcionamiento, muchos usuarios finales de industrias como es telecom y el uso seguro de UNIX, mientras otros usuarios en áreas de la misma organización están trabajando en sistemas Windows. A menudo existe una necesidad de desplegar una solución similar en muchas plataformas. Las aplicaciones basadas en el mapeo de Java permite a los desarrolladores escribir un simple programa para uso en múltiples plataformas que soportan Maquinas Virtuales.

Con MapXtreme Java corriendo en el lado del servidor, existiendo UNIX o Windows los recursos pueden ser usados. Los datos pueden ser almacenados y manipulados en un sistema y programaticamente accedidos de otra maquina corriendo Maquina Virtual.

## **Alta Escalabilidad**

Creando organización empresarial al ancho mapeo de soluciones con MapXtreme necesita aplicaciones que funcionen bien y pueda soportar a todos los usuarios que necesiten acceder. Desde que MapXtreme ofrece este componente basado, confiable y solución multi-hilo. La versatilidad es asegurada. Tu aplicación puede crecer con las necesidades de tu organización.

### **Inteligente Multi-Threading**

MapXtreme usa inteligentemente Java threading para servir a múltiples usuarios concurrentes eficientemente. Si tuvo un consumo bajo de memoria por usuario y el usuario escalable carga como CPU's adicionales fueron adicionados. Las pruebas demostraron que el motor de prueba de MapXtreme requiere cerca de 8 MB de memoria en constante estado y de 100 KB a 200 KB por usuario concurrente. Por ejemplo, mientras un hilo esta procesando una petición de un mapa, otros 3 hilos podrían fluir los resultados de 3 previas peticiones al mapa vía red I/O simultáneamente.

### **Basado en Flexibilidad del Componente**

MapXtreme ofrece un gran flexibilidad de desplegar porque su arquitectura basada en componentes. Hay cuatro altos niveles de componentes: el objeto MapJ, la renderización de mapas para visualizar los mapas, proveedor de datos para acceder varias fuentes de datos y el MapXtreme Servlet. MapXtreme puede ser usado en dos grados despliegue que pone MapJ en el lado del cliente o la configuración de tres grados para Internet con MapXtreme Java y tu negocio lógico en medio grado.

### **Fuerte conectividad para establecer datos remotos**

MapXtreme tiene la tendencia de almacenar datos espaciales en base de datos como es Oracle 8i como la opción espacial y el servidor dinámico de Informix con el spatialWare BaseBlade. Esto te permite proteger tu misión crítica de datos espaciales en un nivel de empresa administrando la Base de Datos del sistema provee acceso apropiado para algún usuario en la World Wide Web.

### **Compatible con cualquier ambiente Web**

MapXtreme es de arquitectura abierta con algún ambiente Web virtual. Y trabajar con algún servidor Web que soporte ISAPI, NSAPI or CGI gateways, como es Netscape. Apache, o Información de Microsoft Internet Server. Como es ambiente del Servlet como un ambiente, esto provee todos los beneficios de los Servlets API de Sun's Java.

## **Ventajas Programando**

### **Orientada a Objetos**

MapXtreme es orientada a objetos con un fácil uso del modelo jerárquico de objetos para el despliegue del mapeo, consultas y análisis.

### **MapJ API**

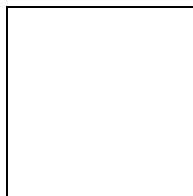
El MapJ API es el server-side API usada para comunicarse con MapXtremeServlet. Cada cliente hace una petición al mapa de MapXtreme usado (o reutilizado) una instancia del objeto MapJ. MapXtreme no tiene plug-ins propiamente, así que entrega el mapa para cualquier navegador o algún sistema operativo. MapXtreme es asíncrono, multi-hilo y sin estado definido para un máximo funcionamiento.

# ANEXO 4

## SQL SERVER

### INTRODUCCIÓN

SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS) que usa Transact-SQL para mandar peticiones entre un cliente y el SQL Server.



### ARQUITECTURA CLIENTE / SERVIDOR:

SQL Server usa la arquitectura Cliente / Servidor para separar la carga de trabajo en tareas que corran en computadoras tipo Servidor y tareas que corran en computadoras tipo Cliente:

- El Cliente es responsable de la parte lógica y de presentar la información al usuario. Generalmente, el cliente corre en una o más computadoras Cliente, aunque también puede correr en una computadora Servidor con SQL Server.
- SQL Server administra Bases de Datos y distribuye los recursos disponibles del servidor (tales como memoria, operaciones de disco, etc) entre las múltiples peticiones.

La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

### SISTEMA ADMINISTRADOR PARA BASES DE DATOS RELACIONALES (RDBMS):

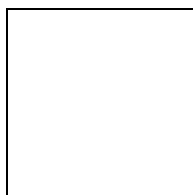
El RDBMS es responsable de:

- Mantener las relaciones entre la información y la Base de Datos.
- Asegurarse de que la información es almacenada correctamente, es decir, que las reglas que definen las relaciones ente los datos no sean violadas.
- Recuperar toda la información en un punto conocido en caso de que el sistema falle.

### **TRANSACT - SQL:**

Éste es una versión de SQL (Structured Query Lenguaje) usado como lenguaje de programación para SQL Server. SQL es un conjunto de comandos que permite especificar la información que se desea restaurar o modificar. Con Transact – SQL se puede tener acceso a la información, realizar búsquedas, actualizar y administrar sistemas de Bases de Datos Relacionales.

### **PLATAFORMAS PARA SQL**



*Figura 2*

Los componentes Cliente y Servidor de SQL Server corren en los Sistemas Operativos mostrados en la siguiente tabla:

<b>PLATAFORMA</b>	<b>COMPONENTE SERVER</b>	<b>COMPONENTE CLIENTE</b>
Microsoft Win 95/98	Si	Si
Microsoft Windows NT Workstation 4.0 y posteriores	Si	Si
Microsoft Windows NT Server 4.0 y posteriores	Si	Si
Microsoft Windows NT Server Enterprise Edition 4.0 y	Si	Si

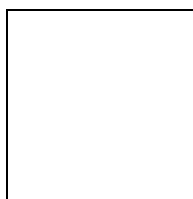
## SEGURIDAD EN SQL SERVER

SQL Server está integrado con el sistema de seguridad de Windows NT. Esta integración permite acceder tanto a Windows NT como a SQL Server con el mismo *user name* y *password*. Además SQL Server una las características de encriptación que Windows NT para la seguridad en red. SQL Server está provisto de su propia seguridad para clientes *no-Microsoft*.

SQL Server valida a los usuarios con 2 niveles de seguridad; autenticación del login y validación de permisos en la Base de Datos de cuentas de usuarios y de roles. La autenticación identifica al usuario que está usando una cuenta y verifica sólo la habilidad de conectarse con SQL Server. El usuario debe tener permiso para acceder a las Bases de Datos en el Servidor. Esto se cumple para asignar permisos específicos para la Base de Datos, para las cuentas de usuario y los roles. Los permisos controlan las actividades que el usuario tiene permitido realizar en la Base de Datos del SQL Server.

## SERVICIOS DE SQL SERVER

Los servicios de SQL Server incluyen MSSQLServer, SQLServerAgent, Microsoft Distributed Transaction Coordinator (MSDTC), y Microsft Search. Aunque estos servicios de SQL generalmente corren en Windows NT, también pueden correr como aplicaciones.



### SERVICIO MSSQLServer:

Este servicio es el motor de la Base de Datos. Este es el componente que procesa todas las declaraciones de Transact-SQL y administra todos los archivos que definen a la Base de Datos dentro del Servidor. Sus características son:

- Asignar los recursos de la computadora a múltiples usuarios simultáneos.
- Previene problemas lógicos, tales como sincronización de peticiones de usuarios que desean actualizar la misma información al mismo tiempo.
- Garantiza la integridad y consistencia de datos.

### **SERVICIO SQLServerAgent:**

Este es un servicio que trabaja conjuntamente con SQL Server para crear y administrar tareas locales o externas; letras y operadores.

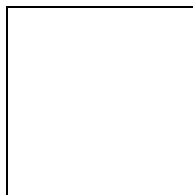
### **HERAMIENTAS Y ASISTENTES PARA ADMINISTRACIÓN DE SQL SERVER:**

Sql Server provee un número de herramientas administrativas y asistentes que atienden aspectos particulares de SQL Server. La siguiente tabla describe las herramientas y asistentes de SQL Server:

<b>HERRAMIENTA GRÁFICA</b>	<b>APLICACIÓN</b>
Configuración Cliente de SQL Server	Utilidad para administrar la configuración cliente para componentes de comunicación
Monitor de Funcionamiento de SQL Server	Archivo usado para integrar SQL Server con El Monitor de Funcionamiento de Windows NT, para informar las estadísticas más recientes de actividad
SQL Server Profiler	Utilidad para capturar el record continuo de la actividad del servidor
Analizador de Queries de SQL Server	Herramienta gráfica de Queries usada para analizar el plan de un query, visualizar información estadística, y administrar varios queries en diferentes ventanas al mismo tiempo.

## **ARQUITECTURA DE SQL SERVER**

### **COMUNICACIÓN:**





SQL Server usa una arquitectura de comunicación por capas para aislar aplicaciones internas de red y protocolos. Esta arquitectura permite desplegar la misma aplicación en diferentes ambientes de red.

### **ADMINISTRACIÓN DE SQL SERVER:**

SQL Server puede ser administrado usando:

- Utilidades Batch incluidas en SQL Server, tales como OSQL o BCP.
- Herramientas de administración gráfica incluidas en SQL Server.
- Aplicaciones COM-compatibles: tal como Visual Basic.

### **IMPLEMENTACIÓN DE UNA BASE DE DATOS EN SQL SERVER:**

Implementar una Base de Datos en SQL Server significa planear, crear y mantener un número de componentes interrelacionados. La naturaleza y complejidad de una aplicación de Base de Datos, así como el proceso de planearla puede variar enormemente. Por ejemplo, una Base de Datos puede ser relativamente simple, diseñada para ser usada por una sola persona, o puede ser grande y compleja, diseñada para atender todas las transacciones de cientos o miles de clientes.

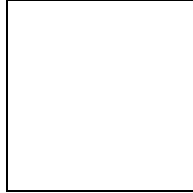
En cuanto al tamaño y complejidad de la Base de Datos, generalmente la implementación de una Base de Datos involucra:

1. Diseñar la Base de Datos de manera que la aplicación optimice el uso de Hardware y permita crecimiento futuro, identificar y modelar objetos de la Base de Datos y aplicaciones de lógica, y especificar tipos de información para cada objeto y tipo de relación.
2. Crear la Base de Datos y los objetos, incluyendo tablas, mecanismos de integridad de datos, entrada de datos y objetos, índices y seguridad.
3. Probar la aplicación y la base de Datos. Cuando se diseña una Base de Datos, se desea asegurar que la Base de Datos realiza las funciones importantes en forma rápida y correcta.
4. Planear el funcionamiento, lo que incluye analizar la carga de trabajo y recomendar una configuración óptima para la Base de Datos de SQL Server.
5. Administrar la aplicación, lo que incluye configurar a los clientes y servidores, monitorear el funcionamiento del server, administrar tareas, alertas y operadores, administrar seguridad y procedimiento de backup de la Base de Datos.

### **BASES DE DATOS EN SQL SERVER**

Cada SQL Server tiene dos tipos de Bases de datos: Bases de Datos del Sistema y Bases de Datos del usuario. Las Bases de Datos del sistema almacenan información acerca de SQL Server como un total. SQL Server usa la Base de Datos del sistema para operar y

administrar al sistema. Las Bases de Datos de usuarios, son Bases de Datos creadas por los usuarios. Una copia del SQL Server puede administra una o más Bases de datos de usuario.



### **ADMINISTRACIÓN DE UNA BASE DE DATOS DE SQL SERVER:**

Abarca 3 puntos importantes:

1. Instalar y configurar SQL Server y establecer la seguridad de red.
2. construir las Bases de Datos: incluye asignar espacio en disco para la Base de Datos y la conexión, transferir datos de y hacia la Base de Datos, definir e implementar la seguridad de la base de Datos y crear trabajos automatizados para ciertas tareas.
3. Administrar actividades entrantes, como la importación y exportación de datos, respaldar y restaurar la base de Datos y la conexión, y monitorear la Base de Datos. Una tarea opcional es automatizar algunas de estas tareas administrativas recurrentes.

# ANEXO 5

## JDBC

Java Database Connectivity (JDBC) es una interfase de acceso a bases de datos estándar SQL que proporciona un acceso uniforme a una gran variedad de bases de datos relacionales. JDBC también proporciona una base común para la construcción de herramientas y utilidades de alto nivel. El paquete actual de JDK incluye JDBC y el puente JDBC-ODBC. Estos paquetes son para su uso con JDK 1.0

### Drivers JDBC

Para usar JDBC con un sistema gestor de base de datos en particular, es necesario disponer del driver JDBC apropiado que haga de intermediario entre ésta y JDBC. Dependiendo de varios factores, este driver puede estar escrito en Java puro, o ser una mezcla de Java y métodos nativos JNI (Java Native Interface). JDBC es el API para la ejecución de sentencias SQL. (Como punto de interés JDBC es una marca registrada y no un acrónimo, no obstante a menudo es conocido como “Java Database Connectivity”). Consiste en un conjunto de clases e interfases escritas en el lenguaje de programación Java. JDBC suministra un API estándar para los desarrolladores y hace posible escribir aplicaciones de base de datos usando un API puro Java.

Usando JDBC es fácil enviar sentencias SQL virtualmente a cualquier sistema de base de datos. En otras palabras, con el API JDBC, no es necesario escribir un programa que acceda a una base de datos Sybase, otro para acceder a Oracle y otro para acceder a Informix. Un único programa escrito usando el API JDBC y el programa será capaz de enviar sentencias SQL a la base de datos apropiada. Y, con una aplicación escrita en el lenguaje de programación Java, tampoco es necesario escribir diferentes aplicaciones para ejecutar en diferentes plataformas. La combinación de Java y JDBC permite al programador escribir una sola vez y ejecutarlo en cualquier entorno.

JDBC expande las posibilidades de Java. Por ejemplo, con Java y JDBC API, es posible publicar una página web que contenga un applet que usa información obtenida de una base de datos remota. O una empresa puede usar JDBC para conectar a todos sus empleados (incluso si usan un conglomerado de máquinas Windows, Macintosh y UNIX) a una base de datos interna vía intranet. Con cada vez más y más programadores desarrollando en lenguaje Java, la necesidad de acceso fácil a base de datos desde Java continúa creciendo.

### ¿Qué hace JDBC?

Simplemente JDBC hace posible estas tres cosas:

- Establece una conexión con la base de datos.
- Envía sentencias SQL

- Procesa los resultados.

### **JDBC es un API de bajo nivel y una base para API's de alto nivel.**

JDBC es una interfase de bajo nivel, lo que quiere decir que se usa para 'invocar' o llamar a comandos SQL directamente. En esta función trabaja muy bien y es más fácil de usar que otros API's de conexión a bases de datos, pero está diseñado de forma que también sea la base sobre la cual construir interfaces y herramientas de alto nivel. Una interfase de alto nivel es 'amigable', usa un API más entendible o más conveniente que luego se traduce en la interfase de bajo nivel tal como JDBC.

### **JDBC frente a ODBC y otros API's**

ODBC no es apropiado para su uso directo con Java porque usa una interface C. Las llamadas desde Java a código nativo C tienen un número de inconvenientes en la seguridad, la implementación, la robustez y en la portabilidad automática de las aplicaciones.

1. Una traducción literal del API C de ODBC en el API Java podría no ser deseable. Por ejemplo, Java no tiene punteros, y ODBC hace un uso copioso de ellos, incluyendo el notoriamente propenso a errores "void \* ". Se puede pensar en JDBC como un ODBC traducido a una interfase orientada a objeto que es el natural para programadores Java.
2. ODBC es difícil de aprender. Mezcla características simples y avanzadas juntas, y sus opciones son complejas para 'queries' simples. JDBC por otro lado, ha sido diseñado para mantener las cosas sencillas mientras que permite las características avanzadas cuando éstas son necesarias.
3. Un API Java como JDBC es necesario en orden a permitir una solución Java "pura". Cuando se usa ODBC, el gestor de drivers de ODBC y los drivers deben instalarse manualmente en cada máquina cliente. Como el driver JDBC esta completamente escrito en Java, el código JDBC es automáticamente instalable, portable y seguro en todas las plataformas Java.

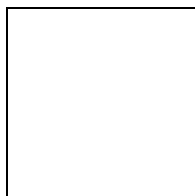
En resumen, el API JDBC es el interfase natural de Java para las abstracciones y conceptos básicos de SQL. JDBC retiene las características básicas de diseño de ODBC; de hecho, ambos interfaces están basados en el X/Open SQL CLI (Call Level Interface).

En el modelo de tres-pisos, los comandos se envían a un 'piso intermedio' de servicios, que envía las sentencias SQL a la base de datos. La base de datos procesa las sentencias SQL y devuelve los resultados a el 'piso intermedio', que a su vez lo envía al usuario. Los directores de IS encuentran este modelo muy atractivo por que el 'piso intermedio' hace posible mantener el control sobre los datos y los tipos de actualizaciones que pueden hacerse en los datos corporativos. Otra ventaja es que el usuario puede emplear un API de alto nivel más sencillo que es traducido por el 'piso intermedio' en las llamadas de bajo

nivel apropiadas. Finalmente en muchos casos la arquitectura de tres niveles puede proporcionar ventajas de rendimiento.

Hasta ahora, este nivel intermedio ha sido escrito en lenguajes como C ó C++, que ofrecen un rendimiento más rápido. De cualquier modo, con la introducción de compiladores optimizadores que traducen el bytecode en código máquina eficiente, se está convirtiendo en práctico desarrollar este nivel intermedio en Java.

Esta es una gran ventaja al hacer posible aprovechar las características de robustez, multiproceso y seguridad de Java.



## **JDBC y URL's**

Una URL JDBC suministra una forma de identificar una base de datos para que el driver apropiado pueda reconocerla y establecer la conexión con ella. Los desarrolladores de drivers son los que determinan actualmente que JDBC URL identifica su driver particular. Los usuarios no tienen por que preocuparse sobre como se forma una URL JDBC; ellos simplemente usan la URL suministrada con el driver que usan. El rol de JDBC es recomendar algunas convenciones a los fabricantes de drivers para su uso.

Dado que los JDBC URL se usan con varios tipos de drivers, las convenciones son necesariamente muy flexibles. Primero, permiten a diferentes drivers usar diferentes esquemas para nombrar las bases de datos. EL subprotocolo `odbc`, por ejemplo, permite que las URL contengan valores de atributos (pero no los requieren).

Segundo, las URL's JDBC permiten a los desarrolladores de drivers codificar toda la información de la conexión dentro de ella. Esto hace posible, por ejemplo, para un applet que quiera hablar con una base de datos dada el abrir la conexión sin necesitar que el usuario realice ninguna tarea de administración de sistemas.

Tercero, las URL's JDBC permiten un nivel de indirección. Esto quiere decir que la URL JDBC puede referirse a una base de datos lógica o un host lógico que se traduce dinámicamente al nombre actual por el sistema de nombramiento de la red.

Esto permite a los administradores de sistemas evitar dar especificaciones de sus hosts como parte del nombre JDBC. Hay una variedad de servicios de nomenclatura de red diferentes (tales como DNS, NIS y DCE), y no hay restricción acerca de cual usar.

La sintaxis para las URL's JDBC que se muestra a continuación tiene tres partes separadas por dos puntos:

**`jdbc:<subprotocol>:<subname>`**

Las tres partes se descomponen como sigue:

1. **jdbc** - el protocolo. El protocolo en una URL JDBC es siempre jdbc
2. <subprotocol> - el nombre del driver o el nombre del mecanismo de conectividad con la base de datos, que puede estar soportado por uno o más drivers. Un ejemplo sobresaliente de un subprotocolo es “odbc”, que ha sido reservado para URL’s que especifican nombres de fuentes de datos estilo ODBC. Por ejemplo para acceder a la base de datos a través del puente JDBC-ODBC, la URL a usar podría ser algo así como lo siguiente:

**jdbc:odbc:fred**

En este ejemplo, el subprotocolo es “odbc” y el subnombre “fred” es el nombre de la fuente de datos ODBC local. Si se quiere usar un servicio de nombre de la red (ya que el nombre de la base de datos en la URL JDBC no tiene por que ser su nombre actual), el servicio de nombre puede ser el subprotocolo. Por tanto, para el ejemplo, podría tener una URL como :

**jdbc:dce:accounts-payable**

En este ejemplo, la URL especifica que el servicio local DCE resolverá el nombre de la base de datos “accounts-payable” para poder conectar con la base de datos real.

<**subname**> - una forma de identificar la base de datos. El subnombre puede variar dependiendo del subprotocolo, y puede tener un subnombre con cualquier sintaxis interna que el fabricante del driver haya escogido. El punto de un subnombre es para dar información suficiente para localizar la base de datos. En el ejemplo anterior “fred” es suficiente porque ODBC suministra la información restante. Una base de datos en un servidor remoto requiere más información. Si la base de datos va a ser accesible a través de Internet, por ejemplo, la dirección de red debería incluirse en la URL JDBC como parte del subnombre y debería seguir a la convención estándar de nomenclatura de URL.

**//hostname:port/subsubname**

Suponiendo que “dbnet” es un protocolo para conectar a un host, la URL JDBC debería parecerse a algo como:

**jdbc:dbnet://wombat:356/fred**

## **Establecer una conexión**

Una vez que la clase Driver ha sido cargada y registrada con la clase DriverManager, se está en condiciones de establecer la conexión con la base de datos. La solicitud de la

conexión se realiza mediante una llamada al método `DriverManager.getConnection`, y `DriverManager` prueba los drivers registrados para ver si puede establecer la conexión.

A veces puede darse el caso de que más de un driver JDBC pueda establecer la conexión para una URL dada. Por ejemplo, cuando conectamos con una base de datos remota, podría ser posible usar un driver puente JDBC-ODBC, o un driver JDBC de protocolo genérico de red, o un driver suministrado por el vendedor.

En tales casos, el orden en que los driver son probados es significativo porque `DriverManager` usará el primer driver que encuentre que pueda conectar con éxito a la base de datos. Primero `DriverManager` intenta usar cada driver en el orden en que ha sido registrado ( los drivers listados en la propiedad `jdbc.drivers` son siempre los registrados primero). Saltará cualquier driver con código 'untrusted', a menos que se cargue desde el mismo código fuente que el código que intenta abrir la conexión.

Prueba los drivers mediante la llamada al método `Driver.connect` cada uno por turno, pasándole como argumento la URL que el usuario ha pasado originalmente al método `DriverManager.getConnection`. El primer driver que reconozca la URL realiza la conexión.

Una primera ojeada puede parecer insuficiente, pero son necesarias solo unas pocas llamadas a procedimientos y comparaciones de cadena por conexión puesto que no es probable que haya docenas de drivers se carguen concurrentemente. El siguiente código es un ejemplo de todo lo necesario normalmente para establecer una conexión con un driver puente JDBC-ODBC:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); //loads the driver
String url = "jdbc:odbc:fred";
DriverManager.getConnection(url, "userID", "passwd");
```

## SERVLETS y JDBC

*¿Qué es un servlet?*

Un servlet de forma intuitiva se puede definir como un programa independiente de plataforma que aporta la misma funcionalidad a la programación en el lado del servidor que tradicionalmente han realizado la interfaz CGI. Con respecto a esta tecnología aporta numerosas ventajas que citaremos a continuación:

- Independencia de la plataforma. (La tan anhelada premisa del "write once run everywhere" aun no totalmente conseguida). Esto proporciona un menor esfuerzo de codificación con respecto a soluciones dependientes del servidor web y de la plataforma como ISAPI o NSAPI.
- Ejecución en paralelo de múltiples peticiones por una sola instancia del servlet. Tradicionalmente en los programas CGI se ejecuta un proceso distinto para cada petición lo que conlleva una gradual degradación del rendimiento y una necesidad de recursos muy elevada.

En un servlet todas las peticiones se atienden en el mismo proceso por distintos hilos y una vez que se ha cargado el servlet este permanece en memoria hasta que se reinicie el servidor o hasta que se le diga lo contrario con lo cual las subsiguientes peticiones son más rápidas al encontrarse el programa ya cargado en memoria.

- Un servlet puede ejecutarse (inciso en esto *puede* no es necesario) en una *sandbox* o recinto de seguridad parecido al modelo que se sigue con los applets. Debido a esto pueden colocarse servlets en servidores dedicados a hosting sin que la empresa tema por la integridad del servidor y la seguridad de las aplicaciones.

### CARACTERÍSTICAS DE LOS *SERVLETS*

Además de las características indicadas en el apartado anterior, los *servlets* tienen las siguientes características:

1. Son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en *Java*, el servidor puede estar escrito en cualquier lenguaje de programación, obteniéndose exactamente el mismo resultado que si lo estuviera en *Java*.
2. Los *servlets* pueden llamar a otros *servlets*, e incluso a métodos concretos de otros *servlets*. De esta forma se puede distribuir de forma más eficiente el trabajo a realizar. Por ejemplo, se podría tener un *servlet* encargado de la interacción con los clientes y que llamara a otro *servlet* para que a su vez se encargara de la comunicación con una base de datos. De igual forma, los *servlets* permiten *redireccionar* peticiones de servicios a otros *servlets* (en la misma máquina o en una máquina remota).
3. Los *servlets* pueden obtener fácilmente información acerca del *cliente* (la permitida por el protocolo *HTTP*), tal como su dirección *IP*, el *puerto* que se utiliza en la llamada, el método utilizado (*GET*, *POST*, ...), etc.
4. Permiten además la utilización de *cookies* y *sesiones*, de forma que se puede guardar información específica acerca de un usuario determinado, personalizando de esta forma la interacción cliente-servidor. Una clara aplicación es *mantener la sesión* con un cliente.
5. Los *servlets* pueden actuar como enlace entre el cliente y una o varias *bases de datos* en arquitecturas *cliente-servidor de 3 capas* (si la base de datos está en un servidor distinto).
6. Asimismo, pueden realizar tareas de *proxy* para un *applet*. Debido a las restricciones de seguridad, un *applet* no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el *servlet* sí puede hacerlo de su parte.
7. Al igual que los *programas CGI*, los *servlets* permiten la generación dinámica de código *HTML* dentro de una propia página *HTML*. Así, pueden emplearse *servlets* para la creación de contadores, banners, etc.

Estructura de un servlet

El API Servlet consiste básicamente en dos paquetes:

- **javax.servlet** En este paquete se definen 6 interfaces y 3 clases para la implementación de servlets genéricos, sin especificación de protocolo. Hoy en día no tienen utilidad práctica más que para servir de base en la jerarquía de clases de los servlets. Conforme pase el



tiempo se supone que constituirán la base para la implementación de otros protocolos distintos de http.

· **javax.servlet.http** Ofrece la implementación específica de servlets para el protocolo http. En estos paquetes se definen todas las clases e interfaces necesarias para la escritura de applets. De hecho cuando se usen los servlets (y hoy en día no hay otra utilidad) para gestionar conexiones http usaremos las clases del paquete **javax.servlet.http**. El ciclo de ejecución de un servlet es análogo al de un applet con ligeras diferencias. Inicialmente el servlet debe extender a la clase **HttpServlet**:

```
import javax.servlet;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class MiServlet extends HttpServlet{  
....  
}
```

Para dotar de funcionalidad a un servlet se han de redefinir una serie de métodos que guardan una analogía con los métodos de funcionamiento de un applet(**init()**,**start()**,**stop()**,**destroy()**).

#### **public void init(ServletConfig config)**

Cada vez que se inicia el servlet el servidor web llama a este método pasando un parámetro de la clase **ServletConfig** que guarda información de la configuración del servlet y del contexto del servidor web en el que se ejecuta. A través de **ServletConfig** se accede a los parámetros de inicialización del servlet que se establecieron al configurar el servlet y a través de la interfaz **ServletContext** (obtenido a partir del método **getServletContext()** de **ServletConfig**) se accede a la información del servidor web.

#### **public void destroy()**

Este método es análogo al método **init()** solo que será llamado por el servidor web cuando el servlet esta a punto de ser descargado de memoria(épito e insisto: no cuando termina una petición).En este método se han de realizar las tareas necesarias para conseguir una finalización apropiada como cerrar archivos y flujos de entrada de salida externos a la petición, cerrar conexiones persistentes a bases de datos...etc.. Un punto importante es que se puede llamar a este método cuando todavía esta ejecutándose alguna petición por lo que podría producirse un fallo del sistema y una inconsistencia de datos tanto en archivos como en BD. Por eso debe retrasarse la desaparición del servlet hasta que todas las peticiones hayan sido concluidas(con respecto a la peticiones que lleguen en ese intervalo el servidor web sencillamente las desestimaré).

#### **public void service(HttpServletRequest request,**

**HttpServletResponse response) throws ServletException, IOException.**

En este método se encuentra la mayor parte de la funcionalidad del servlet. Cada vez que se realice una petición se llamará a este método pasándole dos parámetros que nos permite obtener información de la petición y un flujo de salida para escribir la respuesta. Análogamente tenemos otra serie de métodos que realizan la implementación de respuesta a métodos de comunicación del protocolo http 1.1 como son GET y POST. Estos son respectivamente:

```
public void doGet(HttpServletRequest request,HttpServletResponse  
response)  
public void doPost(HttpServletRequest request,HttpServletResponse  
response)
```

Los dos parámetros que recibe **service()** son esenciales para el funcionamiento del servlet por lo que pasaremos a verlos con mas profundidad: Los dos parámetros que recibe **service()** son **HttpServletRequest** y **HttpServletResponse**

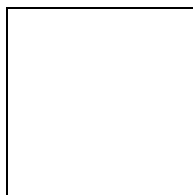
### **HttpServletRequest**

Esta interfaz derivada de **ServletRequest** proporciona los métodos para recuperar la información de la petición del usuario así como del propio usuario. Señalaremos los mas importantes:

### **EL SERVLET API 2.0**

El *Java Servlet API 2.0* es una extensión al *API* de *Java 1.1.x*, y también de *Java 2*. Contiene los packages *javax.servlet* y *javax.servlet.http*. El *API* proporciona soporte en cuatro áreas:

1. Control del ciclo de vida de un *servlet*: clase *GenericServlet*
2. Acceso al contexto del *servlet* (*servlet context*)
3. Clases de utilidades
4. Clases de soporte específicas para *HTTP*: clase *HttpServlet*



### **EL CICLO DE VIDA DE UN SERVLET: CLASE GENERICSERVLET**

Como se ha dicho en el Apartado 4.1, página 15, la clase *GenericServlet* es una clase abstract porque declara el método *service()* como *abstract*. Aunque los *servlets* desarrollados en conexión con páginas web suelen derivar de la clase *HttpServlet*, puede

ser útil estudiar el ciclo de vida de un *servlet* en relación con los métodos de la clase *GenericServlet*. Además, la clase *HttpServlet* hereda los métodos de *GenericServlet* y define el método *service()*. Los *servlets* se ejecutan en el *servidor HTTP* como parte integrante del propio proceso del servidor. Por este motivo, el *servidor HTTP* es el responsable de la inicialización, llamada y destrucción de cada objeto de un *servlet*.

Un *servidor web* se comunica con un *servlet* mediante la los métodos de la interface *javax.servlet.Servlet*. Esta interface está constituida básicamente por tres métodos principales: *init()*, *destroy()* y *service()* y por dos métodos algo menos importantes: *getServletConfig()*, *getServletInfo()*.

## CLASE HTTPSERVLET: SOPORTE ESPECÍFICO PARA EL PROTOCOLO http

Los *servlets* que utilizan el protocolo *HTTP* son los más comunes. Por este motivo, *Sun* ha incluido un package específico para estos *servlets* en su *JSDK: javax.servlet.http*. Antes de estudiar dicho *package* en profundidad, se va a hacer una pequeña referencia al protocolo *HTTP*. *HTTP* son las siglas de *HyperText Transfer Protocol*, que es un protocolo mediante el cual los browser y los servidores puedan comunicarse entre sí, mediante la utilización de una serie de *métodos: GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT* y *OPTIONS*. Para la mayoría de las aplicaciones, bastará con conocer los tres primeros.

### Método GET: codificación de URLs

El método *HTTP GET* solicita *información* a un *servidor web*. Esta información puede ser un fichero, el resultado de un programa ejecutado en el servidor (como un *servlet*, un *programa CGI*, ...), etc. En la mayoría de los servidores web los *servlets* son accedidos mediante un *URL* que comienza por */servlet/*.

El *URL* de esta petición *GET* llama a un *servlet* llamado *MiServlet* y contiene dos parámetros, *nombre* y *apellido*. Cada parámetro es un par que sigue el formato *clave=valor*. Los parámetros se especifican poniendo un *signo de interrogación* (?) tras el nombre del *servlet*. Además, los distintos parámetros están separados entre sí por el símbolo *ampersand* (&). Obsérvese que la secuencia de caracteres *%20* aparece dos veces en el apellido. Es una forma de decir que hay un *espacio* entre “Lopez” y “de”, y otro entre “de” y “Romera”. Esto ocurre por la forma en que se codifican los *URL* en el protocolo *HTTP*. Sucede lo mismo con otros símbolos como las tildes u otros caracteres especiales. Esta codificación sigue el esquema:

En cualquier caso, los programadores de *servlets* no deben preocuparse en principio por este problema, ya que la clase *HttpServletRequest* se encarga de la decodificación, de forma que los valores de los parámetros sean accesibles mediante el método *getParameter(String parametro)* de dicha clase. Sin embargo, hay que tener cuidado con algunos caracteres a la hora de incluirlos en un *URL*, en concreto con aquellos caracteres no pertenecientes al código ASCII y con aquellos que tienen un significado concreto para el

protocolo **HTTP**. Más en concreto, se pueden citar los siguientes caracteres especiales y su secuencia o código equivalente:

" (%22), # (%23), % (%25), & (%26), + (%2B), , (%2C), / (%2F), : (%3A), < (%3C), = (%3D), > (%3E), ? (%3F) y @ (%40).

Adicionalmente, **Java** proporciona la posibilidad de codificar un **URL** de forma que cumpla con las anteriores restricciones. Para ello, se puede utilizar la clase **URLEncoder**, que se encuentra incluida en el package **java.net**, que es un package estándar de **Java**. Dicha clase tiene un único método, **String encode(String)**, que se encarga de codificar el **String** que recibe como argumento, devolviendo otro **String** con el **URL** debidamente codificado. Así, considérese el siguiente ejemplo:

```
import java.net.*;
```

```
public class Codificar {  
    public static void main(String argv[]) {  
        String URLcodificada=URLEncoder.encode("/servlet/MiServlet?nombre=Antonio"+  
        "&Apellido=López de Romera");  
        System.out.println(URLcodificada);  
    }  
}
```

que cuando es ejecutado tiene como resultado la siguiente secuencia de caracteres:

```
%2Fservlet%2FMiServlet%3Fnombre%3DAntonio%26Apellido%3DL%20pez+de+Ro  
mera
```

Las peticiones **HTTP GET** tienen una limitación importante (recuérdese que transmiten la información a través de las variables de entorno del sistema operativo) y es un límite en la cantidad de caracteres que pueden aceptar en el **URL**. Si se envían los datos de un formulario muy extenso mediante **HTTP GET** pueden producirse errores por este motivo, por lo que habría que utilizar el método **HTTP POST**.

Se suele decir que el método **GET** es *seguro* e *idempotente*:

**Seguro**, porque no tiene ningún efecto secundario del cual pueda considerarse al usuario responsable del mismo. Es decir, por ejemplo, una llamada del método **GET** no debe ser capaz en teoría de alterar una base de datos. **GET** debería servir únicamente para obtener información. **Idempotente**, porque puede ser llamado tantas veces como se quiera de una forma segura. Es como si **GET** fuera algo así como *ver pero no tocar*.

### **Método HEAD: información de ficheros**

Este método es similar al anterior. La petición del cliente tiene la misma forma que en el método **GET**, con la salvedad de que en lugar de **GET** se utiliza **HEAD**. En este caso el

servidor responde a dicha petición enviando únicamente *información acerca del fichero*, y no el fichero en sí.

El método **HEAD** se suele utilizar frecuentemente para comprobar lo siguiente:

- La *fecha de modificación* de un documento presente en el servidor.
- El *tamaño del documento* antes de su descarga, de forma que el browser pueda presentar información acerca del progreso de descarga.
- El *tipo de servidor*.
- El *tipo de documento* solicitado, de forma que el cliente pueda saber si es capaz de soportarlo.

El método **HEAD**, al igual que **GET**, es *seguro e idempotente*.

### Método POST: el más utilizado

El método **HTTP POST** permite al cliente *enviar información al servidor*. Se debe utilizar en lugar de **GET** en aquellos casos que requieran transferir una cantidad importante de datos (formularios).

El método **POST** no tiene la limitación de **GET** en cuanto a volumen de información transferida, pues ésta no va incluida en el **URL** de la petición, sino que viaja encapsulada en un *input stream* que llega al *servlet* a través de la entrada estándar.

El encabezamiento y el contenido (en **negrita**) de una petición **POST** tiene la siguiente forma:

```
POST /servlet/MiServlet HTTP/1.1
User-Agent: Mozilla/4.5 (
compatible;
MSIE 4.01;
Windows NT)
Host: www.MiServidor.com
Accept: image/gif, image/x-bitmap, image/jpeg, image/jpeg, */
Content-type: application/x-www-form-urlencoded
Content-length: 39
nombre=Antonio&Apellido=Lopez%20de%20Romera
```

Nótese la existencia de una *línea en blanco* entre el encabezamiento (*header*) y el comienzo de la información extendida. Esta línea en blanco indica el final del *header*.

A diferencia de los anteriores métodos, **POST** *no* es ni *seguro* ni *idempotente*, y por tanto es conveniente su utilización en aquellas aplicaciones que requieran operaciones más complejas que las de sólo-lectura, como por ejemplo modificar bases de datos, etc.

## ANEXO 6

Mapas  
DF1.MAP









# Referencias y bibliografía

## Sistemas de Información Geográfica

- Stillwell, S. G. S. (1999) Geographical Information and Planning, Springer, Italia  
O.  
Stewart, F. M. W. (2000) Spatial Models and GIS, Taylor & Francis, Londres  
Joseph, K. B. (1997) Beyond Mapping Concepts, Algorithms, and Issues in GIS,  
Gis World Books, Fort Collins  
Paul, A. L. (1998) Geocomputation a Primer, Wiley, Ontario  
Korte George (2000) The GIS Book  
MapInfo (1999) MapXtreme reference guide  
Soukup, R (1999) A fondo MS SQL Server 7.0, Mc. Graw Hill, Mexico

## Java

- Deitel, D. (1998) Como programar en Java, Pearson, México  
Bruce Eckel (2000) Thinking in Java  
Hunter Jason (2001) Java Servlet Programming  
Reese George (2000) Database Programming with JDBC and Java 2<sup>nd</sup> Edition  
Jaworski Jamie (1999) Java 1.2 Al descubierto

