

623
291.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

DISEÑO E IMPLEMENTACION DE UN SISTEMA
PARA LA CONSULTA DE USO DE SUELO CON
UNA BASE DE DATOS MULTIMODELO Y UN
LENGUAJE DE CUARTA GENERACION

T E S I S
QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION
P R E S E N T A :
ESTEBAN LORA HERNANDEZ

DIRECTOR DE TESIS :

ING. JUAN JOSE CARREON GRANADOS

MEXICO, D. F.

1997



TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A mis padres, por todo su apoyo
y por darme la oportunidad de
tener educación.**

**A todas las personas que de
alguna forma me ayudaron
durante mis estudios**

**Diseño e Implementación de un
Sistema para la Consulta de Uso
de Suelo con una Base de Datos
Multimodelo y un Lenguaje de
Cuarta Generación.**

ESTEBAN LORA HERNÁNDEZ

Introducción

La idea de este trabajo es mostrar los aspectos generales del software necesario para la implementación del sistema presentado y cómo pueden utilizarse diversos tipos de herramientas para construir sistemas de información geográfica para la solución de un problema real, aprovechando los recursos existentes e integrándose con éstos, incluyendo la información, de tal forma que ésta se pueda explotar de una manera distinta y se pueda aprovechar mejor. También se pretende mostrar la importancia de emplear una metodología para la creación de aplicaciones aún cuando se utilicen herramientas avanzadas de programación, lo cual muchas veces se descuida y trae como resultado sistemas que no cumplen con los requerimientos reales del negocio o usuario, son difíciles de mantener o modificar, y en ocasiones, son más costosos en tiempo y dinero que las aplicaciones a las cuales reemplazarán, a pesar de implementarse con nueva tecnología.

Índice

	Introducción	i
Capítulo 1	Antecedentes Históricos	1
	Introducción.....	2
	Evolución de la Computación.....	3
	Proceso de datos.....	15
	Hardware.....	20
	Software.....	23
Capítulo 2	Conceptos de Bases de Datos	28
	Introducción.....	29
	Sistemas para manejo de archivos.....	30
	Modelo Jerárquico.....	32
	Modelo de Red.....	35
	Modelo Relacional.....	38
Capítulo 3	Conceptos de Lenguajes de Cuarta Generación	44
	Introducción.....	45
	Mercado de los 4GL.....	46
	Funcionalidad.....	49
	Lenguajes no procedurales.....	51
	El modelo relacional y los 4GL.....	53
	Productos CASE.....	55
	Orientación a Objetos.....	57
	Productividad.....	59
	Performance.....	64
Capítulo 4	Tecnología Cliente/Servidor	67
	Introducción.....	68
	Orígenes de la tecnología cliente/servidor.....	69
	Redes de área local.....	72
	SQL.....	74
	GUI.....	75
	Administración de la red.....	76
	Rendimiento.....	79
	Downsizing.....	80
	Desarrollo de aplicaciones.....	84
	OLTP.....	84
	Seguridad.....	86
	RPC.....	87
	Arquitecturas.....	89

Capítulo 5	Conceptos de Cartografía y Sistemas de Información Geográfica	91
	Introducción	92
	Definiciones	93
	Mapas	96
	Escala y Generalización	98
	Contenido del mapa	102
	Proyecciones	105
	Levantamiento de Datos	111
Capítulo 6	Caso Práctico	115
	Introducción	116
	Problemática	117
	Arquitectura	119
	Recursos de Hardware	120
	Recursos de Software	121
	Identificación de Requerimientos	133
	Metodología Empleada	134
	Proceso de creación del sistema	138
Capítulo 7	Presente y Futuro	149
	Introducción	150
	Perspectiva	151
	Cambio de Paradigma	156
	Tendencias en Software	170
	Tendencias en Hardware	179
	Cambios Propuestos al Sistema	183
Capítulo 8	Conclusiones	184
Capítulo 9	Bibliografía	186
Capítulo 10	Glosario	189
Anexo A	Documentación de Usuario	199
Anexo B	Documentación Técnica	209
Anexo C	Catálogos	245

Antecedentes Históricos

Supongamos por un momento que la industria automovilística se hubiera desarrollado al mismo ritmo que las computadoras y en el mismo periodo. ¿Que tan baratos y eficientes serían los modelos actuales?. Podríamos comprar un Rolls-Royce por \$2.75 dls; rendiría tres millones de millas por galón y sería tan poderoso como para remolcar al Queen Elizabeth II. En cuanto a tamaño, se podrían tener media docena en la cabeza de un alfiler.

Christopher Evans
The Micro Millennium

1.1 Introducción

El presente capítulo tiene como objeto el presentar antecedentes y tendencias tanto en software como en hardware con la finalidad de poder entender y establecer estrategias en cuanto a una arquitectura de sistemas teniendo en consideración los avances tecnológicos así como las ventajas en lo que respecta al manejo de la información y optimización de recursos.

Se presenta un breve recorrido histórico con el fin de mostrar los esfuerzos y avances que ha logrado el hombre en cuestiones de cómputo, que hoy día están relacionadas y son prácticamente indispensables en todas las ramas del conocimiento humano.

1.1. Evolución de la Computación

Siempre que el hombre ha necesitado manejar información y realizar cálculos ha trabajado en crear un dispositivo de trabajo exacto y simple de usar. A lo largo de siglos, dos herramientas se usaron en todo el mundo: El ábaco y la regla de cálculo. Su extinción llegó rápidamente con la revolución en cómputo de los 70s.

Primeras Herramientas de Cómputo

El Ábaco

La historia de la computación comienza miles de años atrás con el ábaco, la herramienta de cómputo más antigua que se conoce. Fue usada por los chinos desde el año 600 a.C. y en algunos lugares sigue siendo utilizado con gran maestría.

Está basado en pequeñas esferas o circunferencias que representan la unidad numérica de notación (unidad, decena, centena, millar), colocadas en alambres o hilos dentro de un marco. Las cantidades se forman al mover estas esferas o contadores de tal forma que se coloquen al centro o a la orilla del marco (de acuerdo al tipo de ábaco). Dependiendo de la posición de éstas se contabiliza o no, formando una cantidad representada en forma similar al sistema decimal que utilizamos actualmente.

Huesos de Napier

En 1615 el matemático escocés John Napier inventó una herramienta de cómputo que facilitaba las operaciones de multiplicación y división. Una serie de varitas, que se conocieron como los huesos de Napier, eran marcadas de una manera similar a las tablas de multiplicar que conocemos actualmente. Este aparato reducía las operaciones de multiplicación y división a simples sumas y restas.

En 1620, utilizando los principios de Napier, Edmund Gunter desarrolló la primera regla de cálculo. Desde entonces la regla de cálculo fue utilizada ampliamente por científicos, matemáticos y gente de negocios para realizar cálculos rápidamente. Después de más 350 años de uso, la regla de cálculo se extinguió de la noche a la mañana con la aparición a principios de los 70s de la calculadora electrónica.

Máquina Aritmética de Pascal

En 1642 el matemático y filósofo francés Blaise Pascal, a los 19 años de edad, diseñó un dispositivo de cálculo para realizar operaciones aritméticas tediosas. Esta máquina fue construida utilizando varias ruedas conectadas ingeniosamente entre sí; cada rueda contenía los dígitos del 0 al 9. Mediante engranes se lograba que cada vez que una rueda giraba una vuelta completa, la rueda a la izquierda giraba una décima de vuelta. Diez rotaciones de esta rueda causaban que la otra girara una vuelta, exactamente como funcionan los odómetros en los automóviles. Pascal anunció su máquina al público en 1645, pero no tuvo mucho éxito ya que era muy cara y no era sencilla de operar.

Máquina calculadora por pasos de Leibniz

La máquina de Pascal era capaz de realizar suma y resta, pero requería mucho esfuerzo manual para multiplicación y división. En 1673 el filósofo y matemático alemán Gottfried Leibniz modifica la máquina de Pascal de tal forma que pudiera realizar multiplicación y división directamente. La máquina de Leibniz representa un avance en el desarrollo de calculadoras automáticas.

Esta máquina estaba basada en un carro móvil que operaba sobre ruedas. La multiplicación se implementaba como una serie de sumas realizadas automáticamente por la máquina y la división se lograba con una serie de restas. La máquina entonces realizaba cálculos complejos en base a operaciones básicas, que es como las computadoras actuales funcionan. La gran demanda del invento de Leibniz se debió al empleo de la máquina en el cálculo de tablas para funciones matemáticas. En el siglo XVII el realizar manualmente el cálculo de estas tablas significaba el trabajo de toda una vida.

La máquina para tejer de Jacquard

Una de las mayores contribuciones a los métodos de cómputo proviene de una fuente poco usual: la industria textil. En 1801 Joseph Marie Jacquard desarrolló un dispositivo para máquinas de tejer que utilizaba tarjetas perforadas para "programar" patrones específicos. El mismo patrón podía ser repetido muchas veces de tal forma que era posible hacer copias exactas de tapetes y alfombras. En Francia este invento estuvo disponible en 1805. Siete años después, 11,000 máquinas tejedoras funcionaban con este dispositivo. Durante el siglo XIX muchos desarrollos

importantes, todavía evidentes en las computadoras actuales, estuvieron inspirados en el éxito de las tarjetas perforadas de Jacquard.

Máquina analítica de Babbage

Charles Babbage, matemático inglés, propuso la construcción de una máquina de diferencias cuando todavía era estudiante. Esta máquina calculaba o imprimía automáticamente tablas matemáticas. Su construcción comenzó en 1830 patrocinada por el gobierno inglés. Conforme se construía la máquina de diferencias, Babbage desarrolló la idea de una máquina aún más ambiciosa, a la que llamó la máquina analítica. Abandonó el primer proyecto en 1834 para trabajar en su nueva idea, la cual era capaz de almacenar 1000 números de 50 dígitos decimales cada uno. Los números se almacenarían en ruedas de 10 posiciones cada una. Tarjetas perforadas similares a las de Jacquard se utilizarían para guardar las secuencias de instrucciones a realizar y los datos a capturarse en la máquina, la cual estaba compuesta de cuatro unidades:

1. Almacenamiento de datos y resultados de cálculos
2. Mecanismo central para realizar operaciones matemáticas
3. Un sistema basado en engranes y palancas para mover los datos entre los dos componentes anteriores.
4. Unidad de entrada/salida para leer datos y desplegar resultados.

La importancia del esquema anterior es el ver una organización similar a los sistemas actuales. Babbage realmente estaba cien años adelantado a su época cuando propuso esta máquina.

Babbage nunca terminó este proyecto. La tecnología de la época era inadecuada para producir los componentes necesarios con la precisión requerida. Además no había necesidad real de una máquina tan poderosa, ya que los dispositivos existentes eran más que capaces de satisfacer las necesidades de cómputo de finales del siglo XIX. Como consecuencia, el gobierno británico suspendió el apoyo a Babbage en 1842.

Babbage es recordado como el padre de la computación gracias a la máquina analítica. De haber sido construida, esta hubiera sido la primera computadora de propósito general totalmente automática. Otros siguieron el trabajo de Babbage; en 1855 George Scheutz, un impresor suizo, construyó un modelo funcional de la máquina de diferencias y en 1889 el hijo de Babbage construyó un modelo funcional de una parte de la

máquina analítica. Estas máquinas influyeron grandemente en los futuros inventores, lo cual eventualmente daría lugar a la computadora digital.

El primer programador

En 1842 Augusta Ada Byron, Condesa de Lovelace e hija del poeta Lord Byron, hizo la traducción del francés al inglés de un trabajo sobre la máquina analítica de Babbage. Al realizar la traducción hizo muchas notas y en cierto momento incluyó una secuencia detallada de instrucciones para realizar algunos cálculos complejos, la cual se considera como el primer programa en el mundo. El lenguaje ADA fue llamado así en honor a esta persona.

Lógica Booleana

Las máquinas calculadoras del siglo XIX empleaban números decimales. Los números binarios, o base dos, no se usaron sino hasta el siglo XX. La utilización del sistema binario facilitó el trabajo realizado por el matemático inglés George Boole. En 1854 publicó los principios de lo que hoy conocemos como lógica booleana, matemática de variables cuyos valores solo pueden ser cierto o falso. Transcurrió más de un siglo antes de que se viera la importancia de su trabajo en el diseño de sistemas.

Los investigadores del siglo XX se dieron cuenta que era difícil construir un elemento eléctrico que representara un dígito decimal (0 a 9) en contraste con la facilidad de contar con elementos electrónicos confiables para representar dígitos binarios (0 y 1). La lógica de Boole era la adecuada para este sistema, ya que describía completamente las operaciones necesarias para implementar y controlar elementos de dos valores en circuitos. Lo anterior también sirvió para que se extendiera uno de los desarrollos más significativos del siglo XX, la computadora digital.

Máquina de Hollerith

La constitución de los Estados Unidos requiere que se realice un censo cada 10 años. Este censo originalmente implicaba un simple conteo de personas, pero a lo largo de los años se ha ampliado para incluir datos sociales, étnicos y económicos.

El censo de 1880 fue analizado manualmente, lo cual tardó siete años y medio. Se estimó que si el censo de 1890 se realizaba de la misma forma

se tendrían los resultados hasta 1902, dos años después del censo de 1900.

En 1879, John Billings, empleado de la oficina de censos, sugirió la utilización de tarjetas perforadas para guardar los datos así como el uso de equipo mecánico para procesar dichas tarjetas. Herman Hollerith, colega de Billing, comenzó a diseñar el equipo. Hollerith dejó la oficina de censos en 1882 para trabajar en el Instituto Tecnológico de Massachusetts (MIT) como profesor. Continuó trabajando en sus máquinas y un año después fue empleado de la oficina de patentes, que el 31 de Marzo de 1884 le otorgó la primera patente de una máquina para proceso de datos. En 1889 las máquinas de Hollerith estaban ampliamente probadas y fueron seleccionadas por el gobierno para el censo de 1890, cuyo proceso tardó dos años y medio en realizarse. Se utilizaba una máquina perforadora que hacía agujeros en puntos específicos de una tarjeta para almacenar datos sobre las personas. La máquina tabuladora colocaba puntas en las tarjetas, las cuales al pasar por una perforación tocaban un depósito de mercurio, lo cual cerraba un circuito eléctrico que incrementaba en una unidad un simple contador mecánico.

Hollerith inició en 1896 la Tabulating Machine Company. Después de varios cambios se convirtió en la Computing-Tabulating-Recording Company. En 1924, a cargo de Thomas J. Watson, Sr. la compañía cambió de nombre a International Business Machines Corporation (IBM).

Máquina perforadora simultánea de Powers

El sucesor de Hollerith en la oficina de censos fue James Powers, quien trabajó en mejorar el equipo de Hollerith. La contribución de Powers fue la máquina perforadora simultánea. Este dispositivo permitía a un operador después de capturar datos, el checar y corregir éstos antes de perforar tarjetas, ya que con los primeros equipos basados en tarjetas se tenía que volver a perforar otra tarjeta si se cometía un error, lo cual era un proceso costoso en tiempo y dinero. La oficina de censos compró a Powers 300 máquinas para el censo de 1910. Powers dejó la oficina de censos en 1911 para formar la Powers Accounting Machine Company, que fue creciendo hasta ser Sperry Rand, fabricantes de las computadoras UNIVAC.

El comienzo de la era moderna de las computadoras

Las máquinas basadas en tarjetas perforadas desarrolladas a finales del siglo XIX y principios del XX probaron su utilidad en negocios y en el gobierno, pero estas máquinas no eran verdaderas computadoras. Hasta finales de los 30s empezó a trabajarse en el desarrollo de lo que hoy se considera computadoras modernas. Este trabajo se vio acelerado por las presiones de la Segunda Guerra Mundial. El gobierno vislumbró el potencial de estas máquinas en aplicaciones bélicas y apoyó totalmente su desarrollo.

Aiken y la Harvard Mark I

El equipo basado en tarjetas perforadas simplemente era muy lento para realizar los cálculos masivos requeridos en muchos problemas matemáticos y científicos. En 1939 Howard Aiken, de la Universidad de Harvard, empezó a trabajar en el desarrollo de una máquina que realizara estos cálculos con mayor velocidad. IBM financió a Aiken y en 1944 se terminó la Harvard Mark I. La mayoría de los historiadores se refieren a esta fecha como el inicio de la era de la computación moderna.

La Mark I estuvo en uso 10 años y representó la primera realización funcional de la máquina analítica de Babbage, un siglo después de que éste la concibiera. La Mark I fue la primera implementación exitosa de una computadora digital de propósito general y la primera computadora electromecánica.

La Mark I era una máquina enorme, medía 51 pies de largo, 8 pies de alto y 3 pies de ancho. Pesaba varias toneladas. La máquina contenía 3000 interruptores mecánicos, 750,000 componentes electrónicos y 500 millas de cable. Era controlada por cinta de papel perforado y guardaba los números en interruptores mecánicos. Utilizaba números decimales en lugar de binarios y sus cálculos eran exactos hasta los 23 dígitos. Podía realizar tres sumas por segundo, una multiplicación en seis segundos y una división en 12 segundos. Esta capacidad hacía de la Mark I la máquina más rápida que hasta ese entonces se había construido.

Atanasoff y la ABC

En 1942 John Vincent Atanasoff y Clifford Berry terminaron un proyecto de computadora basada en bulbos. La ABC (Atanasoff-Berry Computer) es reconocida como la primera computadora digital realmente electrónica, aunque muchos le dan el crédito a la ENIAC. Fue diseñada específicamente para resolver ecuaciones simultáneas y utilizaba el sistema binario.

Mauchly, Eckert y la ENIAC

Mientras Atanasoff trabajaba en la ABC conoció a John W. Mauchly, quien a partir de estas pláticas tuvo sus propias ideas sobre como construir una mejor computadora. Hizo equipo con J.Presper Eckert, Jr. y juntos construyeron la primera computadora electrónica de propósito general, la ENIAC (Electronic Numerical Integrator and Computer). Esta computadora fue financiada por la armada de los Estados Unidos debido a la necesidad de calcular con exactitud tablas balísticas, que aumentaron durante la Segunda Guerra Mundial. Sin embargo fue terminada en 1945, meses después de que la guerra había terminado, razón por la cual el gobierno utilizó la Mark I.

Von Neumann y el concepto de programa almacenado

En 1946 el matemático John von Neumann se unió al grupo de Mauchly y Eckert para trabajar en una versión mejorada de la ENIAC, la EDVAC (Electronic Discrete Variable Automatic Computer). Von Neumann propuso almacenar en memoria el programa a ejecutar y los datos a procesar. La ventaja era que la computadora procesaría el programa a velocidades electrónicas en vez de velocidades mecánicas (debido a las tarjetas perforadas). Este concepto se considera uno de los más importantes desarrollos en el campo de la computación del siglo XX.

Wilkes y la EDSAC

Mientras se desarrollaba la EDVAC, un equipo encabezado por Maurice Wilkes en la Universidad de Cambridge desarrolló la EDSAC (Electronic Delay Storage Automatic Calculator). EDSAC también incorporaba el concepto de programa almacenado. Debido a que la EDSAC comenzó a operar en 1949 (antes que la EDVAC) se la considera la primera computadora que tuvo capacidad de almacenar programas.

UNIVAC. La primera computadora comercial

En 1946 Eckert y Mauchly dejaron la Universidad de Pennsylvania para formar la Eckert-Mauchly Computer Corporation. Tres años después Remington Rand adquiere la compañía y las patentes de la ENIAC. Esta compañía se había creado con la intención de desarrollar sistemas de proceso de datos comerciales. En 1951 Rand instala el primero de estos sistemas en la Oficina de Censos. La computadora, llamada UNIVAC I todavía se utilizaba en 1963. Esta computadora gozó de gran publicidad al ser usada por la Columbia Broadcasting System (CBS) para pronosticar la elección de Dwight Eisenhower como presidente de los Estados Unidos en 1952. Utilizando la computadora fue posible predecir el triunfo de Eisenhower con sólo un conteo de aproximadamente 3% de los votos.

IBM

A principio de los 50s IBM ya era una compañía grande que se especializaba en máquinas de escribir, sumadoras y máquinas para proceso de tarjetas perforadas. IBM había sido dirigida por Thomas J. Watson durante tres décadas exitosamente. Thomas J. Watson, hijo, convence a su padre de que la época era propicia para entrar al campo de las computadoras. En 1953 la compañía introduce el modelo 701, enfocado a negocios, pero no es sino hasta que se ofrece el modelo 650 que IBM adquiere renombre en el campo. El sistema 650 fue el más popular en los 50s, lo cual hizo que la compañía creciera tanto en el terreno de la computación.

Generaciones de computadoras.

Desde principios de los 50s las capacidades de las computadoras han aumentado y el costo ha disminuido. Esta evolución ha sido posible por el avance continuo en la electrónica. Los grupos de computadoras asociados con cada descubrimiento significativo se han categorizado como generaciones de computadoras. Hasta la fecha, existen cuatro generaciones:

Primera Generación: 1951-1959

La industria de las computadoras inicia con la UNIVAC I en 1951. Los bulbos eran los componentes clave en las computadoras de esta época. Las computadoras basadas en esta tecnología eran mucho más rápidas que las electromecánicas, como era el caso de la Mark I, sin embargo tenían varias desventajas: Los bulbos eran estorbosos, consumían mucha electricidad, generaban gran cantidad de calor y eran susceptibles a fallas, lo cual hacía que tuvieran dificultad en funcionar confiablemente después de varias horas.

Las computadoras de esta generación típicamente procesaban algunos miles de instrucciones por segundo y tenían la capacidad de almacenar de 10,000 a 20,000 caracteres. Se probaron distintos tipos de memoria, pero el más importante fue el basado en material magnético, desarrollado en 1950 por Jay W. Forrester para la computadora Whirlwind I en el MIT. Estas piezas magnéticas eran partes de metal semejantes a una dona y tenían la capacidad de magnetizarse en una o dos direcciones (correspondiendo al 0 y 1 binario).

La computadora más exitosa de esta generación fue el modelo 650 de IBM, quien originalmente estimaba un mercado de 40 o 50 sistemas para este modelo y terminó vendiendo miles. La fortaleza del sistema 650 eran sus sólidas capacidades para entrada y salida de datos mediante tarjetas perforadas.

Segunda Generación: 1959-1964

En 1948, con la invención del transistor en los laboratorios Bell, era muy claro que los transistores rápidamente sustituirían a los bulbos. Los transistores eran más pequeños, más confiables, usaban menos energía y generaban poco calor. Adicionalmente, las computadoras basadas en transistores operaban mucho más rápido que aquellas basadas en bulbos. Para 1959 se ofrecían sistemas en los que se reemplazaban totalmente los bulbos con transistores. Las computadoras de la primera generación eran sistemas dedicados, es decir, sólo una persona podía usar la máquina a la vez. Los diseñadores de la segunda generación se dieron cuenta de que era necesario construir los sistemas de tal forma que varias personas pudieran utilizarlos simultáneamente. El concepto, llamado multiprogramación, obviamente fue exitoso. Al comienzo de los 60s varios grupos en la industria y en las universidades fueron pioneros en el tiempo compartido, método en el que decenas o cientos de usuarios accesan a la computadora mediante terminales tipo máquina de escribir.

Durante la segunda generación se estableció la importancia de las computadoras en los negocios. La computadora más exitosa de esta etapa fue la IBM 1401. En 1964 IBM había instalado más 1401s que la suma de todos los demás sistemas en el mundo.

Tercera Generación: 1964 - 1970

La tercera generación se origina con la aparición de los circuitos integrados en los que el equivalente a cientos o miles de transistores puede ser colocado en un pequeño chip de silicón. El uso de estos chips en las computadoras redujo dramáticamente el costo, tamaño y consumo de energía de los circuitos electrónicos.

El 7 de Abril de 1964 IBM anuncia uno de los más importantes sucesos en la historia de las computadoras. Había desarrollado el System/360, que era una familia de computadoras diseñada para reemplazar todos los sistemas anteriores. La serie 360 contenía varios modelos, desde una pequeña computadora orientada a negocios hasta una supercomputadora diseñada para uso científico. Los modelos eran compatibles entre sí, es decir, un programa que corriera en un equipo pequeño era ejecutable en un modelo más grande. Se introdujo el concepto de que cualquier empresa podía empezar con el modelo básico e ir creciendo conforme lo hicieran sus necesidades. La serie 360 fue la más vendida en los 60s.

En esta etapa Control Data refuerza su posición como fabricante de supercomputadoras con el modelo CDC 6600. Esta máquina era capaz de realizar tres millones de instrucciones por segundo, lo cual la hacía el equipo más poderoso entonces (1964). En 1969 Control Data introduce el modelo 7600, que realizaba 36 millones de operaciones por segundo.

La tecnología de circuitos integrados de esta generación hizo posible la producción de equipos pequeños pero poderosos, es decir, minicomputadoras. Uno de los más sobresalientes fue el PDP-8, fabricado por Digital Equipment Corporation (DEC) en 1965. El diseño de la PDP-8 la hacía extremadamente útil para el control de procesos. Miles de PDP-8 se utilizaron para el control de líneas de producción, laboratorios, refinerías e inclusive negocios pequeños. Este modelo estableció a DEC como líder en la fabricación de minicomputadoras. La PDP-8 fue la primera computadora comercial con un costo inferior a \$20,000 dls.

Cuarta Generación: 1970 en adelante.

El surgimiento de los circuitos LSI (Large Scale Integrated) con miles de transistores en un solo chip marcó el inicio de la cuarta generación. En 1970 IBM anuncia el reemplazo del modelo 360, el System/370. Este modelo utilizaba tecnología LSI, era capaz de ejecutar los programas de los 360s e introducía la memoria MOS (Metal Oxide Semiconductor). En estos años la industria ya era consciente de las generaciones de computadoras, razón por la cual al anunciarse el sistema 370 hubo críticas con respecto a si esta tecnología era lo suficientemente revolucionaria como para iniciar una nueva generación. Estos equipos se consideraban más bien una evolución que una revolución tecnológica. No fue sino hasta mediados de los 70s que se aceptó denominar a una computadora como de cuarta generación.

Una serie de decisiones legales cambiaron fuertemente el rumbo de la industria, ya que se le indicó a IBM que tenía que vender a cualquier persona los programas, sistemas de programación y documentos relacionados. Esto creó un ambiente intensamente competitivo, favorable para el desarrollo de compañías fabricantes de hardware compatible con IBM. Estas compañías, llamadas PCMs (plug compatible mainframers) eran capaces de construir hardware que competía con los sistemas IBM y utilizar software IBM para sus clientes.

Una de las empresas PCM más importantes fue fundada por Gene Amdahl, quien había sido diseñador del sistema 360. En 1970 dejó IBM y funda Amdahl Corporation. En 1975 instala el primer sistema 470 V/6, que era una máquina que costaba menos que una IBM 370, era más poderosa y utilizaba software IBM.

Durante la cuarta generación la industria de las minicomputadoras creció explosivamente. DEC utilizaba circuitos LSI en las series PDP-11, introducidas al mercado en 1970.

El campo de las supercomputadoras también creció. Seymour Cray, quien era el jefe de diseñadores en Control Data, deja la empresa en 1972 para formar Cray Research Inc. En 1976 introducen la CRAY-1, capaz de realizar 80 millones de instrucciones por segundo. Para 1982 la CRAY X-MP podía realizar 1 billón de instrucciones en un segundo.

Uno de los desarrollos más importantes de la cuarta generación fue la creación de la microcomputadora. El poder de cómputo que hace 30 años costaba miles de dólares ahora cuesta unos cuantos. La

microcomputadora ha cambiado profundamente a la industria y a la sociedad.

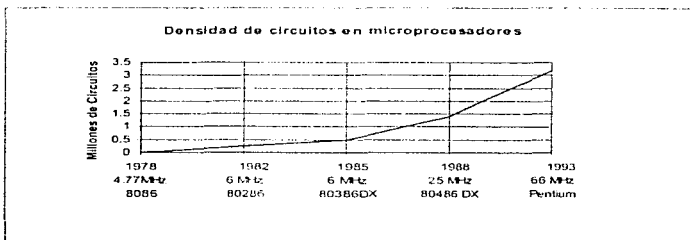
Se piensa que a pesar de que seguirá en aumento el nivel de integración de circuitos en un solo chip, la quinta generación llegará cuando existan avances revolucionarios en software que permitan tener máquinas realmente inteligentes.

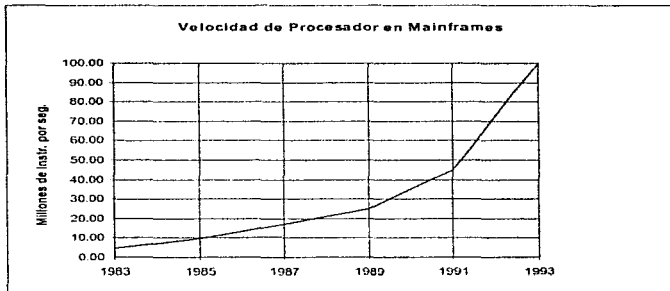
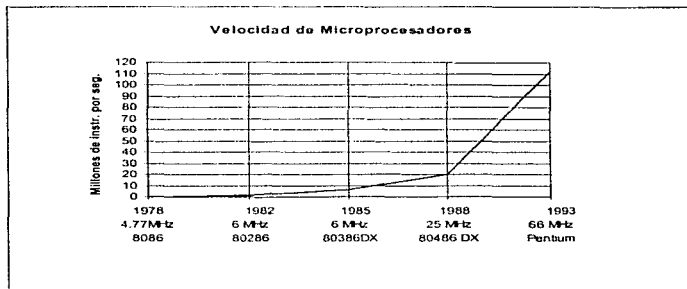
A continuación se presentan varias consideraciones importantes en lo que respecta a hardware y software. Es tal la velocidad con la cual la tecnología avanza que resulta casi imposible mantener referencias al día, ya que cada vez es menos el tiempo que tarda el hardware o software en volverse obsoleto.

1.2 Proceso de datos

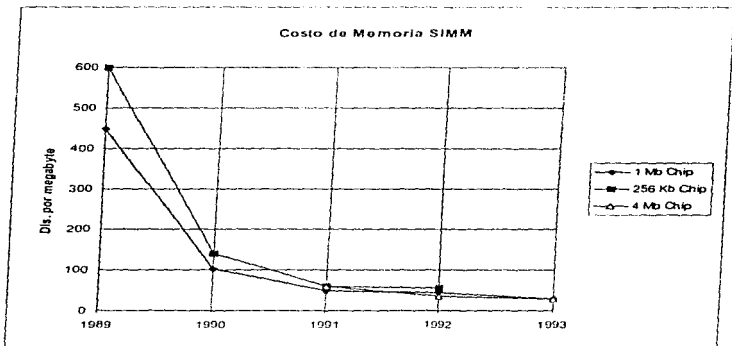
Hablar de sistemas de cómputo involucra hablar de cambio. Cambios en tecnología y en metodologías. Lo que conocemos como hardware ha pasado por varias revoluciones desde que las primeras computadoras aparecen. Los bulbos son reemplazados por semiconductores, interruptores mecánicos por transistores, que a su vez fueron sustituidos por circuitos integrados. La programación a nivel de bit evoluciona a ensambladores, compiladores y ahora lenguajes de cuarta generación y lenguajes visuales.

Los chips cada vez son más pequeños y más económicos. Actualmente un chip puede contener más de 4 millones de transistores. El poder de proceso de estos chips llega al nivel de decenas de centenas de MIPS (millones de instrucciones por segundo). En las gráficas se muestran gráficas con datos sobre la evolución en cuanto a densidad de microcircuitos, velocidad de CPU y velocidad de procesamiento en mainframes.





Los chips de memoria también han tenido un desarrollo similar. Los primeros mainframes a lo mucho tenían 16 KB. Actualmente un mainframe puede tener de 1 a 4 GB de memoria. La cantidad de memoria en un chip ha ido de 4 KB a 16 MB o más. Al mismo tiempo los costos se han reducido de miles de dólares por megabyte a unos cuantos dólares. En la siguiente gráfica se muestra la reducción del costo de memoria SIMM.



El ancho de banda en comunicaciones también se ha incrementado dramáticamente. Los costos han disminuido también grandemente. Debido a la competencia entre compañías telefónicas y de cable, se espera que la reducción en precios sea aún mayor. La utilización de fibra óptica no ha alcanzado todo su potencial; conforme evoluciona de una sola fuente de luz (un color) a un modo multicolor, ancho de banda adicional estará disponible sobre la fibra existente.

La tecnología celular contribuirá al llamado cómputo móvil. Los equipos continuarán disminuyendo de tamaño y peso, lo cual permite contemplar nuevas combinaciones o generaciones de equipo, como son las computadoras notebook, PDAs o palmtops con teléfono integrado. Compañías como Apple Computer y AT&T están trabajando activamente

este mercado. Aunque la interface física limita la reducción en tamaño de estos equipos, su papel es el de funcionar como elementos cooperativos o cliente/servidor funcionando como dispositivos externos de almacenamiento con la capacidad de conectarse a equipos PC en la oficina y la casa, al mismo tiempo que funcionan como computadoras independientes.

Todos estos cambios implican una mayor capacidad y a menor costo. Cada vez el poder de cómputo es más barato. Gráficos de alta resolución y manejo de voz son los elementos que consumen mayores recursos, aunque el incremento en poder de procesamiento no necesariamente implica ahorro, pero permitirá a los usuarios hacer más por la misma cantidad de dinero.

Conforme se expanden las capacidades de comunicación al tener mayor ancho de banda, el procesamiento cooperativo será cada vez mas viable. Siempre y cuando sea costeable mover grandes cantidades de datos entre plataformas será más común la aparición de bodegas de datos, con lo cual se tendrá información sobre demanda. Los usuarios finales tendrán la capacidad de aprovechar los recursos de estaciones de trabajo para convertir grandes cantidades de datos en información útil y de fácil manejo. Aunque existen varios puntos de vista al respecto, las poderosas estaciones de trabajo no harán obsoletos a los mainframes, sino que cambiarán su papel. De hecho la demanda por equipos mainframes aumentará una vez que las aplicaciones no aptas para estas plataformas sean procesadas en otras más adecuadas y el mainframe sea liberado de carga y utilizado para funciones mas apropiadas. La tendencia al "downsizing" cada vez será menor y los equipos mainframe serán utilizados como superservidores, proporcionando almacenamiento, seguridad y administración de red en forma centralizada.

Discos Ópticos

La capacidad de almacenar grandes cantidades de datos en discos ópticos incrementará el volumen de información almacenada. Aunque es difícil de medir, se estima que en promedio menos del 10% de datos de las empresas se encuentran en formatos de computadora. Además, pocas compañías tienen más de 2 terabytes en disco con datos, lo que significa que éstas tienen una demanda potencial de 20 TB para almacenar datos. Una vez más, será necesario contar con almacenamiento y recuperación de datos en una forma más centralizada, inclusive después de trasladar archivos a plataformas más pequeñas.

Compresión de Datos

Con el uso de multimedia los requerimientos de almacenamiento se incrementan grandemente. Archivos de video y sonido puede llegar a ocupar alrededor de 10 MB por segundo. Almacenar una película promedio de 2 horas en formato digital no comprimido requiere al menos de 70 GB. Conforme el mundo se transforma de análogo a digital, inclusive con el uso de técnicas avanzadas de compresión en tiempo real, cantidades inimaginables de almacenamiento serán necesarias.

Varias compañías actualmente investigan formas alternativas para guardar datos. La nanotecnología o habilidad para manipular materiales de una billonésima de metro en tamaño, podrá conducir eventualmente a contar con medios de almacenamiento de muy alta densidad.

La necesidad de almacenar grandes cantidades de datos ocasionará varios fenómenos como son el resurgimiento del mainframe como superservidor, a pesar de la imagen de "dinosaurio" que tienen. La arquitectura del mainframe permite almacenar, administrar y transferir grandes cantidades de bytes casi instantáneamente. La administración centralizada de esta información será realizada mediante un repositorio de datos. La transferencia de estos datos se hará utilizando líneas de comunicación de muy alta velocidad basadas en fibra óptica.

Gráficos.

Además del incremento en datos, también los requerimientos en manejo de gráficos se incrementan dramáticamente debido principalmente a la utilización de interfaces gráficas. La PC original soportaba un modo de color CGA con resolución de 320 (horizontal) y 200 (vertical) pixeles y 4 colores. Esto ha evolucionado a distintos niveles de SVGA a 1280x1024 con 256,000 colores como algo estándar. Ignorando el color, la densidad de información ha aumentado de modos de color de 64 Kb (kilobits) a 1310 Kb, es decir, 20 veces.

Sin embargo el video no es el único componente en la interface humano/máquina que evoluciona rápidamente. La interacción con la máquina ha evolucionado del teclado al "mouse", pluma o lápiz óptico y eventualmente las interfaces de voz. Esta interface expandida requiere poder de cómputo adicional, espacio de almacenamiento adicional y lógica de implementación más sofisticada.

1.3 Hardware

Actualmente no sólo se tiene que evaluar el escoger el proveedor adecuado, sino el tamaño y tipo necesario de hardware de acuerdo a las necesidades de negocio y a la estrategia de sistemas, ya sea centralizada, distribuida, downsizing, outsourcing o rightsizing.

Existen una serie de parámetros que hacen casi imposible el adoptar un método o estrategia definitiva al implantar un esquema. Equipo existente, localización de dicho equipo y cultura informática son algunas de las variables que deben considerarse.

No todas las aplicaciones son adecuadas para mainframes centralizados. No todas las aplicaciones son propicias para equipos PC. Existe, sin embargo una constante: El uso y control de los datos de cualquier empresa como la más alta prioridad. El diseño debe ser lo suficientemente eficiente para utilizar dichos datos sin poner en riesgo la integridad de los mismos. Esto requiere procedimientos, políticas y controles que sean flexibles para adaptarse a situaciones individuales. El balancear las necesidades de la organización y sus usuarios es un factor crítico en los ambientes actuales, aunque siempre tendrán prioridad las necesidades de la empresa, que es a quien pertenece la información. Lo anterior suena elemental pero en muchas ocasiones se pierde de vista este objetivo.

En la mayoría de los casos la aplicación determinará los requerimientos adecuados. Si el software seleccionado sólo está disponible en ciertas plataformas o sistemas operativos, entonces las opciones son muy limitadas. Es necesario conocer los requerimientos de software antes de decidir el proceso de implementación y la compra de hardware.

Si se trata de una reorganización, existe una combinación de centralización y descentralización a través de rightsizing y aplicaciones cliente/servidor. En este caso la información se identifica como corporativa, departamental y personal, por lo que los datos deben ser distribuidos de acuerdo a estas categorías. En todas las situaciones los datos corporativos deben permanecer bajo control centralizado. Lo anterior no significa que los datos deben estar en un sólo equipo central, sino que el acceso y actualización de la información corporativa debe tener un punto de control centralizado.

Después de decidir la estrategia para la aplicación se debe seleccionar el hardware apropiado para soportar dicha aplicación. El diseño y tamaño de ésta determinarán la categoría de PC, LAN, minicomputadora o

mainframe requerido. Otros factores como confiabilidad y disponibilidad deben influenciar el proceso de decisión. Hay que tener en cuenta que no todos los productos en una categoría ofrecen los mismos niveles de rendimiento, confiabilidad, disponibilidad y seguridad.

Equipos PC

Las computadoras personales tienden a tener muchos puntos de falla. Generalmente sólo se utiliza una fuente de energía, la cual en la mayoría de los casos no está filtrada. Sistemas operativos como DOS/Windows, Unix y OS/2 carecen de mecanismos para recuperación en caso de falla, aunque hay que reconocer que cada día mejoran. Los discos duros en las PC tienden a fallar en los momentos más inoportunos y es ignorar los procedimientos de respaldo comunes en otros ambientes. Por las razones anteriores es recomendable almacenar datos no críticos en una PC y utilizarlas como estaciones de trabajo para accesar datos corporativos en un esquema cliente/servidor, en donde dichos datos deben encontrarse en un ambiente más seguro y confiable.

Nuevas generaciones de sistemas operativos como Windows NT ofrecen distintas opciones en cuanto a tipo de hardware y facilidad de uso, aunque todavía no llegan a ser un estándar establecido para aplicaciones críticas.

LAN

Las redes LAN basadas en PC sufren prácticamente los mismos problemas que las PC, aunque los servidores generalmente tienen discos duros de mas capacidad, más confiables e inclusive llegan a contar con un arreglo de discos para asegurar disponibilidad. La administración de la LAN, siempre y cuando se realice de manera profesional, asegurará respaldos regulares. Con un diseño correcto en cuanto a topología, la falla de una computadora de la red no debe afectar el funcionamiento de las demás. Una solución de misión crítica bajo este esquema puede resultar apropiada siempre y cuando se sigan los procedimientos antes mencionados y se cuente con una fuente ininterrumpible de energía (UPS o nobreak). Sin embargo es necesario contar con un servidor de respaldo en caso de falla del equipo.

Minicomputadoras

Las minicomputadoras cada vez son mas sofisticadas. Un ejemplo es el sistema AS/400 de IBM que incorpora múltiples procesadores, múltiples fuentes de poder y múltiples arreglos de discos para garantizar el funcionamiento del sistema en caso de fallas. El sistema operativo de estos equipos también ha sido diseñado para funcionar bien en la mayoría de casos de error.

Los equipos basados en sistemas operativos unix mejoran cada vez más en cuanto a confiabilidad y facilidad de uso, por lo cual pueden llegar a ser una buena opción. El punto fuerte es la gran disponibilidad de hardware y software bajo estos ambientes, que es la base de los sistemas abiertos.

Mainframes

Los equipo mainframes todavia tienen características que los hacen una opción ideal para muchas aplicaciones. Confiabilidad, disponibilidad e integridad de datos son características básicas en estos ambientes. El gran espacio de almacenamiento y el rendimiento transaccional son puntos adicionales a favor. Algunas aplicaciones son tan grandes que actualmente no pueden funcionar en otra plataforma. Si se requiere control centralizado y acceso a terabytes de datos para cientos o miles de peticiones por segundo, un mainframe todavia es un equipo con la suficiente capacidad para realizar el trabajo.

Es por lo anterior que el papel del mainframe evolucionará. Aplicaciones grandes seguirán corriendo en estos equipos, aunque tendrán front-ends gráficos en equipos PC bajo esquemas cliente/servidor, en donde el mainframe será un superservidor.

1.4 Software

Los costos del software pueden superar ampliamente a los del hardware. Lo anterior es cierto en el caso de aplicaciones a la medida, aunque los costos de aplicaciones genéricas también se han incrementado rápidamente.

Existen varias razones para este incremento: El hardware cada vez es menos caro, así sea un mainframe, mini o PC, lo cual ocasiona que existan más sistemas instalados. Cada uno de estos sistemas requiere como mínimo una copia del sistema operativo y una aplicación. Además de el incremento en número de sistemas, se requiere más tiempo del personal para mantener el ambiente de cómputo.

Otra razón es que el software todavía se escribe "desde cero" en organizaciones tradicionales, lo cual involucra una labor muy intensa que resulta en un proceso costoso. También hay que añadir el incremento del costo del software de sistema, especialmente los sistemas operativos propietarios. Esto es debido a que los proveedores tipo IBM solían utilizar las ganancias del hardware para de alguna forma solventar los costos de creación de software. Al caer los precios del hardware y disminuir los márgenes de ganancia es necesario para los proveedores el buscar ganancia en el software.

Es necesario tener en mente que las computadoras y sistemas operativos existen para una sola cosa: Ejecutar aplicaciones. Desafortunadamente, el diseño tradicional de aplicaciones rara vez hace énfasis en eficiencia operativa y facilidad de mantenimiento. Dependiendo del ciclo de vida de la aplicación, lo anterior puede ser un error bastante costoso.

El diseño, desarrollo y mantenimiento de aplicaciones en forma costeable generalmente requiere revisar los procedimientos y metodologías establecidos. Es necesario hacer cuestionamientos como ¿La aplicación tendrá que ser construida a la medida? ¿Todas las características solicitadas son absolutamente requeridas? ¿Es posible comprar una aplicación similar? ¿Que tipo de hardware es necesario para correr la aplicación? ¿Que tan crítica es la aplicación para el negocio? ¿Cuánto tiempo será utilizada la aplicación?. Asignar un valor a cada una de las preguntas nos ayudará a estimar el costo de desarrollar y mantener la aplicación requerida. Es necesario asegurarse de que existe una necesidad de negocio para la aplicación. Muchas veces la mejor estrategia para una aplicación mal diseñada es eliminarla desde el principio.

Las aplicaciones a la medida son muchas veces empleadas por las compañías para tener una ventaja de negocio contra los competidores, como es el caso de las aerolíneas y los bancos. En ocasiones esta ventaja no justifica el alto costo, ya que muchas aplicaciones a la medida pueden ser duplicadas fácilmente, con lo cual la ventaja competitiva existirá por poco tiempo.

Existen varias estrategias para crear una nueva aplicación. El desarrollo de programas en forma tradicional todavía es una alternativa viable. De hecho las disciplinas utilizadas tradicionalmente todavía son necesarias. Lo que se necesita cambiar es la velocidad con la que el desarrollo se lleva a cabo. Mediante una combinación de nuevas herramientas y nuevas técnicas, las aplicaciones pueden ser desarrolladas de manera más económica y en menor tiempo.

CASE

Se ha escrito mucho acerca de CASE. Muchos de los pasos de esta metodología no son nuevos. De hecho la mayoría de los programadores reutilizan código a partir del segundo programa que escriben; adicionalmente, los prototipos pueden ser evolucionados a aplicaciones. Una de las innovaciones que se tiene con CASE es el obligar a seguir buenas técnicas de programación y restringir malos hábitos. Otra de las mejoras es la portabilidad del código.

El desarrollo de aplicaciones debe comenzar con un análisis apropiado a partir de requerimientos claros. Con las herramientas para prototyping disponibles actualmente estos requerimientos pueden ser rápidamente transformados en prototipos o pruebas de concepto para una evaluación inmediata. Los requerimientos entonces pueden ser re-evaluados y posteriormente actualizar el prototipo siguiendo un proceso interactivo e intuitivo hasta que se establecen los requerimientos finales.

La creación de prototipos en forma interactiva es extremadamente efectiva en esta etapa del desarrollo de una aplicación. Deficiencias de comunicación y errores de diseño se eliminan cuando el usuario prácticamente esta viendo y usando la aplicación mientras define los requerimientos. La importancia del prototyping reside en el hecho de que es menos costoso el cambiar una especificación antes que empiece el proceso de codificación que cuando el programa está casi terminado.

Una vez que el prototipo ha sido aceptado por el usuario, la construcción puede comenzar. Algunas herramientas CASE generan código para producción a partir del prototipo, utilizando porciones del código existente que pueden ser empleadas en la nueva aplicación. La re-usabilidad del código es muy importante, ya que no sólo la aplicación es desarrollada más rápido sino que el código es más estable ya que ha sido probado y depurado previamente. Lo anterior es fundamental y parte de las ventajas de utilizar metodologías con orientación a objetos.

Cuando la aplicación se encuentra en producción, el mantenimiento se simplifica. Parte de la metodología CASE implica el documentar todo el código fuente utilizado en el nuevo programa. Si el código está registrado, entonces éste puede ser encontrado siempre que sea necesario realizar un cambio y determinar el impacto de dicha modificación en otros módulos del sistema. Uno de los mayores problemas con el mantenimiento de aplicaciones es la dificultad de localizar código viejo. Con el código fuente accesible y en un formato estructurado de acuerdo a la herramienta CASE, las modificaciones pueden realizarse en formas más sencilla y rentable por cualquier programador.

El uso de alguna metodología implica que se emplean herramientas para desarrollo adecuadas. Estas herramientas generalmente incluyen un módulo para prototyping, un módulo para almacenamiento/re-uso de código, un diccionario de datos para definiciones de tablas y campos, así como un generador de código que transformará el prototipo en código de producción para ejecución posterior. Aunque costosas, estas herramientas incrementan en gran medida los niveles de productividad en lo que respecta a desarrollo y mantenimiento. En ambientes multiplataforma, herramientas que soporten tanto ambientes mainframe como PC son la opción más adecuada y económica, ya que se pueden realizar las actividades de creación de prototipos y codificación en plataformas económicas como es el caso de un equipo PC, de tal forma que las pruebas de integración de módulos como un sistema completo se puedan llevar a cabo en el servidor. Además, resulta conveniente el poder contar con la capacidad de migrar o portar fácilmente el código a otra plataforma si fuera necesario, aunque hay que tener en cuenta que el manejo de archivos y funciones a nivel sistema operativo cambian dependiendo del ambiente.

La otra ventaja de un ambiente integrado CASE (I-CASE) es el control de versiones, ya que la mayoría de estas herramientas están diseñadas para mantener distintas versiones del código incluyendo su distribución, con lo cual se disminuyen errores de operación al automatizar esta tarea.

Decidiendo el diseño correcto

Existen varias opciones en lo que respecta a arquitectura al implementar un sistema. El método tradicional implicaba tener código en una sola estructura o configuración. Actualmente existen varios modelos basados en capas de software para ciertas operaciones que han sido aisladas de funciones específicas del sistema operativo o protocolo de comunicaciones utilizando distintas interfaces estándar, conocidas como API. Estos API son los que permiten la comunicación entre las distintas capas de software evolucionando al grado de contar con servicios o llamadas estándar disponibles a cualquier programa a través de llamadas al API, que es la base del diseño del kernel de la mayoría de los sistemas operativos. El siguiente paso fue la capacidad de poder tener porciones de código de un programa divididas o repartidas en múltiples procesadores o inclusive en múltiples plataformas, en donde una sección del código realiza petición de datos (cliente) y la otra responde con dicha información (servidor). Lo anterior es lo que se denomina arquitectura Cliente/Servidor.

Si se implementa una estrategia basada en capas de código es necesario seleccionar una arquitectura. Varios proveedores han creado especificaciones o estándares para construcción de esquemas cliente/servidor. Los diseños de Sun, DEC, IBM y OSF muestran similitudes en el objetivo de su diseño: Separación de la presentación del procesamiento, mecanismos de transporte, formatos de almacenamiento y métodos de acceso.

Los beneficios de esta implementación son importantes: La capacidad de mover de una plataforma a otra con cambios mínimos, lo cual significa hacer menos re-escritura de código, menos errores e implementaciones más rápidas. Prácticamente todos los proveedores de software permiten implementar ambientes de este tipo, por lo cual es necesario contemplar esta arquitectura en los nuevos desarrollos.

Problemas en el desarrollo basado en PC

Uno de los mayores problemas en aplicaciones de PC es la estabilidad y confiabilidad de la aplicación cuando se mueve del ambiente de desarrollo al de producción. En muchas ocasiones esto se debe a que no se utiliza una metodología de diseño y desarrollo formal. Debido a que el desarrollo

y pruebas se realizan en este ambiente, aparentemente el programa es muy estable. Conforme la carga de trabajo aumenta, típicamente se decide mover el programa o aplicación a un ambiente multiusuario. A partir de este momento pueden surgir varios problemas como es el caso de que el programa nunca haya sido probado en lo referente a integridad de datos al existir múltiples actualizaciones o accesos. Por razones como la anterior, siempre se debe utilizar el programa bajo el mismo ambiente en cual se hicieron pruebas. Se deben evitar suposiciones en cuanto a la transparencia de mover una aplicación de un ambiente stand-alone a uno compartido. Es necesario probar el programa y realizar los cambios pertinentes según sea necesario.

Actualmente la construcción adecuada de aplicaciones es mas critica que nunca. Es importante identificar los puntos en los que la aplicación contribuirá a la rentabilidad de la empresa. Si estos puntos no existen entonces la aplicación debe ser modificada o inclusive reestructurada o bien eliminada. En el primer caso, es conveniente adoptar alguna estrategia de desarrollo como CASE.

En los siguientes capitulos se detallarán aspectos sobre un grupo de herramientas de desarrollo, bases de datos y características de la tecnología cliente/servidor y sistemas de información geográfica para posteriormente explicar la interacción de éstos en un caso práctico.

Conceptos de Bases de Datos

Me pareció que era esencial introducir alguna disciplina en el diseño de bases de datos. La llamé normalización porque en ese tiempo el presidente Nixon hablaba mucho acerca de normalizar las relaciones con China. Pensé que si él podía normalizar relaciones, también yo era capaz de hacerlo.

Dr. Edgar F. Codd
DBMS Magazine
Diciembre 1993

2.1 Introducción

Los sistemas manejadores de bases de datos¹ (DBMS por sus siglas en inglés) organizan y estructuran los datos de tal forma que éstos puedan ser consultados y manipulados por usuarios y aplicaciones. Las estructuras de datos y técnicas de acceso de un manejador en particular se conocen como modelo de datos. Este modelo es el que determina la "personalidad" de un DBMS y las aplicaciones para las cuales es más adecuado.

Conforme el manejo de datos se hizo popular durante los 70s y 80s, aparecieron distintos modelos de datos. Cada uno de estos tuvo ventajas y desventajas que fueron determinantes para la creación de modelos de datos posteriores, como es el caso del modelo relacional.

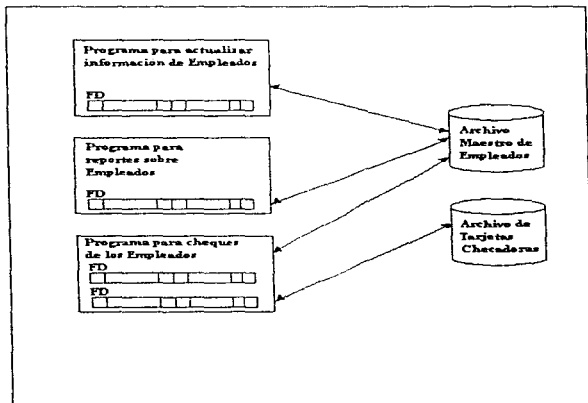
Este capítulo pretende mostrar en forma breve los elementos básicos de los modelos más comunes de bases de datos con el fin de hacer notar sus características principales y poder determinar su utilidad. Se explica con un poco más de detalle el modelo relacional y conceptos asociados por la importancia y vigencia de este modelo.

¹ En este trabajo se utilizarán las siglas DBMS para referirse al manejador de bases de datos

2.2 Sistemas para manejo de archivos

Antes de la aparición de los sistemas manejadores de bases de datos toda la información almacenada permanentemente en la computadora, como los registros de nómina y contabilidad, se almacenaba en archivos individuales. Un sistema manejador de archivos, usualmente proporcionado por el fabricante como parte del sistema operativo, llevaba el control de los nombres y localización de los archivos. Un sistema manejador de archivos no tenía modelo de datos. No incluía información acerca del contenido de los archivos. Para el sistema manejador de archivos era igual un archivo con un documento de un procesador de palabras que un archivo con registros de nómina.

El conocimiento sobre el contenido de los archivos, es decir, los datos que contiene y la organización de éstos, se incluía en los programas aplicativos que usaban determinado archivo como se muestra en la figura:



En este ejemplo de una aplicación de nómina, cada uno de los programas COBOL que procesan el archivo maestro de empleados contiene una descripción de archivo que muestra la disposición de los datos en el archivo. Si la estructura de los datos cambia, por ejemplo, si se almacena alguna información adicional sobre cada empleado, entonces cada programa que accesa al archivo tendría que ser modificado. Conforme el número de programas y archivos crece, el esfuerzo y recursos del área de sistemas se tiene que dedicar al mantenimiento de las aplicaciones existentes en vez de crear aplicaciones nuevas.

Los problemas de mantenimiento de sistemas basados en archivos originó al final de los 60s el desarrollo de sistemas manejadores de bases de datos. La idea de estos sistemas era simple: tomar la definición del contenido y estructura de un archivo a partir de los programas y almacenarla junto con los datos en una base de datos. Utilizando la información en la base de datos, el DBMS que la controla puede administrar dicha información así como los cambios a la estructura de la base de datos.

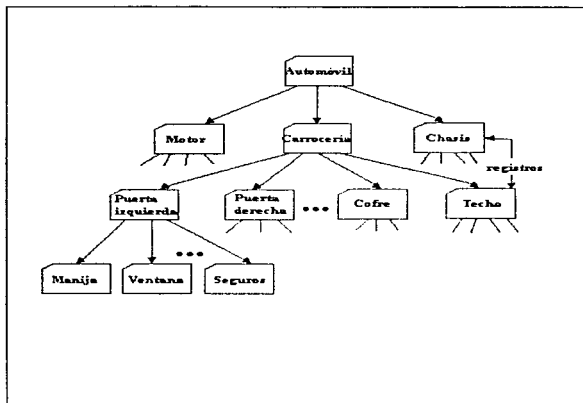
2.3 Modelo Jerárquico

Una base de datos jerárquica consiste en un conjunto de registros que se conectan entre sí por medio de ligas. Un registro es un conjunto de campos o atributos, cada uno de los cuales contiene solo un valor. Una liga es una asociación entre dos registros.

Una de las aplicaciones más importantes de los primeros sistemas manejadores de bases de datos fue la planeación de producción para compañías de manufactura. Si un fabricante de automóviles decidía producir 10,000 unidades de determinado modelo y 5,000 de otro, era necesario saber cuantas partes se tenían que ordenar a los proveedores. Para conocer la respuesta, el producto (automóvil) tenía que ser dividido en partes (motor, carrocería), que a su vez eran descompuestas en subpartes (válvulas, cilindros, bujías) y estas en sub-subpartes, etc. El manejo de esta lista de partes era un trabajo ideal para utilizar una computadora.

La lista de partes para un producto tenía de forma natural una estructura jerárquica. Se desarrolla el modelo jerárquico para almacenar este tipo de información (ver siguiente figura). En este modelo cada registro de la base de datos representaba una parte específica. Los registros tenían relaciones padre/hijo, ligando cada parte a su subparte. Para acceder los datos en la base de datos un programa podría:

- buscar una parte en especial por número (una puerta izquierda)
- moverse "abajo" al primer hijo (manija de la puerta)
- moverse "arriba" al padre (la carrocería) o
- moverse "de lado" al siguiente hijo (la puerta derecha)



De acuerdo a lo anterior, el obtener los datos almacenados en una base de datos jerárquica requerirá “navegar” o moverse por los registros hacia arriba, abajo o de lado un registro a la vez.

El conjunto de todas las piezas está organizado en forma de un árbol con raíz, en el cual ésta se denomina nodo de trabajo. Una base de datos jerárquica está formada por un conjunto de árboles de este tipo, formando así un bosque. Un árbol de esta clase, dotado de raíz, se denomina árbol de la base de datos.

Uno de los manejadores de bases de datos jerárquico más popular fue el Information Management System (IMS) de IBM. Como ejemplo, se muestran a continuación las ventajas de IMS y su modelo jerárquico:

- Estructura Simple. La organización de una base de datos IMS era fácil de comprender. La jerarquía de la base de datos era igual a la de un organigrama o a la de un árbol genealógico.

- **Organización Padre/Hijo.** Una base de datos IMS era excelente para representar relaciones padre/hijo, como "A es parte de B" o "A pertenece a B"
- **Rendimiento.** IMS almacenaba las relaciones padre/hijo como apuntadores físicos de un registro de datos a otro, por lo que el movimiento a través de la base de datos era rápido. Debido a que la estructura de datos era simple, IMS podía colocar registros padre/hijo cerca uno de otro en el disco, de tal forma que se minimizaba I/O.

El contenido de un registro específico puede repetirse en varios lugares. Por ejemplo, una manija puede pertenecer a más de una puerta. La información correspondiente a esa manija, o la relacionada a las puertas que puede pertenecer, tendrá que repetirse. Este repetición puede darse tanto en el mismo árbol de la base de datos como en varios árboles distintos. La repetición de registros tiene dos desventajas principales:

- Puede producirse una inconsistencia en los datos al llevar a cabo una actualización.
- Será inevitable el desperdicio de espacio.

La referencia a IMS y en general al modelo de datos jerárquico se hace en tiempo pasado (era, hacia, permitía, etc.) debido a que prácticamente es un modelo en vías de extinción. IMS es todavía la base de datos predominante en mainframes IBM. Se utiliza en aproximadamente 25 por ciento de las instalaciones IBM mainframe.

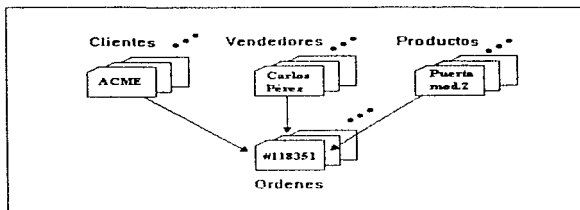
La primera versión de IMS se desarrolló a finales de la década de 1960 por IBM y North American Aviation (Rockwell International) para el programa Apolo de alunizaje.

2.4 Modelo de Red

Una base de datos de red también consiste en un conjunto de registros y ligas (links). Una liga es una asociación entre dos registros.

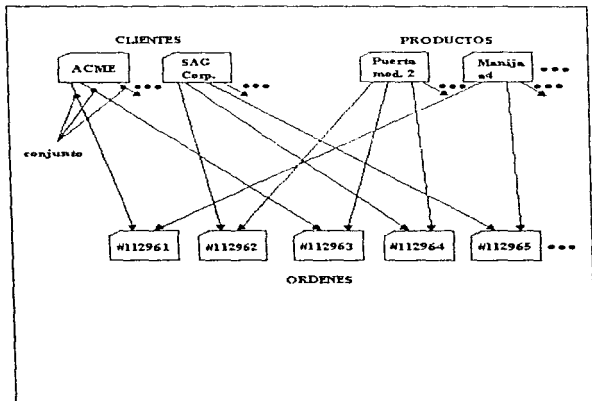
Un diagrama de estructura de datos para una base de datos de red consta de dos componentes básicos: cuadros (tipo de registro) y líneas (ligas).

La estructura simple de una base de datos jerárquica se convirtió en una desventaja cuando los datos tenían una estructura más compleja. Por ejemplo, en una base de datos para proceso de órdenes, una orden podía encontrarse en tres diferentes relaciones padre/hijo, ligando la orden al cliente que la realizó, al vendedor que levantó dicha orden y al producto ordenado, como se muestra en la siguiente figura:



La estructura manejada por este tipo de datos simplemente no encajaba en la estricta jerarquía de bases de datos como IMS.

Se desarrolló un nuevo modelo de datos para implementar aplicaciones de este tipo. El modelo de datos de red extendía el modelo jerárquico permitiendo que un registro participara en múltiples relaciones padre/hijo (ver siguiente figura)



Estas relaciones se conocieron en el modelo de red como conjuntos. En 1971 la organización CODASYL (Conference on Data Systems Languages) publicó un estándar oficial para bases de datos de red, que llegó a conocerse como el modelo CODASYL. IBM nunca desarrolló una base de datos de red, sino que extendió a IMS por varios años. Sin embargo durante la década de 1970 varias compañías de software independientes se apresuraron a adoptar el modelo de red, creando productos muy populares como IDMS de Cullinet y Total de Cincom.

Para el programador el acceder una base de datos de red era muy similar a acceder una base de datos jerárquica. Un programa podía:

- Encontrar un registro padre específico por llave (como número de cliente)
- Moverse abajo al primer hijo de un conjunto en particular (la primera orden hecha por un cliente)
- Moverse al lado de un hijo al siguiente en el conjunto (la siguiente orden hecha por ese cliente)
- Moverse arriba de un hijo a su padre en otro conjunto (el vendedor que recibió la orden)

El programador tenía que navegar la base de datos registro por registro especificando la relación así como la dirección.

Las bases de datos de red tenían varias ventajas:

- Flexibilidad. Múltiples relaciones padre/hijo permitían representar datos que no tenían una estructura jerárquica simple.
- Estandarización. El estándar CODASYL aumento la popularidad del modelo de red. Proveedores como Digital Equipment Corporation y Data General hicieron implementaciones de este modelo.
- Performance. A pesar de su complejidad, las bases de datos de red incrementaron el performance al nivel de una base de datos jerárquica. Los conjuntos eran representados por apuntadores a registros físicos de datos, y en algunos sistemas el administrador podía especificar agrupamientos o acoplamientos de datos en base a una relación.

Las bases de datos de red también tuvieron desventajas. Al igual que las bases de datos jerárquicas, eran poco flexibles. Las relaciones entre conjuntos y la estructura de los registros tenían que ser especificadas desde el inicio. El cambiar la estructura de la base de datos típicamente requería volver a construirla.

2.5 Modelo Relacional

Las desventajas de los modelos jerárquico y de red ocasionaron gran interés en el modelo relacional descrito por Codd en 1970. Este modelo fue un intento de simplificar la estructura de una base de datos. Eliminaba las estructuras explícitas padre/hijo de la base de datos y en su lugar representaba todos los datos de la base de datos como simples tablas formadas por renglones y columnas. La siguiente figura muestra la versión relacional de la base de datos para proceso de órdenes utilizada para ilustrar los modelos anteriores:

TABLA DE ORDENES

ORDEN	COMPANIA	PRODUCTO	CANT.
112963	ACME	41004	28
112975	SAG Corp.	2A44G	6
112983	ACME	41004	6
113012	ACME	41003	35

TABLA DE PRODUCTOS

DESCRIPCION	PRECIO	CANTIDAD DISPONIBLE
Manija Tamaño 3	\$94.00	100
Manija Tamaño 4	\$102.50	33
Puerta Mod. 4	\$519.00	4

TABLA DE CLIENTES

COMPANIA	REP.	CREDITO
ACME	105	\$50,000.00
SAG Corp.	103	\$10,000.00

Desafortunadamente la definición práctica de lo que era una base de datos relacional se alejó de lo que era la definición matemática expuesta en el trabajo de Codd en 1970. Los primeros sistemas relacionales fallaron en implementar algunas partes clave del modelo de Codd y que actualmente apenas comienzan a aparecer en los productos comerciales.

Conforme el modelo relacional crece en popularidad muchas bases de datos se denominan relacionales aunque no lo son. En respuesta a la corrupción del término relacional, Codd escribe en 1985 un artículo en donde indica doce reglas que deben ser seguidas por cualquier base de datos que se diga relacional. Estas doce reglas han sido aceptadas como la definición de un RDBMS verdadero. Sin embargo es más fácil comenzar con una definición informal:

Una base de datos relacional es una base de datos donde todos los datos visibles al usuario están organizados como tablas de datos, y en donde todas las operaciones de la base de datos se aplican a dichas tablas.

Esta definición pretende distinguir estructuras como las de apuntadores de las bases de datos de red y jerárquicas. Un DBMS relacional puede representar relaciones padre/hijo, pero éstas son representadas estrictamente por los datos contenidos en las tablas.

Tablas

El principio de organización de una base de datos relacional es la tabla, que es un arreglo rectangular de renglones y columnas de datos. Cada tabla en la base de datos tiene un nombre único que identifica su contenido; en la mayoría de los RDBMS cada usuario puede escoger los nombres de las tablas sin preocuparse por los nombres escogidos por otros usuarios.

En una tabla cada renglón representa una entidad física. Cada columna de la tabla representa un elemento de los datos que se almacenan en la base de datos. Se identifican con el nombre de columna, que comúnmente se escribe como encabezado de ésta. Las columnas de una tabla deben tener nombres diferentes, aunque se pueden tener dos columnas con el mismo nombre pero de diferente tabla.

Las columnas de una tabla tienen un orden de izquierda a derecha que se define cuando se crea la tabla. Una tabla siempre tiene al menos una columna. El estándar ANSI/ISO SQL no especifica un máximo de columnas por tabla pero casi todos los RDBMS comerciales tienen un límite de alrededor de 255 columnas por tabla.

A diferencia de las columnas, los renglones de una tabla no tienen un orden en particular. Si se realizan dos consultas consecutivas a una tabla el resultado no necesariamente tendrá el mismo orden, a menos que explícitamente se indique un ordenamiento.

Una tabla puede tener cualquier número de renglones. Una tabla de cero renglones es totalmente válida ya que una tabla vacía cuenta con una estructura definida por sus columnas. El estándar ANSI/ISO no impone límite para el número de renglones en una tabla; muchos productos permiten que ésta aumente de tamaño hasta que se agote el espacio en disco.

Llaves primarias

Debido a que los renglones de una tabla relacional no están ordenados, no es posible seleccionar uno específico por su posición en la tabla. En una base de datos relacional bien diseñada cada tabla tiene una columna o combinación de columnas cuyos valores identifican en forma única cada renglón de la tabla. Esta columna (o columnas) se denomina llave primaria de la tabla.

La llave primaria tiene un valor único, por lo que no pueden existir dos renglones duplicados. Una tabla en donde cada renglón es diferente de los demás se denomina en términos matemáticos relación. El nombre base de datos relacional se deriva de este término.

A pesar de que las llaves primarias son un concepto esencial en el modelo relacional, los primeros RDBMS como System/R, DB2 y Oracle no soportaban explícitamente a las llaves primarias. Los diseñadores de bases de datos generalmente se aseguraban que todas las tablas de la base de datos tuvieran una llave primaria, pero la base de datos no proporcionaba una forma de identificar la llave primaria de una tabla. La versión 2 de DB2, introducida en Abril de 1988, incluyó el soporte a llaves primarias. Posteriormente el estándar ANSI/ISO se expandió para soportarlas.

Relaciones

Una de las mayores diferencias entre el modelo relacional y los modelos de datos anteriores es que los apuntadores explícitos, como en las relaciones padre/hijo de una base de datos jerárquica, no están permitidos. Obviamente este tipo de relaciones existen en un RDBMS, pero no se expresan con un apuntador explícito almacenado en la base de datos. En vez de esto la relación se representa por valores comunes almacenados en las tablas. Esta es la forma en como se representan las relaciones en un RDBMS.

Llaves Foráneas

Una columna en una tabla cuyo valor coincide con el de la llave primaria en otra tabla se conoce como llave foránea. Al igual que las llaves primarias, una llave foránea puede estar formada por una o más columnas. Una tabla puede tener más de una llave foránea si se relaciona con dos tablas o más.

Las llaves foráneas son parte clave del modelo relacional debido a que crean relaciones entre las tablas de la base de datos. Desafortunadamente, al igual que las llaves primarias, éstas no eran soportadas en los primeros RDBMS.

SQL

SQL es un lenguaje para organizar, administrar y consultar datos almacenados en una base de datos relacional. El nombre SQL es un acrónimo de Structured Query Language.

SQL fue definido por IBM para los productos SQL/DS y DB2. Muchos proveedores lo han adoptado como estándar pero le han agregado extensiones que inhiben la interoperabilidad entre RDBMSs, razón por la cual la necesidad de crear estándares fue crítica.

ISO y ANSI han creado grupos de trabajo para definir estos estándares. El estándar ANSI/ISO incluía el documento ANSI X3.135-1986 junto con el Addendum 1, en donde se delineaba la sintaxis y semántica de la integridad referencial.

En 1989 se hizo una adición al estándar que se denominó ANSI Database Language SQL with Integrity Enhancement. Esta especificación detallaba la sintaxis para los comandos del lenguaje de definición de datos (DDL o Data Definition Language) y del Lenguaje de Manipulación de Datos (DML o Data Manipulation Language) así como comandos SQL incrustados dentro de otro lenguaje.

El estándar de 1989 menciona dos niveles de adhesión. El nivel 2 es todo el paquete SQL. Nivel 1 es un subconjunto del nivel 2 y representa a varias implementaciones de SQL vigentes en ese tiempo. La variante SQL2 ya fue liberada. SQL3 sigue en elaboración.

Mientras tanto el SQL Access Group (SAG) trata de resolver las diferencias entre las implementaciones comerciales del lenguaje aunque

algunos proveedores importantes no forman parte de este grupo, por lo cual no es seguro el éxito de éste.

X/Open es otra organización relacionada con estándares SQL, especialmente en ambientes UNIX y sus especificaciones apenas siguen las de ANSI. El estándar de acceso remoto a bases de datos (RDA o Remote Database Access) se diseñó para facilitar el acceso a servidores de bases de datos desde aplicaciones cliente.

A pesar de que una de sus más importantes funciones es la de consultar datos, SQL es utilizado para controlar otras funciones de un RDBMS como:

- **Definición de Datos.** SQL permite a un usuario definir la estructura y organización de los datos almacenados y las relaciones entre los elementos almacenados.
- **Consulta de Datos.** SQL permite a un usuario o a una aplicación leer información de una base de datos y utilizarla.
- **Manipulación de Datos.** SQL permite a un usuario o aplicación actualizar la base de datos al agregar nueva información, borrando datos o modificando datos ya existentes.
- **Control de Acceso.** SQL puede ser utilizado para restringir la capacidad de un usuario para leer, agregar y modificar datos.
- **Compartir Datos.** SQL es utilizado para coordinar y asegurar la disponibilidad de los datos para varios usuarios sin que éstos interfieran entre sí.
- **Integridad de Datos.** SQL define reglas de integridad en la base de datos protegiéndola de interrupciones constantes debido a actualizaciones inconsistentes o fallas en el sistema.

SQL no es un lenguaje de programación completo como sería el caso de COBOL, FORTRAN o C, ya que no contiene sentencias IF para prueba de condiciones, no contiene GOTO para moverse en el programa y no soporta DO o FOR para ciclos. SQL puede ser visto como un sublenguaje para manejo de datos. Consta de alrededor de 30 verbos orientados para manejar una base de datos. Estos enunciados SQL pueden ser incrustados en un 3GL para extender dicho lenguaje en el uso de la base

de datos. Los enunciados SQL también pueden ser mandados en forma explícita a la base de datos mediante distintas herramientas.

SQL no es un lenguaje estructurado como C o Pascal. Al contrario, está formado por comandos que forman sentencias parecidas al inglés ya que incluye verbos que no realizan ninguna función mas que la de hacer más naturales los enunciados. Lo anterior es una de las razones por las cuales SQL es el lenguaje estándar para acceso a un RDBMS, ya que es un lenguaje poderoso y relativamente fácil de aprender.

Existen varias ventajas del modelo relacional entre las cuales se puede mencionar la separación del modelo de datos de la implantación física, lo cual no ocurría en los modelos anteriores. Otro característica importante es la correspondencia directa entre el concepto de una tabla y el concepto matemático de una relación, por lo cual existe una firme base teórica asociada al modelo relacional.

Actualmente las bases de datos son parte clave de cualquier sistema. El modelo relacional es el modelo más utilizado, sin embargo conforme se definen nuevos estándares, capacidades adicionales son contempladas para estos productos. Es interesante notar que actualmente muchos proveedores preparan versiones Objeto-Relacionales de sus productos, por lo que habrá que ver hasta que punto dichas implementaciones ofrezcan ventajas suficientes como para migrar los ambientes y aplicaciones relacionales existentes.

Otro aspecto importante asociado a una base de datos es la forma o los medios disponibles para acceder la información almacenada. Como se vio en este capítulo, SQL es responsable en gran parte de la aceptación que ha tenido el modelo relacional por lo cual es necesario conocer las adiciones hechas al lenguaje con los nuevos estándares como el SQL3 y determinar en que forma se soportan nuevos modelos de datos.

La disponibilidad de herramientas de desarrollo que soporten al modelo de datos también es determinante para su aceptación. Al igual que SQL, los lenguajes de cuarta generación o 4GLs, aunque no diseñados exclusivamente para el modelo relacional, han influido en gran medida a la difusión de este modelo. En el siguiente capítulo se exponen diversos temas relacionados con estas herramientas de desarrollo.

Conceptos de Lenguajes de Cuarta Generación

Cualquier primer intento en convertir al folklore en conocimiento y a un juego de azar en disciplina, está sujeto a ser mal interpretado como una degradación de la habilidad individual y su sustitución por un libro de reglas. Por supuesto, cualquier intento de este tipo no tendría sentido. Ningún libro hará que un tonto se convierta en un hombre inteligente o hará un genio de un incompetente. Sin embargo actualmente la base en una disciplina le proporciona a un médico la capacidad de desempeñarse más allá que el más hábil doctor de hace un siglo, y permite al médico sobresaliente de hoy hacer lo que el genio de ayer ni siquiera hubiera soñado. Ninguna disciplina puede extender el brazo del hombre, pero puede extender su alcance levantándolo sobre los hombros de sus predecesores. El conocimiento organizado en una disciplina sirve bastante bien para quien es apenas competente; le viste de cierta efectividad. Hace infinitamente más para el que es verdaderamente capaz; le viste con la excelencia.

Peter F. Drucker
Managing for Results
1964.

3.1 Introducción

Actualmente una de las principales influencias en el progreso de la computación es la velocidad de los microprocesadores. Estos chips son componentes versátiles que pueden ser utilizados en diversos dispositivos, por lo cual un aumento de la capacidad y velocidad de estos chips impacta directamente en las posibilidades de los equipos que los utilizan. Sin embargo existe un periodo de adaptación o aprovechamiento de la tecnología debido a la necesidad de cambiar los dispositivos asociados así como escribir software que utilice todas las capacidades de los nuevos procesadores, lo cual da como resultado que el avance no sea instantáneo. Esto nos lleva a situaciones como el hecho que PCs con procesadores 486 o Pentium cuentan con un bus y periféricos que son muy lentos para el procesador. Al cambiar la arquitectura de todo el sistema es cuando la diferencia en rendimiento es bastante notoria.

Existe otra influencia diferente, pero igual de importante, en lo que respecta a la interface hombre/máquina. Si la potencia del procesador determina "que" se puede hacer, entonces la interface hombre/máquina determina "como" se utilizará este poder de cómputo. A un nivel muy básico esta interface determina que puede ser puesto en la computadora y por lo tanto tiene un efecto sobre el tipo de aplicaciones que pueden ser creadas.

La transición a ambientes cliente/servidor generalmente va en paralelo con el cambio de interfaces caracter a interfaces gráficas. Lo anterior es un cambio dramático provocado por el incremento cada vez mayor del poder de procesamiento de los CPUs. Inicialmente los 4GL¹ surgieron como productos con interface modo caracter; requerimientos en cuanto al manejo de datos tipo bitmap y gráficos así como capacidades para utilizar hipertexto y texto no estructurado han dado surgimiento a gran variedad de productos con capacidades muy interesantes.

Los principales requerimientos para un 4GL incluyen un soporte sólido a ambientes cliente/servidor así como la capacidad de manejar objetos de alguna forma. Lo anterior puede sugerir que un 4GL necesite convertirse un lenguaje OO, aunque no necesariamente ya que actualmente un lenguaje completo con orientación a objetos puede ser demasiado sofisticado para el programador promedio.

¹ Se utilizara 4GL como abreviacion de lenguaje de cuarta generacion (Fourth Generation Language)

3.2 Mercado de los Lenguajes de Cuarta Generación

La diversidad de características y la rapidez con la que aparecen nuevas versiones de lenguajes de programación con características adicionales y mejoradas hacen un tanto complejo el esfuerzo de categorizar los distintos ambientes de desarrollo, ya que existen diferentes tipos y tienen diferentes orígenes. Se puede decir que el único factor que tienen en común es que pueden ser utilizados para construir sistemas multiusuario, ya que el tratar de compararlos uno a uno sería inútil. Se pueden considerar los siguientes grupos:

- 4GL a nivel mainframe
- 4GL modo caracter para plataformas medianas
- 4GL modo GUI para plataformas medianas
- 4GL con orientación a objetos
- 4GL para AS/400
- 4GL para PC

4GL para mainframe

En este grupo sobresalen tres productos: Huron de Amdahl, LINC de Unisys y NATURAL de Software AG. Huron es un ambiente de desarrollo orientado a desarrollo rápido para ambientes corporativos. Contiene un diccionario de datos dinámico y soporta prototyping para desarrollo de sistemas. Incluye una base de datos propia que podría competir con otras bases de datos en estos ambientes.

LINC es un ambiente de desarrollo establecido que se ha hecho poderoso a través de los años. Es ampliamente utilizado en la base instalada de Unisys, lo cual ha hecho que sus ventas de hardware no disminuyan tan rápido.

NATURAL sigue siendo uno de los lenguajes de desarrollo más completos y mejor diseñados disponible para mainframes y plataformas medianas.

4GL para plataformas medianas

En estas plataformas se hace la distinción entre ambientes de desarrollo con interface modo caracter y ambientes de desarrollo con interface GUI. Dentro de los lenguajes con interface caracter Magic es uno de los productos más poderosos. SeaChange también tiene varias características que lo hacen bastante completo ya que cuenta con un diccionario integrado.

Recital es un producto diseñado para programadores con experiencia en Xbase que necesitan migrar de PC a UNIX. Proporciona un ambiente completo, incluyendo una base de datos así como capacidad para manejar y crear formas. Empress también es un ambiente de desarrollo con buenas capacidades de distribución y que se integra bastante bien con su base de datos.

4GL gráficos (GUI)

Varios de los 4GL para ambientes gráficos surgieron con una interface caracter como es el caso de Systemator, que ofrece varias funciones como capacidad para CASE, buen performance y una implementación GUI decente.

JAM también es uno de los productos sobresalientes. Cuenta con un implementación bastante completa para la creación de formas. Además resulta atractivo por el hecho de no cobrar por la licencia de runtime. Uniface y Oracle también sobresalen en esta categoría.

4GL Orientados a Objetos

Existen productos interesantes como Windows 4GL de CA (antes ASK/Ingres), ya con varios años en el mercado. También SmartStar Vision, SQL Windows y Powerbuilder sobresalen en esta categoría, aunque la cantidad de productos de este tipo está aumentando rápidamente.

4GL para AS/400

Uno de los productos líderes es LANSA, ya que al ser nativo de esta plataforma trabaja de una manera más integrada. Sin embargo no se encuentra disponible en otras plataformas, por lo cual muchos usuarios de AS/400 consideran a PRO IV, Focus, Powerhouse o Progress como mejores opciones.

4GL para PC

Esta categoría incluye gran cantidad de productos y diversidad de características. Algunos de estos productos se encuentran disponibles para otras plataformas, lo cual debe tenerse en cuenta para sistemas con orientación corporativa ya que la mayoría de estos productos sólo funcionan en PC o bien sus implementaciones en otras plataformas son bastante pobres. SQLWindows y Powerbuilder son bastante utilizados, aunque otros como Magic y JAM contienen una serie de características notables.

Existen versiones PC de lenguajes nacidos en plataformas mainframe o medianas como NATURAL, que cuentan con la ventaja de portabilidad y disponibilidad de distintas aplicaciones para diversos entornos.

Herramientas como Visual Basic de Microsoft son también muy populares y cada vez cuentan con más características aunque tienen la desventaja de estar orientadas a ambientes PC, lo cual restringe sus posibilidades.

3.3 Funcionalidad

Los 4GL generalmente incluyen un conjunto de rutinas para implementar el tipo de operaciones más comunes de los sistemas comerciales, entre ellos mantenimiento de tablas. Al tratar de implementar algo que no está en las rutinas estándar es cuando uno se ve en la necesidad de codificar todo desde cero.

Como regla los 4GLs no contienen todas las construcciones de bajo nivel encontradas en los 3GLs, lo cual los hace un poco menos sofisticados en este respecto y en ocasiones es necesario hacer llamadas a rutinas escritas en un lenguaje de bajo nivel para realizar cierto tipo de operaciones.

La mayoría de los 4GLs proporcionan la facilidad de soportar macros de alto nivel, las cuales son fáciles de utilizar y constan de verbos o cláusulas opcionales dentro del lenguaje. El manejo de formas y acceso a datos se simplifica al contar con un mecanismo de llamada comprensible, lo cual es una de las facilidades más importantes de un lenguaje de cuarta generación, el cual puede ser visto como una biblioteca de rutinas con una interfase amigable, que es el lenguaje en sí.

Para las aplicaciones comunes los 4GLs ofrecen mejor funcionalidad que un 3GL. Por ejemplo, el manejo de ventanas y texto son difíciles de codificar en la mayoría de los 3GL mientras que en un 4GL éstas son funciones estándar. Inclusive con algunos productos es posible el incorporar nuevas rutinas, lo que permite extender la funcionalidad del lenguaje y crear un ambiente que se ajuste más a necesidades particulares. Muchos 4GL constituyen sólo un componente de un conjunto o línea de productos que integrados ofrecen mayor funcionalidad, pero la idea de contar con rutinas estándar para integrar dichos productos es la misma y es la base para desarrollo productivo de software.

Funcionalidad y productividad están relacionadas en cierto modo ya que la funcionalidad ofrecida por un 4GL generalmente contribuye a aumentar (o en algunos casos disminuir) la productividad; el hecho de contar con un diccionario de datos y un ambiente de desarrollo integrado puede hacer la diferencia. Vale la pena señalar que casi siempre la funcionalidad se obtiene a expensas del performance. Ventanas, interfaces gráficas y el uso de construcciones de alto nivel generalmente consumen cantidades significativas de memoria y CPU. Lo anterior es una de las razones por las que las arquitecturas cliente/servidor representan el esquema ideal para sistemas basados en un 4GL. En un ambiente de este tipo se busca utilizar

el no tan costoso y poco utilizado poder de cómputo de las PCs o estaciones de trabajo.

Es importante que los requerimientos de usuario se tengan presentes al evaluar la funcionalidad de un 4GL. Tratar de extraer funcionalidad que no es parte natural de un 4GL resulta costoso en tiempo como en rendimiento, ya que en algunos casos es mejor implementar ciertas funciones con un 3GL y utilizarlas desde el 4GL.

A continuación se muestra una tabla con algunas diferencias entre 4GLs dependiendo del ambiente de cómputo:

Características Específicas	Ambiente en Línea	Ambiente Cliente/Servidor
Medio y dispositivo de entrada	Terminal modo caracter, teclado	Dispositivos gráficos (Monitor, mouse, scanner)
Interface de Usuario	Caracter	GUI
Medio de salida	Pantalla y papel	Pantalla, papel, sonido
Resolución en impresión	Caracter	Gráfica
Medio de almacenamiento	Discos magnéticos	Disco y memoria magneto/óptica
Volúmenes de datos	Megabytes/Gigabytes	Gigabytes/Terabytes
Arquitectura básica	Un solo procesador	cliente/servidor, múltiples servidores
Software para manejo de datos	Base de datos de red o base de datos relacional	Base de datos relacional, base de datos orientada a objetos
Datos típicos	Caracteres, numéricos	Adicionalmente incluye objetos complejos y datos no estructurados (imágenes, sonido)
Operación del 4GL	Procedural	En base a eventos
Características principales del 4GL	Diccionario Generador de formas Generador de Reportes Soporte para archivos locales y bases de datos	Estructura basada en clases Herencia Diseñador GUI Interoperabilidad entre front-ends Interoperabilidad entre DBMS Soporte a Objetos
Ambiente de Desarrollo	Integrado, pero con funcionamiento en serie	Integrado, pero con funcionamiento en base a múltiples ventanas
Tareas concurrentes de usuario	Una	Muchas
Configuraciones de Hardware	Organización, Departamental	Red

3.4 Lenguajes no procedurales

Uno de los conceptos asociados a un 4GL es el de lenguaje no procedural (o declarativo). Por definición un lenguaje no procedural o declarativo es aquel en donde el orden de ejecución de las sentencias no depende de la secuencia de éstas en el programa. Por ejemplo:

```
VALIDATE FIELD_1  
GET FIELD_1  
OUTPUT FIELD_2
```

Si pensamos en que existen prioridades asociadas con cada tipo de sentencia, en el ejemplo anterior GET tiene mayor prioridad que VALIDATE, por lo cual este último no se ejecutará sino hasta después de GET. De manera similar, OUTPUT tiene mayor prioridad que GET y VALIDATE, de tal forma que será el primero en ejecutarse no importando su posición en el programa.

Este comportamiento es adecuado, sin embargo no es deseable cuando se necesita funcionalidad procedural. Por ejemplo, la sentencia VALIDATE podría necesitar ejecutar algunas sentencias de validación:

```
PROCEDURE VALIDATE FIELD_1  
IF FIELD_1 = 1  
FIELD_1 := FIELD_1 + 1  
ENDIF  
IF FIELD_1 = 1  
PRINT "ERROR"  
ENDIF
```

El segundo IF es dependiente del primero, por lo cual la rutina no daría el mismo resultado si se cambiara la posición de las sentencias. Este tipo de lógica es más fácil de programar en un lenguaje procedural. Pocos 4GLs no son procedurales, sin embargo varios tienen construcciones que no son procedurales. En el caso de generación de reportes un 4GL normalmente tiene una estructura similar a la siguiente:

- Definición de tablas y llaves
- Definición de condiciones de selección
- Definición del criterio de ordenación
- Definición de los cortes para totales
- Definición de los elementos a imprimir

No importa el orden en el que se escriban las sentencias ya que el 4GL siempre realizará un join sobre las tablas que incluyan los renglones seleccionados y los objetos requeridos para imprimir y totalizar. Después ordenará e imprimirá, calculando los totales conforme avanza.

Si un 4GL no cuenta con construcciones procedurales entonces es necesario que soporte llamadas a rutinas escritas en un 3GL. Un 4GL no procedural puede resultar muy pobre en sistemas de mediana complejidad, razón por la cual una buena interface a 3GL es importante. La alternativa (algo que se hace comúnmente) es el tratar de "forzar" al lenguaje para que trabaje en modo procedural (para lo cual no fue diseñado). En estas condiciones puede llevar toda una semana escribir un programa que llevaría sólo algunas horas si se hiciera en COBOL, por ejemplo.

Algunos lenguajes no procedurales (Prolog) son capaces de implementar procesos muy complejos. Sin embargo el orden de la lógica reside en el código compilado, lo que hace difícil comprender el orden de proceso. Los 4GLs que se anuncian como no procedurales no incluyen estructuras internas tan sofisticadas como los lenguajes tipo Prolog.

3.5 El modelo relacional y los 4GL

La mayoría de los 4GLs están basados implícitamente en el modelo relacional. Este ha sido ampliamente adoptado debido a su poder y simplicidad aunque impone algunas limitaciones.

Por ejemplo, el detalle de clientes sería almacenado en la tabla CLIENTES con varias columnas (o campos) representando a los atributos (código de cliente, nombre del cliente, etc.). Cada registro en una tabla se denomina renglón y cada renglón debe ser identificado en forma única (llave primaria).

Las tablas pueden ser manipuladas mediante operadores relacionales como SELECT, PROJECT y JOIN. SELECT permite seleccionar renglones específicos de una tabla para desplegarlos o manipularlos posteriormente. PROJECT es una manera de reducir el número de columnas en una tabla. Un JOIN permite juntar datos de varias tablas como si éstas fueran una.

Los operadores relacionales utilizan tablas como entrada y pueden producir nuevas tablas o vistas, ya sea en forma permanente o temporal. El manejo de los datos se hace mediante conjuntos.

Bases de datos relacionales y 4GLs

El uso del modelo relacional implica varias cosas en un 4GL. Toda la arquitectura del producto estará orientada a tablas relacionales, por lo cual no podrán manejarse grupos periódicos o repetidos (arreglos). La creación y manejo de formas se basa en el manejo de relaciones uno a uno o uno a muchos entre tablas. Esto permite que sea más fácil la creación de formas.

El diccionario del 4GL probablemente soporte la definición y uso de dominios. Un dominio puede verse como un rango de valores que un campo puede tener. Por ejemplo, el RFC debe tener letras y números en posiciones específicas. Si se define un dominio para RFC todas las instancias del dominio RFC deberán apegarse a esta definición, lo que se conoce como integridad por dominios.

También puede ser posible el definir la integridad referencial en el diccionario o en las formas. Esto implica el verificar valores en diversas tablas de la base de datos. Específicamente, es la garantía de que el valor

de una llave foránea en una tabla tenga un valor correspondiente a una llave primaria en otra. Por ejemplo, una tabla de facturas tendrá una columna para el código del cliente el cual debe corresponder al valor de una llave primaria en la tabla de clientes (si pensamos que no puede existir una factura sin cliente).

Las vistas son tablas virtuales. Como el nombre sugiere, una vista se compone de una selección de campos en una o más tablas. Si una vista utiliza solo una tabla se denomina vista sencilla; vistas creadas a partir de varias tablas se denominan vistas complejas. En un sistema de captura de órdenes de venta sería posible construir una vista formada de campos de las tablas Cliente, Factura y Detalle con los campos necesarios para la transacción. Sin embargo, pocos 4GLs y bases de datos soportan la actualización en vistas complejas. Algunos 4GLs permiten definir las vistas en el diccionario.

El esquema relacional

El hecho de que la mayoría de los 4GL utilicen el modelo relacional como base para su arquitectura influye fuertemente en como se debe diseñar el sistema. La normalización se convierte en algo muy importante y el no realizarla al menos a la tercera forma dará lugar a problemas en el momento de programar el sistema.

3.6 Productos CASE

Varios 4GLs tienen un producto CASE (Computer-Aided Software Engineering) como es el caso de Oracle y Natural. El analista de sistemas utiliza la herramienta CASE para realizar la descomposición funcional y crear diagramas de estructura así como de flujo de datos. Estos pueden ser utilizados para generar código del 4GL utilizado.

El término "Upper CASE" es utilizado para describir a una herramienta CASE que tiene las funciones básicas de diagramación y llenado del diccionario de datos. El término "Lower CASE" se utiliza para las herramientas que se enfocan a la interface con el ambiente de desarrollo y la generación de código. Algunas herramientas CASE como las de Oracle, funcionan con ambos esquemas.

"Upper CASE" es la parte del ambiente de desarrollo utilizada por el analista. Es más rápido crear diagramas utilizando herramientas CASE y realizar modificaciones a estos; estas herramientas incluyen verificaciones de consistencia que ayudan a eliminar algunos errores de análisis.

Uno de los aspectos importantes de CASE es la metodología de análisis que el producto soporta, que en la mayoría de los casos no son más que convenciones de diagramación. Debido a que muchas empresas no utilizan metodologías reconocidas se ha abandonado CASE y se continúa implementando con métodos propios.

Además de la productividad que ofrece al analista, el mayor beneficio de usar CASE es que puede garantizar una documentación completa. Durante la etapa de diseño del proyecto el analista llena el diccionario de datos, el cual debe estar completo y listo para la etapa de construcción.

El mayor problema con las herramientas CASE es que la mayoría de éstas no ofrece una interface apropiada al ambiente de desarrollo. Comúnmente el diccionario que llena la herramienta CASE no es el mismo que se utiliza durante la construcción. Debido a esto muchas de las herramientas CASE incluyen la función de exportar/importar datos de otros diccionarios. Esto puede conducir a que existan muchos formatos, lo cual no es deseable ya que los diccionarios pueden desincronizarse. En proyectos grandes en donde existen varios equipos de analistas es esencial que el software CASE sea multiusuario, que todos los miembros del equipo accedan el mismo diccionario y que se implementen los esquemas de seguridad necesarios.

"Lower CASE" generalmente significa generación de código. La generación de código no es propia de las herramientas CASE, ya que muchos productos para diseño de pantallas comúnmente generan el código requerido para la forma o pantalla que se crea.

Existen límites serios en cuanto a la generación de código. Los problemas surgen cuando se hacen cambios al diseño básico a partir del cual se generó el código. Esto sucede comúnmente en un ambiente de prototyping, por lo cual con algunas herramientas no es posible agregar código a un diseño existente, lo que significa que la generación de código necesita ser supervisada y que se requiere atención extra en el diseño inicial.

Los productos CASE son una inversión costosa ya que además del costo del software necesitan equipos PC o workstations con muchos recursos.

3.7 Orientación a Objetos

Es un hecho que la próxima generación en tecnología de software y por lo tanto todos los productos basados en dicha tecnología tendrán orientación a objetos. Esto ya es notorio en varias áreas como es el caso de los ambientes gráficos, en donde varios proveedores ya han anunciado o están por anunciar nuevas versiones de sus productos con orientación a objetos (OO).

Algunas personas creen que los ambientes gráficos en donde existe una interface con mouse e iconos tienen orientación a objetos. En cierta forma lo son, ya que se pueden manipular objetos en la pantalla casi como si fueran objetos reales. Sin embargo esto no necesariamente significa que contengan funcionalidad OO. La orientación a objetos es en esencia una tecnología relacionada con lenguajes de programación, los cuales para poder ser catalogados como OO necesitan tener ciertas características.

Conceptos de Orientación a Objetos	
Estructura de Clases	Es un diccionario único que define y relaciona a todos los elementos del lenguaje OO.
Métodos y Encapsulación	Es la capacidad de definir métodos que apliquen a tipos de datos específicos y los encapsulen dentro de una clase.
Abstracción de Datos	Es la capacidad de definir nuevos tipos de datos y formarlos a partir de combinaciones de otros tipos de datos.
Herencia	Es la capacidad de un tipo de dato para heredar los métodos de otro.

4GLs comparados con lenguajes OO

Los lenguajes con orientación a objetos son más poderosos que los lenguajes de cuarta generación, sin embargo la curva de aprendizaje es grande, por lo cual no han sido ampliamente aceptados en ambientes corporativos.

4GL	Lenguaje OO
Diccionario	Jerarquia de Clases
Funcionalidad Limitada	Funcionalidad Extensible
Vocabulario Limitado	Vocabulario Extensible
Número limitado de tipos de datos	Cualquier tipo de dato
Reutilización de código limitada o no existente	Reutilización de código
No abarca el problema de mantenimiento directamente o en forma efectiva	Abarca los problemas de mantenimiento
Reduce el tiempo de programación (vs. 3GL)	Reduce el tiempo de programación (vs. 3GL por un factor menor)
Curva de aprendizaje corta	Curva de aprendizaje significativamente mayor debido a que se debe comprender la estructura de las clases
Tecnología bien desarrollada y madura	Tecnología no madura

Muchos proveedores de 4GLs están agregando capacidades OO a sus productos. Algunos como Ingres Windows 4GL, Dataflex y SmartStar Vision incluyen estructuras de clases OO, por lo que se puede decir que son implementaciones completas o parciales de orientación a objetos.

Lo anterior hace que la comparación de distintos productos sea complicada. Los 4GLs con orientación a objetos tendrán un mayor rango de funcionalidad, aunque sin garantizar que el ambiente de desarrollo que incluyen sea igual o más productivo que los 4GL tradicionales. Además es posible que la funcionalidad adicional resulte redundante en muchas aplicaciones.

Adicionalmente, debido a la dependencia de un ambiente gráfico o GUI, los requerimientos de CPU de un 4GL basado en objetos implicará el utilizarse solamente en una arquitectura cliente/servidor, limitando las opciones de migración a partir de ambientes tradicionales.

3.8 Productividad

A partir del surgimiento de los 4GL el aspecto de productividad ha sido cuestionado. Algunos afirman que en la práctica un 4GL ofrece a nivel general muy pequeñas mejoras en lo que respecta a productividad, entendiéndose a nivel general todo el proyecto, es decir análisis, diseño, pruebas y mantenimiento. Esto es opuesto a lo que señalan los proveedores de 4GLs, ya que sus estadísticas están basadas sólo en el aspecto de construcción de un proyecto. Por lo anterior, si una aplicación consiste en un simple mantenimiento a una tabla entonces la productividad para programar esa transacción tendrá un aumento de 10:1 o mayor comparada con un 3GL como COBOL.

Hasta hace poco, los fabricantes de 4GLs anunciaban las capacidades no procedurales de sus lenguajes como una ventaja. Aparentemente es una ventaja debido a que los lenguajes no procedurales son más fáciles de aprender. El concepto de estos lenguajes es que sólo es necesario especificar que se va a realizar sin necesidad de decir como se va a hacer.

Sin embargo en muchas ocasiones es más fácil manejar procesos complejos utilizando código procedural simplemente por el hecho de que comúnmente es necesario especificar como se debe obtener un resultado. Tratar de hacer funcionar un lenguaje no procedural como procedural puede tener resultados desastrosos.

El 4GL proporciona una forma rápida de codificar una transacción utilizando pocos comandos, lo cual es adecuado en muchos casos. Sin embargo hay ocasiones en las cuales el programador debe implementar una forma no estándar para resolver cierto problema. Existe una tendencia a que los 4GLs incluyan el soporte a construcciones procedurales (por ejemplo PL/SQL de Oracle).

Lenguajes basados en Eventos

Las ventajas de un lenguaje no procedural y la flexibilidad de un lenguaje procedural pueden ser combinadas en un ambiente basado en eventos. El concepto es que el componente para el manejo de formas del 4GL contenga formatos estándares para diferentes tipos de transacciones. Una vez que la forma se diseña ésta genera el código necesario o bien proporciona el módulo objeto que realiza la tarea. Dentro de éste es posible agregar partes de código fuente para tener mayor funcionalidad

(código procedural). Un ambiente basado en eventos es menos restrictivo que un ambiente totalmente no procedural. Sin embargo pueden existir diversos problemas, entre ellos la dificultad de depurar el código. El programador puede saber que es lo que supuestamente hace el código que escribió, pero no sabe exactamente como trabaja el código generado.

Diccionario

El uso de un diccionario es crucial en la productividad. La definición central de tablas, campos, dominios y formas en un diccionario permiten crear valores por omisión para ser utilizados por el 4GL con lo cual sólo es necesario codificar casos diferentes. Algunos diccionarios permiten definir integridad referencial y almacenar procedimientos. El diccionario es la parte clave en lo que respecta a productividad.

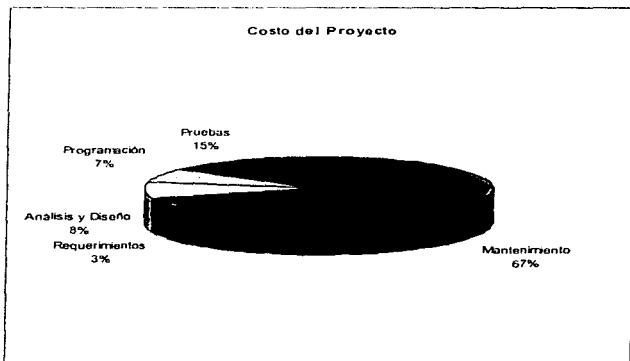
Ambiente de Desarrollo

La mayoría de los 4GLs incluyen un ambiente de desarrollo que proporciona distintas funcionalidades para el programador, como el permitirle ir de un diseño a otro, edición de código, utilización del diccionario así como probar y depurar programas. Estos ambientes de desarrollo proporcionan aumentos en productividad, aunque varios 3GLs también proporcionan excelentes ambientes de desarrollo, como es el caso del lenguaje C de Microsoft.

Una de las características que muchos ambientes de desarrollo no incluyen es un depurador o debugger completo que permita seguir el código por pasos, desplegar valores de variables o crear break-points. Estas características son esenciales en el desarrollo de sistemas complejos.

Ciclo de vida del proyecto

Vale la pena analizar el aspecto productividad con un enfoque de todo el proyecto. Los 4GLs están orientados a mejorar la velocidad de desarrollo, pero esto afecta sólo una de las tareas necesarias para terminar un proyecto. La siguiente gráfica muestra un porcentaje estimado del costo de las distintas etapas de un proyecto.



Fuente: Ready Systems Inc.

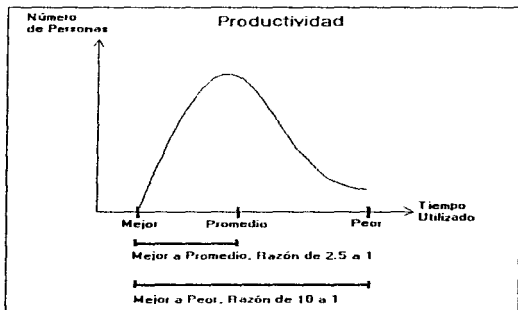
Es claro que la etapa de desarrollo de los programas no constituye la mayor parte de un proyecto, sólo un 7% del total. Los 4GLs contribuyen a otras etapas como la creación de prototipos en la fase de análisis y diseño. Sin embargo uno de los mayores problemas reside en el mantenimiento del sistema, que es un aspecto muy serio en ambientes corporativos ya que gran parte del presupuesto de sistemas (se calcula que alrededor del 60%) se consume en estas actividades. No existe evidencia que un 4GL ayude a disminuir estos costos.

Habilidades de Programación

Se puede medir la productividad con velocidad y calidad. Si la velocidad de construcción es buena pero la calidad deficiente entonces el producto final tiene que ser revisado. Un 4GL u otro tipo de lenguaje o ambiente de desarrollo no convertirá en buenos programadores a personas que no saben programar.

En muchos casos las compañías no invierten suficiente para capacitar a su gente. Muchos 4GLs se venden con la promesa de facilidad de uso y ser amigables al usuario, lo cual en ocasiones es interpretado como que no es necesario invertir en capacitación. A pesar de que la mayoría de los productos son fáciles de usar, muchas veces no resulta sencillo utilizarlos exitosamente sin el entrenamiento y la práctica adecuada. Lo anterior se debe a que se necesita ser consciente de las implicaciones técnicas en el uso del producto, lo cual requiere invertir en capacitación. El construir un sistema deficiente rápidamente no significa que se tenga una ganancia en productividad.

La productividad varía mucho dependiendo del programador. La siguiente gráfica muestra la variación estándar en la productividad asociada a cualquier grupo grande de trabajadores que realizan cualquier tarea que requiere cierto grado de habilidad. (Básicamente se trata de una distribución Poisson). Según investigaciones hechas por Tom DeMarco y Timothy Lister, explicadas en su libro "Peopleware", la productividad de los programadores también se apega a esta curva.



Este estudio indica que los programadores con mejor rendimiento sobrepasaron al promedio por una razón de 2.5 a 1 y superaron a los peores por una razón de 10 a 1. De acuerdo con DeMarco lo anterior aplica independientemente del lenguaje de programación. También afirma

que la calidad es más o menos constante en toda la curva, lo que significa que aquellos con gran rendimiento no lo hacen sacrificando calidad. Otro resultado interesante de este estudio es que parece existir una tendencia a que estos programadores de alto rendimiento se junten en las mismas compañías.

En 1984 DeMarco estimaba que las ganancias en productividad asociadas con un 4GL eran de un 15% sobre todo el proyecto. Actualmente este porcentaje se ha incrementado hasta alcanzar el doble o el triple.

Sin duda los 4GL proporcionan una ganancia en productividad. Con aplicaciones relativamente simples, un 4GL generalmente incrementa la productividad. Para sistemas complejos se requiere dedicar atención a aspectos de análisis y diseño, y posiblemente complementar algunas funciones con un 3GL.

3.9 Performance

Es sabido que los usuarios de lenguajes de cuarta generación generalmente tienen que sacrificar cierto nivel de eficiencia en comparación con el mismo sistema escrito con un 3GL. La sobrecarga en proceso asociada con un 4GL va de 3 a 4 veces la de un 3GL si es interpretado. Cuando el lenguaje es compilado es más o menos equivalente.

Estos datos son resultado de lo que se ha descubierto en compañías grandes al realizar pruebas con varios 4GLs y un sistema escrito en un 3GL. Sin embargo el esquema no es tan simple: Algunos 4GLs implementan una etapa intermedia (código con tokens) que prácticamente es código parcialmente interpretado y parcialmente compilado. La rutinas ligadas al compilar generalmente son muy eficientes, por lo cual muchos productos pueden proporcionar un rendimiento muy parecido a un 3GL. En 4GLs basados en compilador el performance depende de la habilidad del creador del compilador y la eficiencia de las rutinas comunes.

Con un 4GL basado en intérprete cada instrucción debe ser traducida por el CPU antes de ejecutarse, lo cual causa una carga extra y genera un bajo performance. Típicamente el código que demanda mucho proceso de CPU, como es el caso de rutinas matemáticas (inclusive las más simples), puede ocasionar pésimo rendimiento si se escribe en un 4GL. Sin embargo conviene notar que en ambientes cliente/servidor la interpretación de código causa menos sobrecarga debido a que es más probable que existan MIPs extra.

Se puede controlar el impacto en rendimiento de un 4GL poniendo atención en:

- 1) La habilidad del programador y diseñador del sistema.
- 2) El diseño de la base de datos para el sistema.
- 3) Combinación inadecuada de cargas de trabajo (por ejemplo, actividades pesadas de reportes y captura de datos).
- 4) El grado en que se permite a los usuarios finales el generar partes del sistema (por ejemplo las herramientas para crear reportes).
- 5) La estrategia en el uso de candados utilizada por el 4GL.

Candados

Es importante señalar que algunos 4GLs implementan los candados en una forma muy pobre. La mejor estrategia son los candados suaves (soft locking), la cual es difícil que cause problemas en cuanto al tiempo de respuesta y garantiza consistencia. Esta estrategia realmente no crea un candado sino que verifica si un registro ha cambiado a la hora de realizar la actualización. Sin embargo en una tabla pequeña que es frecuentemente actualizada, esta estrategia no evitará problemas de tiempo de respuesta. Estas situaciones implican el rediseñar las transacciones.

Algunos 4GLs crean candados cuando se accesa un registro para una posible actualización. Esto puede causar retrasos considerables a otros usuarios si están compitiendo por el mismo registro y se sabe que puede ocasionar que algunos sistemas se paren.

Portabilidad

Para que un 4GL tenga buen rendimiento necesita afinarse especialmente para el software y hardware con el cual se utilizará, aunque esto dificulta el implementar la portabilidad que tanto se anuncia actualmente.

El costo de desarrollo y soporte de productos para cada plataforma es elevado, lo cual causa que algunos proveedores escojan soluciones simples a expensas del performance sin diseñar el producto específicamente para cada ambiente. Un ejemplo es el caso de Oracle, que utiliza tipos de datos propios independientemente de la plataforma. Si estos no coinciden con los tipos de datos del hardware nativo es necesario realizar algún tipo de traducción o mapeo al momento de ejecución del sistema, lo cual causa cierta sobrecarga al CPU. En ocasiones entre más portable es un producto existen mayores probabilidades de que su performance sea malo.

Costos de hardware

Los proveedores de 4GLs argumentan que el costo del hardware extra necesario no es nada comparado con las ventajas de tener un sistema listo en una fracción del tiempo que se llevaría implementarlo con un 3GL. Este argumento es válido en casos donde se obtengan grandes mejoras en productividad. Sin embargo la productividad no está garantizada y en algunos casos el gasto en hardware extra puede ser mayor a lo esperado.

Los ambientes cliente/servidor ayudan en gran forma a solucionar el problema anterior, ya que es posible implementar y ejecutar rutinas como la interface al usuario en procesadores económicos (como es el caso de las PC), lo cual hace a un 4GL una mejor solución para implementar sistemas grandes en línea. Por lo anterior se puede considerar el uso de un 4GL sin muy buen rendimiento sólo en configuraciones cliente/servidor y después de probar sus características en dicho ambiente.

Arquitectura de hardware

Al comprar un mainframe se adquiere una máquina capaz de manejar grandes cantidades de memoria, muchos gigabytes en disco y ser utilizado como servidor en sistemas sofisticados. A pesar de la reducción del precio vs. poder de proceso, los mainframes siguen siendo equipos que pueden manejar grandes cantidades de datos debido a que así se diseñaron. Se incluye en esta categoría a hardware poderoso para UNIX como el ofrecido por Sequent y NCR (ahora AT&T GIS). En estas máquinas el costo del hardware resulta barato comparado con un mainframe pero todavía muy caro comparado con una PC o workstation.

Este esquema proporciona una doble ventaja. Si se dedica un gran recurso para el manejo de datos será más fácil afinarlo para esta función, lo cual ampliará su rendimiento. La carga de trabajo asociada con el manejo de formas y acceso a datos no es compatible y por lo tanto difícil de balancear. Es mejor que cierta funcionalidad se ejecute en el CPU cliente, en donde existe la suficiente potencia para manejar la interface, inclusive si ésta es gráfica.

Lo anterior es la razón por la cual una arquitectura cliente/servidor es tan importante para un 4GL y porqué la mayoría de los proveedores la soportan y promueven su uso.

En el siguiente capítulo se analizarán distintas consideraciones y aspectos relacionados con el esquema o arquitectura cliente/servidor.

Tecnología Cliente/Servidor

**Todo debe realizarse de la forma
más sencilla posible, pero no la
más simple.**

Albert Einstein

4.1 Introducción

La tecnología cliente/servidor es uno de los desarrollos más importantes de la década de los 90s en lo que respecta a computación. Ayudada por el aumento del poder de cómputo en los equipos PC, mayor uso de redes LAN, diversas formas de acceso a bases de datos así como interfaces gráficas más poderosas, cliente/servidor es uno de los esquemas más utilizados por organizaciones que realizan un downsizing o que están desarrollando nuevos sistemas.

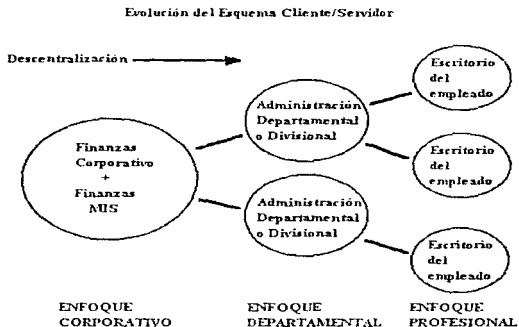
En un ambiente cliente/servidor las estaciones de trabajo o equipos PC solicitan servicios a uno o más servidores en la red. Esto permite al equipo cliente realizar funciones más adecuadas como manejar la interface con el usuario o bien iniciar el acceso a una base de datos. Mientras tanto el servidor controla todas las actividades relacionadas con agregar, borrar y almacenar información. El servidor también puede realizar procedimientos específicos de acuerdo a las necesidades de procesamiento.

La ventaja de esta arquitectura es que permite al usuario combinar las características de una interface gráfica amigable con la flexibilidad de una computadora personal y el poder de una base de datos multiusuario o servidores de aplicaciones. El surgimiento de estándares para acceso a bases de datos, como es el caso de SQL aumenta las opciones de portabilidad y mayor uso de recursos distribuidos.

El objetivo de este capítulo es el mostrar los componentes más representativos de un esquema cliente/servidor para poder determinar cuáles son las ventajas y desventajas que puede tener una implementación de este tipo.

4.2 Orígenes de la tecnología cliente/servidor

Como ya se mencionó anteriormente, cliente/servidor representa uno de los cambios más importantes de la década. Cada vez es mayor el número de empresas privadas y dependencias de gobierno que comienzan proyectos de transición a nueva tecnología, por lo cual se ven en la necesidad de convertir sistemas centralizados a sistemas distribuidos para implementar una arquitectura cliente/servidor.



Las promesas de esta tecnología son muchas. Los procesadores tipo RISC, SPARC o x86 ofrecen opciones atractivas con precios competitivos en relación a los equipos mainframe, aunque cabe notar que independientemente del poder de procesamiento cada plataforma tiene un diseño específico que la hace ideal para determinado tipo de procesamiento.

Los dispositivos servidores y clientes proporcionan gran flexibilidad. Bajo un esquema bien diseñado e implementado, el descentralizar recursos evitando la dependencia de un solo equipo proporciona confiabilidad. El

contar con aplicaciones que se ajusten al hardware y software más apropiado proporciona un ahorro en costos.

Cliente/servidor es la implementación de todo el hardware, software, procesamiento y nueva tecnología como puede ser gran capacidad de proceso, recursos distribuidos y asignación modular de tareas. Consiste de elementos clave como LANs con la capacidad de integrarse entre sí, GUIs con interface sencilla de usar, módulos de bases de datos distribuidas para acceso eficiente a la información, SQL para acceso a dichas bases de datos y distintos tipos de herramientas para ayudar a diseñar el modelo cliente/servidor.

Cliente/servidor no pertenece a ningún proveedor en especial. Es una arquitectura cuyo principio básico incluye estaciones de trabajo (clientes) conectados a repositorios de datos (servidores) a través de una red (típicamente una LAN). Representa un completo rediseño de la interacción entre el usuario, la información y la tecnología, en ocasiones cambiando el modelo operacional de la empresa.

Lo anterior es una de las razones por las cuales los integradores de sistemas tienen una gran perspectiva de negocio, ya que muchas organizaciones carecen de la experiencia necesaria para realizar una transición exitosa por sí mismas.

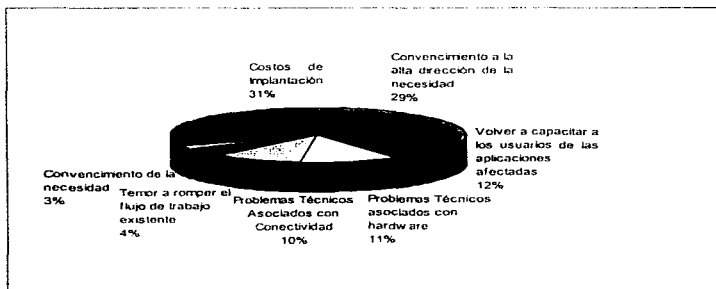
Otros factores que refuerzan la validez de esta tecnología son:

- Proveedores importantes en plataformas IBM mainframe como Computer Associates y Software AG han hecho revisiones de sus productos para soportar esquemas cliente/servidor.
- Diversos tipos de herramientas para administración de bases de datos distribuidas y herramientas de desarrollo ayudan de cierta forma a aquellos usuarios que desean migrar de un ambiente centralizado. Oracle y Sybase son ejemplos de algunos proveedores.
- Windows NT, OS/2, UNIX System V R4.2 y otros sistemas operativos contienen características que facilitan la existencia de clientes y servidores.
- Algunas soluciones cliente/servidor no tan recientes como DEC Pathworks para PC y Macintosh han probado ser viables.

- Cada vez mejora la tecnología en servidores ha mejorado al incluir configuraciones de multiprocesadores y arreglos de discos, lo cual reduce la vulnerabilidad de estos sistemas.

Gracias al crecimiento de las computadoras personales, estaciones de trabajo poderosas, redes avanzadas y bases de datos distribuidas, la tecnología cliente/servidor está siendo ampliamente implementada. Existen impedimentos fuertes para la implementación de esta tecnología, entre ellos el costo inicial y aspectos culturales como la resistencia al cambio.

Factores que retrasan la aceptación de un esquema Cliente/Servidor



4.3 Redes de área local

Un componente indispensable en esquemas cliente/servidor es el medio de comunicación. A pesar de que otros elementos como bridges, routers y brouters se utilizan para interconectar distintas redes, es la tecnología LAN (Local Area Networks) la que distribuye los datos entre clientes y servidores.

Las redes de área amplia o WAN (Wide Area Network) también juegan un papel importante en el caso de tener servidores dispersos en una área geográfica extensa, aunque las LANs típicamente son la parte central en comunicaciones cliente/servidor. La siguiente tabla muestra algunas de las principales tecnologías empleadas para implementar este tipo de redes:

- A. Redes
Ethernet, Token-Ring, LocalTalk, X.25, FDDI, Frame Relay, ATM
- B. Sistemas Operativos de Red
Banyan VINES, Microsoft LAN Manager, IBM LAN Server, Novell NetWare, Apple LocalTalk, DEC Pathworks, Sitka 10Net.
- C. Protocolos
OSI, TCP/IP, NetBIOS, X.25, ATM, AppleTalk, DECnet, LU6.2 (APPC), GOSIP, MAP/TOP.
- D. Internetworking
 1. Hubs Inteligentes
 2. Routers - IBM, DEC, Wellfleet, cisco Systems.

Un estudio hecho por IDC (International Data Corporation) muestra que más de 400,000 sistemas operativos de red fueron vendidos en 1991, significando más de mil millones de dólares.

Hasta antes del surgimiento de Windows NT, NetWare de Novell tenía el 58% del mercado. Otros sistemas operativos incluyen a VINES de Banyan, LAN Manager de Microsoft, Pathworks de DEC (un derivado de LAN Manager) y LAN Server de IBM. Este grupo tenía alrededor de 30% del mercado. Todos estos productos son sistemas operativos de servidor

dedicado, ya que asignan varios recursos exclusivamente para la función de servidor.

El otro 10% del mercado estaba ocupado por los sistemas operativos punto a punto (peer-to-peer). Tradicionalmente orientados a configuraciones pequeñas, sus nodos pueden trabajar como clientes o servidores (típicamente funcionan con MS-DOS). Algunos de estos productos incluyen LANStep de Hayes Microcomputer Products, 10Net de Sitka Corporation, PowerLAN de Performance Technology Inc, entre otros.

La ventaja de las redes punto a punto contra el esquema de servidores dedicados radica en la flexibilidad y el ahorro potencial en costos de hardware y software. Sin embargo se tienen desventajas como la administración y el tiempo de respuesta, ya que en redes punto a punto éste se incrementa demasiado conforme el número de usuarios aumenta.

4.4 SQL

Actualmente la información contenida en una base de datos es uno de los recursos más valiosos de una empresa. Con el énfasis en proyectos de downsizing y descentralización, el acceso a estos datos cada vez será desde las computadoras personales y estaciones de trabajo hacia los servidores. Aplicaciones dispersas en diferentes puestos de trabajo interactúan con las bases de datos corporativas, que pueden residir en servidores LAN, sistemas medianos como AS/400, UNIX o VAX, o bien en mainframes actuando como superservidores.

La interacción entre los usuarios y las bases de datos debe realizarse de manera uniforme. Como se vio en el capítulo de bases de datos, IBM desarrolla SQL en 1972 con la finalidad de proporcionar un lenguaje estándar para consulta de bases de datos relacionales. SQL proporciona un método conciso, no procedural, para consultas. En un ambiente cliente/servidor una aplicación en el cliente transmite un comando SQL al servidor quien a su vez manda de regreso los resultados de dicho comando.

A pesar de haberse concebido como un medio estándar, el SQL actual difiere dependiendo del proveedor, ya que éstos han agregado extensiones al lenguaje, con lo cual la interoperabilidad se ha vuelto problemática.

4.5 GUI

Otro elemento clave en la tecnología cliente/servidor es la interface gráfica. La GUI es la interface a la red, en donde muchas de las funciones son realizadas por componentes intermedios denominados middleware.

Los GUI son imágenes tipo bitmap que sirven como front-end a un sistema operativo. Su funcionalidad incluye iconos que representan aplicaciones y recursos, menús y diálogos. El ambiente está formado por varias entidades en forma de ventanas de múltiples tamaños.

Los ambientes de este tipo más populares son Windows de Microsoft, Motif de OSF, Open Look de Sunsoft y USL y Presentation Manager de IBM.

Los GUIs ofrecen mayores posibilidades en la interacción con el usuario en un ambiente cliente/servidor. Los servicios y la información están disponibles en los equipos personales y se presentan en una forma más comprensible para el usuario. A diferencia de las pantallas tipo caracter que se enfocan en la línea de comando y texto asociado, un GUI proporciona distintas imágenes gráficas que junto con la disposición de los datos facilita el trabajo.

El diseño de esta interfaces contempla la manipulación de múltiples aplicaciones en distintas ventanas ayudada por comandos iniciados por la selección de un icono, con lo cual dispositivos como el mouse o similares sustituyen en muchas ocasiones a ciertas funciones que anteriormente se realizaban con el teclado. Lo anterior permite la ejecución de tareas en forma menos estructurada, lo cual resulta en un manejo más intuitivo de las funciones disponibles.

4.6 Administración de la red

Tradicionalmente se considera como la parte más débil en cuestiones de seguridad, aunque cada vez los productos y soluciones para realizar esta función mejoran. Sin embargo en los ambientes distribuidos las complicaciones parecen surgir más rápido que las soluciones.

Uno de los objetivos de la administración de la red es monitorear eventos. Por ejemplo, la disponibilidad de recursos en la red es información que debe ser capturada y evaluada.

Otro objetivo es el manejar las fallas conforme estas aparecen. Esto debe incluir opciones como el poder ignorar o aislar las unidades que tengan problemas o el ejecutar procedimientos correctivos en el momento dependiendo de la naturaleza del problema encontrado.

La administración de la red también debe incluir una bitácora. Esta información puede utilizarse posteriormente con aplicaciones basadas en sistemas expertos para monitoreo de recursos.

La integración de datos y algoritmos de respuesta es otro objetivo del control de la red. Debe tenerse acceso a paquetes aislados de información sobre la red que permitan mayor análisis de su comportamiento. Por ejemplo, cuando ocurre una falla, se debe presentar un panorama completo al personal técnico sobre dónde, qué, quién y cuando puede ser corregido el problema.

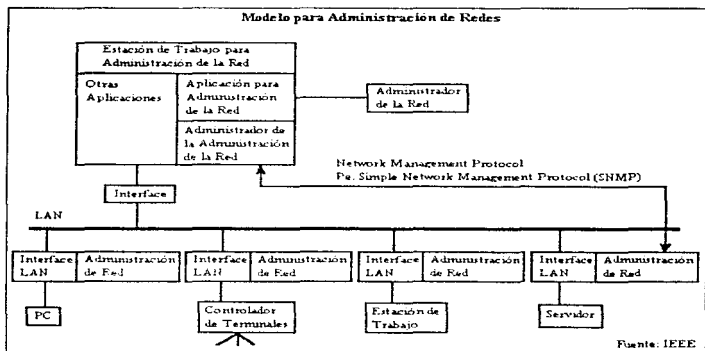
Una definición de la funcionalidad para la administración de una red se encuentra en la especificación CMIS/CMIP o Common Management Information Services/Command Management Information Protocol creada por ISO. Este estándar está basado en la especificación de red creada por el OSI, la cual no es todavía implementada ampliamente.

El ISO ha identificado cinco funciones básicas para la administración de red en una configuración empresarial. Pocos productos incluyen todas las funciones.

- **Manejo de Fallas.** La detección y corrección de fallas en cualquier lugar de la red se realiza en este nivel. Esto incluye hardware y protocolos. Cuando la corrección no se puede realizar en forma inmediata se debe iniciar un aislamiento del problema.

- **Administración del Rendimiento.** Esta capacidad ayuda a personal encargado de la red a monitorear, modificar y controlar la utilización de recursos, incluyendo el performance, el tiempo de respuesta promedio y el flujo de datos en general. Es una herramienta indispensable para los administradores del sistema.
- **Administración de la Configuración.** Comúnmente presentada a los usuarios en forma gráfica, la imagen visual de toda la red también puede indicar el estado actual de cada componente mayor de ésta. Se les de nombre a estos componentes con el objeto de lograr una mejor comprensión de estos esquemas.
- **Administración de la Seguridad.** Esta característica se encarga de procurar la integridad de los datos en la red. No todos los productos, especialmente los que manejan arquitecturas distribuidas, son fuertes en esta área. Existen controles de seguridad en los distintos niveles de procesamiento, es decir, a nivel sistema operativo, base de datos o interface de usuario.
- **Contabilidad.** Como la palabra lo indica, mide el uso promedio y costos de la red. Al igual que la seguridad, estos servicios están disponibles en varios niveles. Pueden surgir de aplicaciones aisladas, ser parte de servicios del sistema operativo, por ejemplo.

La administración en redes LAN es una tecnología más volátil en comparación con la centralizada o basada en un host. Los proveedores de LANs y desarrolladores independientes trabajan en mejorar sus sistemas y la interacción de éstos con los sistemas de administración corporativos. Como se muestra en el esquema, la administración de una LAN abarca varios elementos:



A pesar de que los sistemas operativos para LAN ofrecen algunas funciones de administración, todavía les falta mucho para alcanzar una funcionalidad completa. Los datos históricos creados para análisis detallado son insuficientes. Las señales de alarma o aviso en caso de problemas en la red no son completas. Los principales proveedores como Novell y Microsoft todavía no ofrecen todas las características necesarias para una administración completa y segura de la red, especialmente en el control remoto del sistema y la integración con otros productos y aplicaciones.

Uno de los problemas en la administración de redes es la variedad de protocolos. Aparte de los protocolos propietarios de soluciones anteriores existen dos estándares en evolución totalmente incompatibles. La especificación CMIS/CMIP de ISO ha sido desarrollada para un ambiente OSI y la especificación SNMP, creada por la comunidad de Internet, que actualmente se hace más común entre los productos comerciales existentes.

SNMP ha crecido en popularidad debido a que está disponible, funciona y porque es mayor el número de usuarios de Internet con el esquema TCP/IP en comparación con los promotores del modelo OSI que todavía esperan la aceptación general de esta tecnología.

4.7 Rendimiento

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Una de las ventajas fundamentales de cliente/servidor es la armonía del flujo de información ente los distintos procesos. El hardware que se encuentra en cada escritorio, típicamente una computadora personal, permite a los usuarios acceso en tiempo real a información y recursos de cómputo que les permiten desarrollar mejor su trabajo. Cliente/servidor también proporciona una influencia de integración a nivel corporativo.

Al evaluar las ventajas de cliente/servidor, la flexibilidad mencionada anteriormente y otras capacidades de integración son determinantes. El costo y la confiabilidad también son factores de importancia. Sin embargo en la mayoría de las ocasiones la aceptación está en función del rendimiento tanto a nivel cualitativo como a nivel cuantitativo.

Los servicios que pueden ser soportados en la red y que mejoran el rendimiento incluyen funciones como captura de órdenes, servicio a clientes, comunicación entre departamentos, situación de la producción y entrega, entre otros. El desarrollo, prueba e implementación de aplicaciones se mejora gracias a interfaces más completas y herramientas de desarrollo más potentes.

Existen varios casos en los cuales se han mejorado los tiempos de respuesta al migrar de un esquema mainframe o centralizado a una arquitectura cliente/servidor. Una casa de bolsa en Estados Unidos tuvo un cambio de 11 o 10 minutos por transacción a 2 minutos con el nuevo esquema.

El proceso distribuido implementado en configuraciones cliente/servidor logra que la información esté más cerca de quien la necesita. Una arquitectura cliente/servidor correctamente diseñada e implementada siempre redituará en mejoras en rendimiento.

Es importante enfatizar lo anterior, ya que a pesar de todas las promesas de esta tecnología también existen muchos casos en donde se ha subestimado la necesidad de un análisis y diseño adecuado. La mala implementación de un esquema cliente/servidor puede resultar contraproducente.

4.8 Downsizing

Antes de definir que es downsizing, es importante establecer qué no es. Downsizing no es un simple reemplazo del mainframe. El objetivo es crear una solución de negocio utilizando tecnología nueva. La sustitución del mainframe puede estar considerada pero no debe ser el único factor de atención. De hecho algunos diseñadores están a favor de mantener estos equipos y sólo cambiar su papel al de un superservidor, pensando que el caso del negocio justifica este tipo de solución.

Downsizing como concepto tiene muchas definiciones como en el caso de cliente/servidor. Puede ser visto como la migración de un esquema mainframe altamente centralizado a dispositivos más económicos como estaciones de trabajo o equipos personales. Como se mencionó anteriormente, esto no significa que sea imperativo la sustitución del mainframe, aunque es un hecho que la función de esta plataforma debe cambiar para que siga formando parte de la solución del negocio.

Uno de los motivadores más fuertes del downsizing es el contraste entre el costo de proceso entre plataformas: El costo de procesar una aplicación en una PC es aproximadamente de 1% del equivalente en un mainframe. Es cierto que el utilizar un mainframe proporciona otros elementos de interés, sin embargo es imposible ignorar la relación 100 a 1 entre plataformas.

Conforme avanza la tecnología, el tema de downsizing ha sufrido variaciones entre las cuales están:

- En ocasiones el primer paso al downsizing literalmente involucra el reducir de tamaño los componentes de hardware, por ejemplo, cambiar de un procesamiento en mainframe a procesamiento en una LAN. Las aplicaciones corren en PCs y sistemas operativos de red como NetWare o LAN Manager en un ambiente de servidor de archivos. Lo anterior no es una implementación cliente/servidor típica debido a que la funcionalidad de servidor de archivos se reduce a simular un gran disco local, sin realizar tareas como seguridad, administración de red, o control del RDBMS. En ocasiones se realizan algunas de estas tareas pero en forma deficiente o no apropiada para un ambiente corporativo. Conforme el poder de cómputo es mayor y este tipo de productos maduran las deficiencias anteriores son corregidas.

- Después de realizar un proyecto de downsizing se logra un ambiente cliente/servidor real al incluir un manejador de bases de datos. Las aplicaciones se ejecutan en servidores o en PCs. El sistema operativo de red realiza algunas tareas de administración y el DBMS controla todos los aspectos de la información. Es común encontrar la utilización de protocolos estandar como TCP/IP; compañías como Sequent y Pyramid fabrican potentes servidores de datos. El sistema operativo de red puede ser NetWare, Banyan VINES u otro similar. Sistemas operativos como Windows, MS-DOS, OS/2 o inclusive UNIX soportan los dispositivos cliente.
- Otra estrategia para downsizing puede involucrar al mainframe como la plataforma en donde estará el DBMS. Redes LAN o PCs aisladas pueden funcionar como front-ends a la base de datos. Es difícil categorizar esto como downsizing, ya que la configuración actual puede terminar "upsized". Lo anterior explica el surgimiento de términos como rightsizing.

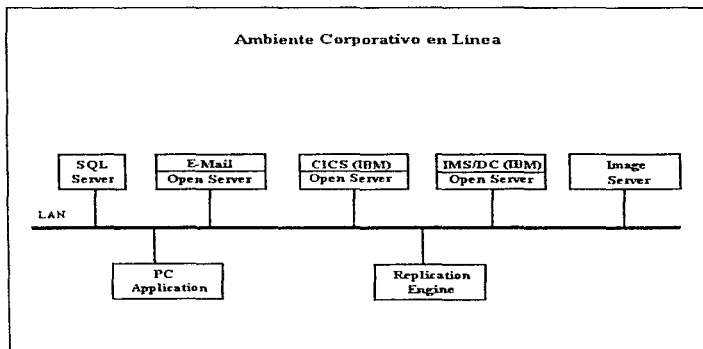
Estas son sólo algunas de las posibles estrategias. Se debe evaluar y seleccionar la que más se adapte al modelo de negocio. Vale la pena mencionar que independientemente del esquema implementado siempre la tendencia será a tener altos costos de implementación, los cuales, de diseñarse adecuadamente la solución, tenderán a bajar o estabilizarse en determinado plazo.

4.9 Desarrollo de Aplicaciones

Esta es un área clave en el tema de cliente/servidor. Los modelos tecnológicos y diseños de arquitectura son importantes, pero el desarrollo de las aplicaciones es la parte sensible del esquema.

En los ambientes distribuidos generalmente las aplicaciones corren en un servidor o en una estación de trabajo. La tecnología cliente/servidor puede elevar la distribución de actividad asociada con la ejecución de una aplicación. Por ejemplo, diferentes partes de una aplicación pueden ser ejecutadas en distintos nodos de la red mientras que para el usuario esta división permanece transparente.

Para poder crear aplicaciones distribuidas es necesario que los desarrolladores cuenten con herramientas especiales de programación y conectividad. El software de comunicaciones tradicional resulta inadecuado ya que es necesario un esquema para la interacción y control entre aplicaciones así como un soporte de comunicaciones básico. Es posible el utilizar herramientas CASE que sean útiles en el proceso de desarrollo de aplicaciones, especialmente cuando existen distintas configuraciones como se muestra en la figura:



La especificación DCE creada por OSF y apoyada por IBM, DEC y otros es la más utilizada actualmente para el desarrollo de aplicaciones distribuidas. El ONC (Open Network Computing) de Sun Microsystems también es fuerte en esta área. Para soportar aplicaciones distribuidas es necesario contar con lo siguiente:

- Los segmentos de una aplicación que esté en la red deben ser capaces de llamar a otros segmentos que se encuentren en otra parte de la red.
- Los segmentos deben ser capaces de ejercer control sobre otros segmentos en la red.
- Los segmentos deben ser capaces de intercambiar datos.
- Los segmentos deben ser capaces de compartir información de control con otros segmentos en la red para determinar el estado actual de todo el ambiente de proceso.

4.10 OLTP

Las funciones típicas de un ambiente OLTP incluyen aquellas que dependen de una respuesta rápida cuando existe un gran número de usuarios, transacciones que involucran acceso frecuente y en línea a la base de datos manteniendo la integridad de éstos. Las aplicaciones OLTP son frecuentemente asociadas con tareas de misión crítica.

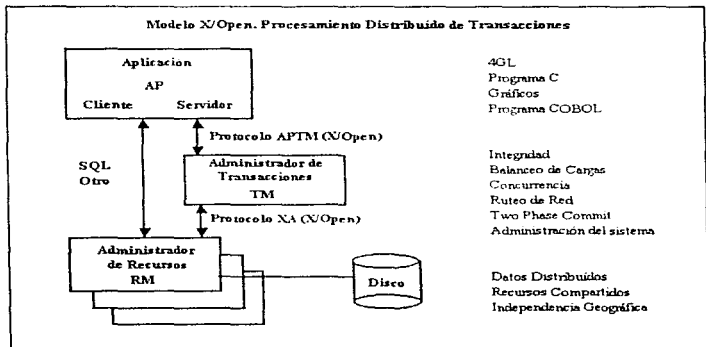
Un ejemplo es el sistema de cajeros de cualquier banco. El cajero debe funcionar los siete días de la semana, 24 horas al día. Es raro que no funcionen y no deben permitir errores.

Los requerimientos de un ambiente OLTP en lo referente a integridad de datos, recuperación a fallas, control y rendimiento son más estrictos en comparación con un proceso convencional. Aplicaciones como captura de órdenes, control de inventario, administración financiera y manufactura requieren una implementación OLTP.

Existen cuatro métricas para determinar la validez de un sistema OLTP:

- **Integridad.** Las transacciones que alteran un recurso del sistema, como una base de datos, deben funcionar en un esquema todo o nada, de tal forma que si se cancela la transacción la base de datos debe regresar exactamente a su estado original.
- **Consistencia.** Una transacción debe cambiar a la base de datos hacia el siguiente estado válido. No es permisible tener cambios parciales o que la base de datos se quede en el "limbo".
- **Aislamiento.** Una transacción impacta solo su ambiente. Múltiples transacciones son transparentes entre ellas y no pueden tener efecto en ninguna otra transacción.
- **Durabilidad.** El manejo exitoso de las transacciones proporciona cambios permanentes a los datos. Fallas del sistema subsecuentes no pueden regresar o retroceder procesos previos.

Donde existe tecnología comúnmente existe un estándar. Para OLTP este estándar surge de la organización de estándares X/Open. El modelo DTP (Distributed Transaction Processing) de X/Open soporta el desarrollo de aplicaciones OLTP.



4.11 Seguridad

Los beneficios de la tecnología cliente/servidor (acceso global, información oportuna, etc) así como la distribución de los datos, ha llegado a causar severos problemas de seguridad. Las posibles soluciones a este problema todavía están en evolución, incluyendo la disponibilidad de herramientas. Debido a la naturaleza del esquema cliente/servidor se debe contar con un plan de seguridad basado en distintos niveles dentro de la organización.

Es necesario considerar la seguridad ya sea a nivel aplicación, base de datos, red o sistema operativo. No existen muchos estándares ampliamente aceptados, por lo cual las soluciones muchas veces son a la medida y varían dependiendo de cada ambiente.

Cada compañía implementa controles de seguridad al nivel más bajo posible. Esto evita al usuario problemas de acceso al sistema, degradación del performance así como la reducción en costo que implica la seguridad en cada operación realizada.

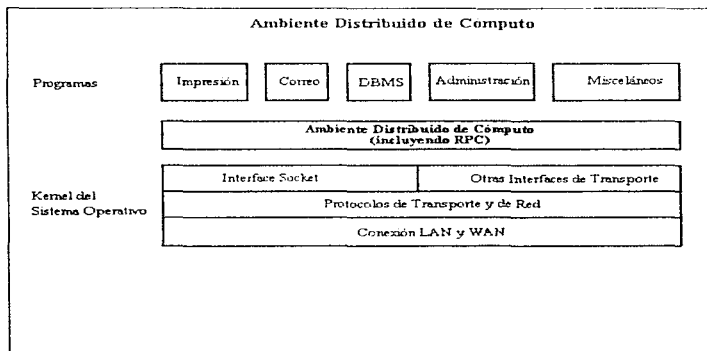
En algunas empresas se ha tomado la costosa decisión de asignar diferentes redes físicas a distintos grupos de usuarios para incrementar el aislamiento de los datos. Otras compañías están en espera de mejores soluciones por parte de los proveedores. Algunas implementan sus propios estándares y modelos de seguridad.

El mercado de tecnología cliente/servidor todavía no está maduro. Los proveedores comienzan a crear productos para seguridad conforme los usuarios se van dando cuenta de la vulnerabilidad de la información corporativa. Las empresas creadoras de bases de datos parecen ser las más indicadas para crear y ofrecer las herramientas necesarias.

4.12 RPC

Los RPC o Remote Procedure Calls fueron diseñados para facilitar la creación de aplicaciones distribuidas. Para el usuario un RPC esconde la complejidad de la red aparentando que los procedimientos se encuentran en forma local. La funcionalidad RPC es una característica importante en un ambiente cliente/servidor en donde todos los usuarios involucrados con redes desean un procesador dedicado, una base de datos compartida y comunicaciones transparentes.

Una configuración de cómputo distribuido proporciona a un conjunto de sistemas ensamblados la imagen de ser una sola entidad. La relación de RPC con aplicaciones y redes se muestra en la siguiente figura:



Funciones como impresión, correo electrónico, administración de bases de datos y de la red, así como programas de usuario, utilizan los servicios proporcionados en la capa de ambiente de cómputo distribuido, que incluye la funciones RPC.

Las aplicaciones se mantienen independientes de las interfaces en el nivel de transporte. Solamente el ambiente distribuido de cómputo necesita tener conocimiento de los detalles específicos de este nivel. El protocolo de transporte utilizado puede ser TCP/IP, OSI u otro. La aplicación del usuario simplemente inicia el mecanismo RPC, el resto es transparente.

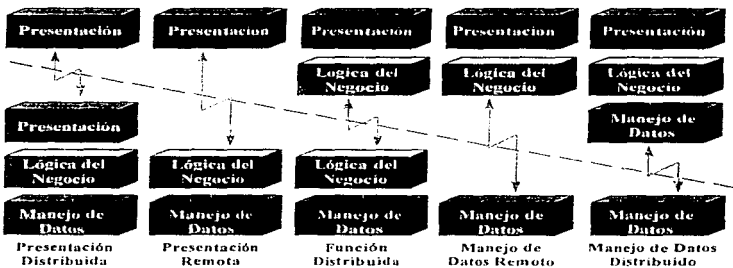
Los RPCs son apropiados para una configuración cliente/servidor. En este esquema los clientes solicitan servicios a los servidores que se encuentran en la red. Por ejemplo, un servidor de bases de datos puede proporcionar datos a varios usuarios y aplicaciones procesando los RPCs iniciados por los clientes.

Los RPCs proporcionan dos componentes principales que ayudan en la construcción de aplicaciones cliente/servidor. Existe una combinación de lenguaje/compilador que procesa los parámetros del RPC y oculta la complejidad asociada con el acceso remoto. Un componente de runtime implementa los mecanismos de llamada representados por un RPC. Esto ofrece una transparencia desde el punto de vista aplicativo con respecto a los protocolos de transporte y arquitecturas.

Actualmente existen diferentes tipos de esquemas RPC. Solaris, de SunSoft, soporta un mecanismo RPC independiente del método de transporte. Adicionalmente, Solaris puede incorporar distintos mecanismos RPC como el de DCE o el de OSI, los cuales coexisten sin problema. Lo anterior permite contar con soporte a servicios distribuidos de múltiples proveedores contruidos a partir de estas tecnologías. También proporciona a los desarrolladores mayor flexibilidad en el diseño e implementación de aplicaciones basadas en RPC que puedan interoperar en un ambiente distribuido heterogéneo.

4.13 Arquitecturas

Existen cinco configuraciones básicas en un ambiente distribuido:



Modelos Cliente/Servidor

- **Presentación Distribuida.** El procesamiento necesario para la presentación de la información se realiza en el servidor y en el cliente.
- **Presentación Remota.** Las funciones de despliegue de información se realizan totalmente en el cliente.
- **Función Distribuida.** La lógica de negocio es procesada en el cliente y en el servidor. Se tienen procesos cooperativos.
- **Manejo de Datos Remoto.** La aplicación se ejecuta totalmente en el cliente. El servidor sólo proporciona información.
- **Manejo de Datos Distribuido.** El cliente también almacena parte de los datos. A nivel lógico se ve como una sola base de datos.

Tendencias

El avance tecnológico en hardware aumenta rápidamente. Cada dos años o menos aparecen nuevos diseños y mejoras en performance. Los desarrolladores tienen el reto de crear las herramientas necesarias para aprovechar esta potencia y ponerla a disposición de los creadores de aplicaciones.

Se hace mucho énfasis en los ambientes de desarrollo para la construcción de soluciones. Los ambientes portables son también una prioridad aunque las herramientas de cómputo bajo un esquema cooperativo son indispensables para configuraciones distribuidas y heterogéneas.

Los desarrolladores necesitan herramientas flexibles de software que les permitan construir aplicaciones cliente/servidor de una manera sencilla. A pesar de que existen varios productos en el mercado es necesario contar con más opciones.

Una de las tendencias que más promete es la incorporación de tecnología orientada a objetos. La programación OO puede ser la solución para el dilema impuesto por una base tecnológica de software volátil. Puede ofrecer un esquema de bloques y componentes reusables, lo cual es esencial para un diseño efectivo de nuevos esquemas de procesamiento.

Conceptos de Cartografía y Sistemas de Información Geográfica

**Si deseas avanzar al infinito,
explora lo finito en todas
direcciones.**

Johann W. Goethe

5.1 Introducción

Cartografía es el arte, ciencia y tecnología de hacer mapas. Esta es una de las definiciones que da la Sociedad Británica de Cartografía. Los inicios de la Cartografía datan, como muchas otras cosas, a la antigua Grecia.

Actualmente, con la existencia de equipo para procesar datos geográficos, la eficiencia y utilidad que proporcionan los sistemas con capacidad de mostrar al usuario información utilizando objetos gráficos es algo comprobado. El viejo dicho de "una imagen vale más que mil palabras" aplica perfectamente a este tipo de sistemas.

Lo que empezó hace algunos años como interfaces gráficas para el usuario o GUI, se extiende aún más al contar con herramientas que permiten visualizar e interactuar con información de espacio muchas veces contenida en los sistemas tradicionales.

Para concebir e implementar sistemas de información geográfica no es necesario ser cartógrafo, sin embargo es útil conocer los principios básicos asociados a los sistemas de información geográfica o GIS¹ (Geographic Information Systems), lo cual es el objetivo de este capítulo.

¹ En todos los capítulos se utilizarán indistintamente cualquiera de los dos términos.

5.2 Definiciones

La cartografía es una disciplina que interactúa con muchas otras, como es el caso de la ingeniería. Sin embargo está muy relacionada con dos ciencias en particular, la geodesia y la geografía.

Geodesia

La Geodesia es una rama de las matemáticas aplicadas que se encarga de determinar la localización y extensión de los objetos visibles en la superficie terrestre.

Su objetivo es el procesar información geométrica y medir cantidades como distancia, dirección y altura. Esto implica, por ejemplo, determinar exactamente la forma de la Tierra y la teoría de las proyecciones de mapas.

Geografía

La geografía es la ciencia que describe y explica los objetos con un aspecto de espacio en la superficie terrestre. Estos aspectos incluyen clima, topografía, vegetación y demografía.

La geodesia proporciona los fundamentos para la cartografía y está muy relacionada con la producción de mapas topográficos. La geografía desempeña un papel determinante en la producción de mapas temáticos y utiliza los resultados de la cartografía (mapas) como medios de investigación.

Tradicionalmente los mapas se han producido utilizando diversas técnicas pero siempre en base a elementos físicos, principalmente papel. Con el procesamiento electrónico de datos, la información contenida en forma analógica en un mapa (puntos y nombres) también puede ser también almacenada en forma digital (coordenadas numéricas y conjuntos de caracteres). Esto proporciona la capacidad de procesar la información y producir mapas utilizando diversos dispositivos, como impresoras de alta resolución o plotters.

Cartografía por computadora

El campo que se desarrolló a partir de la evolución de la tecnología de la información (IT o Information Technology) en beneficio de la cartografía se denomina cartografía por computadora. Algunos sinónimos comunes son cartografía asistida por computadora y creación digital de mapas.

La cartografía por computadora es un ejemplo del primer paso que se realiza a partir de la introducción de la tecnología de la información: la automatización de alguna parte de los procedimientos existentes. El segundo paso, que responde a la pregunta de "que nuevas cosas se pueden hacer", ha dado lugar al procesamiento de imágenes y a los sistemas de información geográfica.

Estos dos campos también tienen que ver en el proceso de datos de espacio, pero con sus diferencias. El tema central de la cartografía por computadora es la presentación de la información en mapas mediante una pantalla. El procesamiento de imágenes es la manipulación de imágenes tipo raster obtenidas a través de diversos sensores o dispositivos (un satélite, por ejemplo) con el objeto de tener disponibles todos los elementos de información de una imagen. Existen algoritmos para mejorar el contraste, filtrar el ruido, etc.

Sistemas de Información Geográfica

Un sistema de información geográfica es un sistema que incluye captura, manejo y despliegue gráfico de datos con un contenido de espacio. Permite la interacción gráfica y el desarrollo de aplicaciones complejas. Puede ser utilizado para la producción de mapas (por lo tanto incluyendo el concepto de cartografía por computadora).

Integración

Idealmente un GIS también proporciona acceso a otros datos disponibles en la organización, como bases de datos comerciales, registros administrativos o bien datos de clientes, de tal forma que permite la construcción de sistemas que se pueden integrar con otras aplicaciones, como el manejo de inventarios, administración de clientes, sistemas de recuperación de texto, entre otros. A partir de este momento se denominará GIS integrado a un GIS con estas capacidades. Es evidente que un GIS integrado permite la creación de sistemas de información muy poderosos. Como ejemplo se pueden mencionar sistemas de

administración y logística de distribución, sistemas de información para ventas o bien sistemas de administración de servicios como redes de agua o electricidad.

5.3 Mapas

Mapa

Una **mapa** es un modelo generalizado, plano y a escala de información de espacio. Esta información es cualquier tipo de dato que involucre un aspecto de localización o extensión. Un plano de una construcción entraría dentro de esta definición. No existe ninguna referencia a una hoja de papel o algo similar en la definición, solamente se hace referencia a un modelo.

Modelo

Un modelo es un sistema de abstracciones acerca de algún aspecto de la realidad. Típicamente se dejan fuera los detalles con el objeto de simplificar las cosas al crear un modelo. Por esta razón un mapa no es idéntico al terreno, solo lo representa. Dicha representación podría ser un dibujo en una hoja de papel o bien datos en un sistema de información. Inclusive podría ser el mapa mental que utilizamos para dirigirnos al trabajo en las mañanas. Los mapas digitales almacenados en computadoras entran en esta definición.

Existen distintos tipos de mapas. Comúnmente los mapas se dividen en dos clases, mapas topográficos y mapas temáticos.

Mapas Topográficos

Existen muchas definiciones para este tipo de mapas. Un mapa topográfico es aquel que representa objetos visibles en la superficie terrestre con su localización y extensión correcta dentro de los límites de su escala. Por ejemplo, un mapa topográfico mostrará relieve (que es una variación de la elevación), ríos, calles, vías de tren, casas, etc. Por esta razón se puede decir que un mapa topográfico es un modelo más o menos detallado de la superficie de la Tierra.

Los mapas topográficos por lo tanto son mapas base y de propósito general. En muchos países son producidos por dependencias nacionales o bien su producción está controlada por instituciones públicas y sujetos a estándares rigurosos.

Frecuentemente contienen algunos aspectos temáticos como divisiones políticas, por lo cual algunos autores se refieren a los mapas topográficos como un caso especial de mapas temáticos.

Mapas temáticos

Los mapas temáticos representan información con un aspecto de espacio independientemente del tipo de información que se trate. No modelan la superficie de la Tierra, pero representan la información temática de tal forma que su relación con la superficie terrestre es reflejada correctamente. Por ejemplo, un mapa temático podría mostrar a la República Mexicana con sus estados coloreados de acuerdo a los resultados de las elecciones. Otro mapa de este tipo podría representar la carga promedio de tránsito en las calles de una ciudad utilizando flechas de distintos tamaños.

La producción de mapas temáticos es un campo muy amplio y ha ganado mucho interés debido a las enormes posibilidades que proporciona el proceso de datos.

Mapas catastrales

Los mapas catastrales son un tipo especial de mapas que en esencia son mapas o planos con una escala muy grande utilizados en el catastro. El catastro es llevado por autoridades públicas y contiene información acerca de esquinas y límites de manzanas o pedazos de tierra. El catastro tiene una larga y diferente historia dependiendo del país que se trate, por lo cual los métodos y estándares utilizados difieren bastante.

A pesar de que los mapas de catastro tienen una relación estrecha con los mapas topográficos, no existe en la mayoría de las ocasiones una organización común que permita el manejo de ambos.

El catastro proporciona información importante para la administración pública así como para empresas y organismos privados. Por ejemplo, el catastro es utilizado típicamente para establecer los derechos legales de una propiedad (catastro legal) así como para cobro de impuestos (catastro fiscal). Los catastros de propósito múltiple pueden contener información acerca de estadísticas referentes a la población o vehículos, ambientales, etc. El catastro es buen candidato para la automatización mediante un GIS.

5.4 Escala y Generalización

Escala

La escala de un mapa proporciona un factor por el cual deben ser multiplicadas las distancias en un mapa para obtener la distancia real en la superficie terrestre. Se expresa como una división o proporción "1:factor". Por ejemplo, una escala de 1:20,000 significa que un centímetro del mapa corresponde a 20,000 centímetros, o 200 metros.

Escala grande y pequeña

Los mapas de escala grande son los mapas que presentan una escala con un factor pequeño. Por otro lado, los mapas de escala pequeña son aquellos que presentan un factor grande. Por definición, un mapa de escala grande es todo aquél con una escala de 1:75,000 o mayor. Los mapas de escala pequeña son aquellos con una escala de 1:600,000 o menor.

Limites de la escala

Debido a que la superficie de la Tierra está curvada, existen problemas al tratar de representarla en una superficie plana (como un mapa). Debido a los límites de las proyecciones en mapas, no es posible crear un mapa en el que todas las distancias dentro de éste puedan ser convertidas a distancias reales utilizando la escala, ya que ésta es una escala plana para un objeto curvado. En el mejor de los casos, esto es posible para un subconjunto, por ejemplo, todas las distancias medidas a partir de un punto central. Por lo anterior, en todos los mapas y de distinta manera, la escala es dependiente de la localización y dirección de las medidas. Muchos mapas inclusive despliegan líneas rectas sobre la superficie de la Tierra como curvas.

En mapas de escala grande, los límites de escala pueden ignorarse. Lo anterior se debe a que la superficie de la Tierra puede ser aproximada en pequeñas regiones por un plano de tal forma que las distancias representadas en el mapa son muy aproximadas a las reales.

Escala en un GIS

En un GIS los datos geográficos típicamente se almacenan como coordenadas, por lo cual no existe el concepto de escala. La escala aplica solamente para los mapas desplegados en la pantalla o dibujados por algún dispositivo en especial.

Ejemplos de escalas

Algunas escalas típicas en mapas empiezan a partir de 1:10,000 para planos y diferentes tipos de catastro; aquellas de 1:25,000 a 1:100,000 son apropiadas para mapas topográficos. Una escala de 1:1,000,000 está planeada para el proyecto del Mapa Internacional del Mundo en la Escala del Millón, conducido por la ONU.

Generalización

Un mapa ha sido definido como cierto tipo de modelo de información de espacio; y a un modelo como un sistema de abstracciones. El abstraer casi siempre involucra dejar fuera ciertos detalles, simplificar, etc. Esto también aplica a los mapas. En cartografía la introducción de la abstracción se denomina generalización.

Razones para la generalización

Conforme la escala de un mapa es menor, es necesaria mayor generalización debido a que más cosas tendrían que ser desplegadas dentro del mismo espacio del mapa. Adicionalmente, dado que el mayor límite en precisión de dibujo es alrededor de 0.1 mm, nada más pequeño puede ser desplegado. Otra razón para incluir generalización es la de no sobrecargar a los mapas y presentar la información en una forma clara.

Combinación y simplificación

Por ejemplo, una casa de 20 metros de largo sería representada en un plano con escala 1:10,000 (no requiere mucha generalización) con una longitud de 2mm. En un mapa con escala 1:100,000 algunas casas se dejarían fuera o se combinarían, con lo cual se tendría una idea más general de la distribución. De manera similar, un camino de 10 metros de ancho se vería como una línea de 0.1mm. El camino sería al menos

simplificado haciéndolo más suave. Lo que sucede con las casas se llama combinación y lo que sucede en el caso del camino se denomina simplificación.

Clasificación y Selección

Inclusive en escalas pequeñas una ciudad podría representarse como un símbolo, el cual dependería del número de habitantes. Lo anterior se llama clasificación. Las villas y ciudades pequeñas podrían dejarse fuera del plano. Esto se llama selección.

Exageración y Desplazamientos

En un mapa de caminos con escala 1:1,000,000 un camino de 30 metros de ancho sería una línea de 0.03 mm de ancho. Como tal, un mapa así no sería útil ya que los caminos se dibujarían como dos líneas con una distancia de 2mm entre ellas. Esto se denomina exageración. Si un camino debe pasar por un valle estrecho junto con un río y vías de tren, las líneas correspondientes deben ser movidas para permitir el dibujo. Esto se llama desplazamiento.

Todos estos procesos son aplicados al mismo tiempo en todas las escalas y no en secuencia como podría parecer. Por ejemplo, el camino de 10 metros de ancho probablemente no sería desplegado con un ancho de 0.1 mm en un mapa con escala 1:100,000 sino que sería exagerado.

Ejemplo para Precisión

Como ejemplo del orden de magnitud de los efectos por generalización, en un mapa topográfico con escala 1:100,000 no se debe esperar una precisión de localización de más de 1 mm, o 100m reales.

Generalización en un GIS

En un GIS la información geográfica se tiene en forma digital, por lo cual en teoría no existe impedimento para almacenar todo con una precisión real de 1 cm.

Generalización por los datos de entrada

La generalización también está presente en un GIS. Por ejemplo, si los datos geográficos se digitalizan a partir de un mapa de papel, la digitalización no puede ser más precisa que el mapa. En el ejemplo del mapa con escala 1:100,000 la precisión de los datos obtenidos no sería mayor a 100 metros, inclusive si es almacenada con una precisión de 1 cm. Toda la generalización presente en el mapa es automáticamente introducida dentro de los datos almacenados en el GIS.

Generalización Automática

Este todavía es un tema en investigación. La generalización automática es útil para dibujar mapas de pequeña escala a partir de datos con poca generalización, por ejemplo datos digitalizados a partir de mapas de escala grande. Esto es interesante para la cartografía por computadora debido a que la generalización automática evitaría tener que manejar datos para diferentes escalas.

Aspectos Prácticos

En la práctica, especialmente al construir una aplicación, estos problemas no son muy importantes debido a que la mayoría de las ocasiones la pantalla de despliegue y el dibujo del mapa estarán dentro de un rango de escala pequeño. Esto se debe a que la recopilación de datos resulta costosa al igual que el procesamiento y almacenamiento de detalles innecesarios. Por lo anterior es recomendable tener tanta generalización como sea posible dentro de los límites requeridos por la aplicación, dejando fuera detalles irrelevantes.

Por ejemplo, si los datos geográficos se obtienen digitalizando un mapa, se escogerían mapas con la menor escala aceptable. Conforme el área aumenta con el cuadrado del largo, el escoger mapas con una escala de 1:25,000, en vez de 1:100,000, significaría 16 veces más datos a digitalizar y 16 veces más costo. O bien, podría incluir generalizaciones arbitrarias hechas por la persona que está digitalizando.

5.5 Contenido del mapa

A continuación no se hará referencia al contenido de mapas temáticos, sino al contenido de mapas topográficos. Todo posible contenido podría ser delineado como:

Simbolización

Los símbolos son lo que se dibuja en el mapa aparte del relieve (variación de la elevación) y textos. Existen símbolos formados por puntos, líneas y áreas. No existe un estándar internacional para los símbolos en mapas aunque se utilizan símbolos similares para representar el mismo objeto. Es buena práctica el adherirse a los símbolos generalmente utilizados, especialmente los de mapas topográficos. El significado de los símbolos se explica con una leyenda a lado del mapa o en un papel adicional.

Simbolos de puntos

Los símbolos de puntos son usados para marcar la localización de edificios, pueblos, ciudades y puntos específicos. El uso de símbolos de puntos, en lugar de dibujos exactos, ya ha sido sugerido en el tema de generalización. Símbolos comunes son utilizados para evitar textos adicionales en el mapa.

Simbolos de líneas

Estos símbolos difieren en estilos y colores. Son utilizados para marcar ríos, calles, vías de tren, tuberías así como límites políticos y administrativos.

Simbolos de áreas

Estos símbolos marcan diferentes tipos de vegetación o cosechas. Las áreas son diferenciadas por colores o por pequeños símbolos geométricos.

Relieve

El relieve es la variación de la elevación en base al nivel del suelo. Existen distintas maneras de representar relieve en mapas, siendo las líneas de contorno la forma geoméricamente adecuada, por lo que son utilizadas para prácticamente todos los mapas topográficos.

Líneas de contorno

Una línea de contorno es una línea que pasa a través de varios puntos con la misma elevación.

El dibujar líneas de contorno para diferentes elevaciones, por ejemplo para todas las elevaciones que pueden ser divididas entre diez, proporciona una representación adecuada del relieve. Algunas líneas de contorno pueden ser dibujadas utilizando líneas más anchas.

El uso de líneas de contorno solo está limitado para algunos tipos de relieve, como regiones montañosas y elevaciones bruscas. Las líneas de contorno estarían muy cerca debido a la inclinación, por lo que no serían útiles. En estos casos son reemplazadas o complementadas con otros métodos como el sombreado.

Texto

Los nombres y números identifican lo que está dibujado en el mapa. La elección cuidadosa del tipo de letra, tamaño y color es necesaria para crear mapas entendibles. Diferentes tipos de letra y tamaños pueden ser utilizados dependiendo de la importancia y tipo de objeto representado. Los números son utilizados para elevaciones y algunas líneas de contorno.

No es fácil encontrar el nombre correcto para algunos lugares. Los llamados nombres estandarizados son definidos por el Grupo de Expertos en Nombres Geográficos de las Naciones Unidas. Los nombres usados en el lenguaje de determinado país para nombrar a lugares en el extranjero son conocidos como nombres convencionales.

Colores

Los colores son utilizados para diferenciar diferentes tipos de símbolos de línea. Por ejemplo, los ríos podrían ser líneas de color azul, los límites rojos, etc. Los nombres y símbolos también se representan con diferentes colores.

Otro posible uso de los colores puede ser para diferenciar entre diferentes elevaciones. Existen diversos sistemas de coloreado para este propósito.

El uso del color es uno de los aspectos menos estudiados en el diseño de mapas y muchas veces sujeto a críticas. El color debe buscar un significado o hacer que el mapa sea más legible, lo cual involucra un fuerte aspecto estético.

El uso adecuado del color en una aplicación GIS puede requerir una concepción distinta a la utilizada para las interfaces de usuario, sin embargo mayor número de colores no necesariamente significa algo mejor o con mejor presentación.

5.6 Proyecciones

Prácticamente todos los mapas son planos. La Tierra, como fue descubierto por los antiguos griegos, no lo es. Este hecho presenta un problema en el sentido de como representar una superficie curva en el espacio en un plano de dos dimensiones.

El Globo

Existe una forma de representar la superficie terrestre sin distorsiones: el globo. Desafortunadamente los globos están limitados a escalas muy pequeñas. Un globo con una escala 1:100,000 (como un mapa topográfico típico) tendría un diámetro de 130 metros aproximadamente.

Debido a que todos los mapas contienen distorsiones que solo pueden ser pasadas por alto cuando se trata de escalas grandes, se han desarrollado proyecciones para distintos propósitos, algunas de las cuales se tratarán a continuación.

Sistemas de coordenadas

Sistema de coordenadas

Al usar un sistema de coordenadas se puede asignar un conjunto de números a los puntos en el espacio o en superficies que corresponda a la localización de éstos. El sistema de coordenadas describe como se lleva a cabo la asignación de estos conjuntos de números llamados coordenadas.

Dimensión

Un conjunto de dos números es suficiente para identificar en forma única a un punto en un plano o superficie, mientras que se requieren tres números para puntos en el espacio, por lo cual se hace referencia a dos o tres dimensiones.

Elección del sistema de coordenadas

Con un sistema de coordenadas, las relaciones y características geométricas pueden ser expresadas en ecuaciones entre las coordenadas. Se han creado distintos tipos de sistemas de coordenadas con el fin de simplificar las ecuaciones por lo que la elección del sistema de coordenadas depende del problema que se tenga.

Coordenadas Rectangulares

El sistema de coordenadas más común y más usado es el de coordenadas rectangulares, también conocido como coordenadas cartesianas. Un sistema de este tipo en un plano consistiría simplemente en dos líneas que se intersectan en ángulos rectos. Una de las líneas puede ser imaginada como orientada horizontalmente y se denomina eje X o abscisa. La otra línea, orientada verticalmente, se denomina eje Y u ordenada. El punto de intersección de los ejes se llama origen y tiene las coordenadas cero para ambos ejes.

A cada punto en el plano se le pueden asignar sus coordenadas midiendo su distancia del eje Y para obtener su coordenada X; y desde el eje X para calcular su coordenada Y. Las coordenadas X a la derecha del origen son positivas; las coordenadas Y arriba del origen también son positivas. Las distancias prácticamente siempre son medidas en las mismas unidades para ambas coordenadas.

Coordenadas rectangulares en el espacio

Un sistema rectangular de coordenadas en el espacio tiene un tercer eje, llamado eje Z, que intersecta los ejes X y Y en el origen con ángulos rectos. Las coordenadas no se definen como distancias de los ejes, sino como distancias desde los planos que se forman con los otros dos ejes.

Coordenadas Esféricas

Un sistema de coordenadas tridimensional que es de los más importantes en cartografía es el de coordenadas esféricas. La localización de un punto en el espacio puede ser definida en forma única dando su distancia al origen, llamado radio, y dos ángulos. Los ángulos utilizados en cartografía son definidos en forma distinta a los utilizados en otras ciencias y en matemáticas.

Latitud y Longitud

La latitud de un punto se define como el ángulo entre el radio y un plano fijo y una línea fija en dicho plano. Los valores positivos de latitud son aquellos puntos arriba del plano.

La longitud de un punto se define como el ángulo entre la proyección del radio en el plano fijo y alguna línea fija en este plano. Vista desde arriba, la longitud se mide en sentido contrario a las manecillas del reloj.

Independencia de Coordenadas

Esta opción pudiera verse innecesariamente complicada, pero las coordenadas esféricas se adaptan exactamente al problema planteado: localizar puntos sobre la superficie terrestre. Escogiendo un radio constante nos deja con una esfera y un par de ángulos que representan un sistema de coordenadas para la superficie de la esfera.

En muchos casos, la tierra puede ser aproximada a una esfera, por lo tanto las coordenadas esféricas proporcionan un sistema global de coordenadas para la Tierra. (la superficie de la tierra es bidimensional, inclusive si tiene curvatura en el espacio.

Esta opción es la más simple debido a que proporciona dos coordenadas que pueden variar en forma independiente. El utilizar coordenadas rectangulares en el espacio para localizar puntos en la superficie terrestre nos dejaría con tres coordenadas mutuamente dependientes. Este método es ineficiente.

La latitud y longitud se miden en grados, en donde un ángulo recto tiene 90 grados. Las latitudes máximas y mínimas son de 90 y -90 grados en los polos. La longitud varía entre 180 y -180 grados.

Geometría en una esfera: Líneas Rectas

Es posible definir geometría en la superficie de una esfera de manera similar a la geometría en un plano.

En un plano la distancia más corta entre dos puntos es una línea recta. En una esfera la conexión más corta entre dos puntos es un gran círculo a través de dichos puntos. Un gran círculo es un círculo con el máximo radio

posible. Todos estos círculos pueden definirse como la intersección de la esfera con algún plano a través del centro de la esfera.

Ecuador, Meridianos y Paralelos

La intersección de la esfera con el plano fijo utilizado para definir las coordenadas es un gran círculo llamado ecuador. Todas las líneas de longitud constante también son grandes círculos, llamados meridianos. Las líneas de latitud constante no son, con excepción del ecuador, grandes círculos. Estas líneas se denominan paralelos debido a que pueden imaginarse como la intersección de la esfera con algún plano que es paralelo al plano del ecuador.

Cuando se utilizan coordenadas esféricas el plano del ecuador se escoge perpendicular al eje de rotación de la Tierra. Las longitudes son medidas con referencia al meridiano cero de Greenwich.

Geometría en una esfera. Ángulos

Tres puntos en una esfera definen un triángulo. Sus lados se obtienen como grandes círculos entre los puntos. Es posible definir los ángulos en las esquinas del triángulo como los ángulos entre los planos a través de los diferentes grandes círculos, pero la suma de los ángulos de un triángulo esférico se siempre mayor que dos ángulos rectos).

Límites de las proyecciones

Debido a que es posible definir distancias, líneas rectas (como grandes círculos) y ángulos entre ellas, también es posible construir una geometría completa en la esfera. Las reglas son un poco diferentes a las de la geometría plana, pero en la vida diaria no nos damos cuenta de la diferencia debido a que la esfera en que vivimos es muy grande.

Las proyecciones de mapas describen como mapear puntos de la superficie de una esfera, descritos en coordenadas esféricas, en un plano. Dado que las geometrías de planos y esferas obedecen a diferentes leyes es imposible pensar en una proyección que preserve todas las características geométricas.

El Geoide

En la parte de sistemas de coordenadas se estableció que la Tierra puede ser aproximada a una esfera por varias razones. Para algunos casos, como la alta precisión requerida en los mapas topográficos, esto no es suficiente.

Forma de la Tierra

Las distancias desde el centro de la Tierra a un polo y al ecuador difieren en un 0.3%. Esto puede verse poco, pero dado que el radio de la Tierra mide alrededor de 6,400 km, la diferencia viene a ser de unos 20 km. Si se desean medir elevaciones en la superficie terrestre 20 km es una gran diferencia. La montaña más alta no llega a los 9 km de altura.

Geoide

La superficie de la Tierra no puede ser descrita con una fórmula matemática, inclusive si no se tomaran en cuenta objetos como montañas. Un geoide, la forma de la Tierra, se define como una superficie con potencial de gravitación constante. Cualquier superficie de agua es una superficie con potencial de gravitación constante. Sin embargo, debido a que la Tierra no está completamente cubierta de agua, el determinar la forma del geoide es parte de la tarea de los geodésicos.

Elipsoides Estándares

Por razones prácticas, los elipsoides estándares han sido definidos como aproximaciones de geoides. Un elipsoide puede ser descrito con una fórmula matemática simple. Puede imaginarse como una esfera que ha sido uniformemente deformada en la dirección de un eje. (En general, deformada en la dirección de dos ejes en diferentes grados). Una esfera es un caso especial de un elipsoide.

El primer elipsoide fue definido por W. Besse en 1841. Los elipsoides estándar usados actualmente incluyen al de J. Hayford (1924), F.N. Krassovski (1940) y de la Unión Internacional de Geodesia y Geofísica (1967).

Estos elipsoides proporcionan simultáneamente cálculos básicos y una muy buena aproximación al geode. Por ejemplo, el elipsoide de Hayford no difiere más de 50 metros que el geode.

Elevación

La elevación puede ser definida como una distancia vertical desde la superficie de referencia dada por el elipsoide. Vertical significa paralela a la dirección de gravedad, y por lo tanto perpendicular a la superficie del elipsoide. Naturalmente, es necesario definir alguna elevación de referencia como cero. Estos datos se definen de forma diversa en distintos países, como el nivel del mar promedio. Promedio debido a las mareas y otros factores, ya que los océanos se mueven constantemente.

Por ejemplo, el NGVD (National Geodetic Vertical Datum) usado en Estados Unidos de Norteamérica está basado en 26 referencias de este país y Canadá. En Alemania, el NN (Normalnullpunkt) está basado en el nivel del mar promedio de Amsterdam. En Francia, se utiliza el nivel del mar promedio de Marsella. En los países del Este de Europa se utiliza el de Kronstadt. En Inglaterra el de Liverpool, etc.

Para este dato, se puede construir una superficie de referencia en base a un elipsoide estándar. Esto permite definir y, con los métodos apropiados, medir elevaciones inclusive lejos de la costa.

Proyecciones

Como se mencionó anteriormente, es imposible mapear la superficie terrestre en un plano sin perder características geométricas. A pesar de que no existe proyección perfecta se han creado diferentes tipos de proyecciones para diversos propósitos. En Cartografía, por ejemplo, la elección de la proyección adecuada es crucial.

Proyección

Una proyección, en el ámbito de la cartografía, es una función matemática que eventualmente asigna a un punto en la esfera un punto en el plano. Si la función puede ser construida geométricamente, se habla de una proyección real. La mayoría de las proyecciones pueden ser calculadas y ser referidas como mapeos.

5.7 Levantamiento de Datos

Para producir mapas, es necesario recopilar datos gráficos y datos no gráficos y, al menos en una ocasión, se debe hacer a partir de una fuente primaria. El levantamiento de datos no gráficos (atributos) requiere el dirigirse físicamente a la localidad y preguntar directamente por los nombres y otros datos de este tipo, lo cual no es una tarea trivial como pudiera parecer. La recopilación de datos gráficos, o datos de locación, se realiza por mediciones de terreno y por medición remota.

Triangulación

La triangulación es la definición de puntos de referencia, llamados estaciones, en la superficie terrestre mediante la construcción de una red de triángulos.

Los triángulos se construyen midiendo líneas base. Comúnmente se utilizan alambres de 24 metros de longitud para este propósito. Los alambres son de un material que varía muy poco en longitud a diferentes temperaturas. Los ángulos se determinan utilizando teodolitos u otros instrumentos de precisión para medir ángulos. Es necesario una teoría completa de corrección de errores para mejorar los resultados.

También existen métodos modernos para medir distancias con dispositivos electrónicos que permiten construir triángulos al medir directamente la longitud de sus lados. Esto se denomina trilateración.

Los puntos de referencia obtenidos por triangulación son marcados o alzados en el campo con pilares de piedra y construcciones similares dependiendo de su importancia. También son indicados en los mapas topográficos, típicamente con un símbolo en forma de triángulo.

Los puntos de referencia forman redes de triangulaciones nacionales de diferente orden. Cada orden es la base para la triangulación del siguiente orden. Puede haber hasta cuatro ordenes, pero esto depende totalmente del país.

La longitud típica para los lados de triángulos en una red de primer orden es de 20 a 50 km. Para los de cuarto orden es de 1 a 3 km.

Elevación

Existen varios métodos para obtener la elevación de un punto, los cuales difieren en precisión y costo. Si se cuenta con suficientes puntos con una elevación conocida, es posible obtener líneas de contorno en base a la interpolación entre dichos puntos.

Medición Barométrica

Como la presión del aire depende de la elevación, ésta puede ser obtenida con un aneróide (barómetro de precisión). Esta es el método más rápido, pero también el menos preciso.

Medición Trigonométrica

La elevación se obtiene a partir de una distancia horizontal desde un punto de elevación y ángulo conocidos. Los resultados pueden corregirse para ajustarse a la curvatura de la Tierra y la refracción de luz de la atmósfera.

Nivelación

Es el método más preciso y caro. Los puntos de comienzo y fin son siempre los mismos, o bien son puntos de elevación conocida. Los resultados se corrigen mediante cálculo de errores.

Por ejemplo, un famoso cálculo de este tipo se realizó desde Bolzano. Cuando la nivelación terminó de nuevo en Bolzano, una diferencia de elevación de 1,234 metros había sido nivelada con un error final de 7mm.

Medición Remota

La medición remota es la recopilación de información sobre un objeto sin que se tenga contacto físico entre el observador y el objeto.

La medición remota permite un levantamiento de datos rápido y confiable, especialmente en terreno difícil. Se requieren algunos puntos de referencia que sean visibles y de coordenadas conocidas para la corrección de distorsiones y determinación de coordenadas.

Los dos métodos empleados actualmente son la fotogrametría aérea, realizada desde avión, y la medición remota por satélite. La fotogrametría terrestre también se viene practicando desde 1850.

Fotogrametría Aérea

La fotogrametría aérea ha sido utilizada para levantar datos desde 1915. Un avión sobrevuela a una altura constante la región a ser medida utilizando un barómetro de precisión entre uno o dos metros. Se toman una serie de fotos estereoscópicas del terreno. Con un comparador estéreo es posible corregir las distorsiones y obtener las coordenadas identificando los puntos de referencia (de coordenadas conocidas) en las fotos.

Las líneas de contorno pueden ser directamente medidas en las fotos estereoscópicas. Sin embargo, debido a que la mayor precisión que se obtiene con la fotogrametría aérea es de 30 cm, es posible obtener valores más precisos utilizando otros métodos.

Medición Remota por Satélite

Las imágenes obtenidas por satélite todavía no alcanzan la precisión obtenida con la fotogrametría aérea. Sin embargo este método permite levantar datos de toda la Tierra en solo algunos días, lo cual proporciona la posibilidad de realizar cambios necesarios muy rápidamente.

Los satélites proporcionan imágenes tipo raster que son enviadas a la Tierra, las cuales pueden ser corregidas y asociadas a sistemas de coordenadas utilizando puntos de referencia de manera similar a la fotogrametría aérea. Las imágenes son tomadas en distintas bandas del espectro electromagnético, como la luz visible y distintas bandas del infrarrojo. Se han desarrollado técnicas para el procesamiento de imágenes como el retoque, correcciones geométricas, filtrado, supresión de ruido y transformaciones, entre otras, que permiten hacer visible toda la información contenida en las imágenes.

Mientras que los satélites LANDSAT envían imágenes con una resolución hasta de 30 metros, los satélites SPOT proporcionan una resolución de alrededor de 10 metros con la capacidad de imágenes estereoscópicas. Esto es suficiente para producir mapas topográficos con una escala de

1:100,000. Sin embargo, la mayoría de las imágenes obtenidas por satélite se utilizan para mapas temáticos.

Fotogrametría por Satélite

Cuando se pueden traer fotografías a la Tierra después de una misión espacial, como fue el caso del Skylab y como sucede en las misiones de los transbordadores espaciales, estas fotografías pueden ser utilizadas en forma similar a las fotos aéreas.

Sistemas de Posicionamiento por Satélite

Con la disponibilidad de satélites con parámetros conocidos de órbita, las señales de los satélites pueden ser usadas para determinar posiciones en la superficie terrestre. Existen varias configuraciones de satélites disponibles que cubren varias regiones de la Tierra con diferente precisión. Los sistemas de Posicionamiento son ampliamente utilizados para navegación y actualmente también permiten el ajuste preciso entre puntos sin requerir la visibilidad entre éstos.

El Sistema Global de Posicionamiento (GPS o Global Positioning System), uno de estos sistemas, permite un ajuste del orden de centímetros entre puntos.

Resulta interesante el ver como algunas de las técnicas utilizadas en cartografía han evolucionado a través del tiempo mientras que otras se han modificado muy poco, debido a la importancia de conceptos que se remontan a la antigua Grecia.

Con el avance de la tecnología las posibilidades en lo que respecta al conocimiento físico de la Tierra se amplían, ya que como se vio en este capítulo, actualmente es posible realizar mapas de cualquier lugar del planeta en pocos días y sin necesidad de estar físicamente en el terreno para levantar los datos.

De manera similar, cada vez son mayores las posibilidades de explotar e integrar este tipo de datos con sistemas de información comunes, que es la esencia de un GIS. Las ventajas son muchas, ya que el poder consultar gráficamente información proporciona una flexibilidad muy difícil o imposible de lograr con los sistemas tradicionales.

En el siguiente capítulo se mostrarán detalles de un producto de este tipo y un ejemplo de la utilización que se le puede dar.

Caso Práctico

**El saber no basta, debes aplicarlo.
El deseo no es suficiente, debes
hacerlo.**

Johann W. Goethe

6.1 Introducción

El objetivo de este capítulo es el mostrar la manera en que se puede aplicar un manejador de bases de datos, un lenguaje de cuarta generación y la implementación de una arquitectura cliente/servidor en la creación de un sistema real. Se plantean los antecedentes y problemática de la Delegación Tlalpan en lo que respecta al control y administración del equipamiento urbano. Se describen los pasos seguidos en cuanto a la metodología empleada, descripción del hardware y software utilizado así como los resultados obtenidos después de la implementación.

Como antecedente se expone la totalidad del esquema de solución determinado por la Delegación Tlalpan, parte del cual se implementó como un prototipo creado con la finalidad de mostrar la viabilidad de la solución, que es el que se analizará como caso práctico.

6.2 Problemática

Antecedentes

Sistema Integral de Demanda Ciudadana. Delegación del D.D.F. en Tlalpan.

La Delegación Tlalpan, unidad desconcentrada del D.D.F. como entidad de atención a la ciudadanía ha experimentado un crecimiento demográfico en la extensión territorial, lo que ha generado una mayor necesidad de servicios, mismos que son demandados a diversas áreas de atención al público originando en varios casos la duplicidad de solicitudes para un mismo servicio.

Los sistemas de correspondencia y seguimiento existentes en la computadora de esta delegación tales como los utilizados en la Secretaría Particular del C. Delegado, la Subdelegación Administrativa, la Subdelegación de Obras y la Subdelegación de Agua Potable, son operados en forma independiente. Dichos sistemas tienen casi la misma estructura de datos, además de contener registros de una misma petición de servicio en varios archivos, lo cual resulta en duplicidad innecesaria de información que repercute en el mal aprovechamiento de los recursos de cómputo, especialmente los dispositivos de almacenamiento.

Lo anterior originó iniciar el compromiso de esta Delegación con el sector central para elaborar e implementar un sistema de atención a la ciudadanía, lo cual se plasmó en el documento del programa anual de simplificación administrativa correspondiente al año de 1993.

Alternativa de Solución

Debido a que el problema reside en el control de información, se plantea la elaboración de un sistema integral de recepción y seguimiento de la demanda ciudadana, el cual permita a las áreas involucradas en la atención a la misma identificar la posible duplicidad de información, recursos y esfuerzos en el momento de ser recibida, lo que requerirá concentrar los datos en un sistema central que realice la validación de éstos.

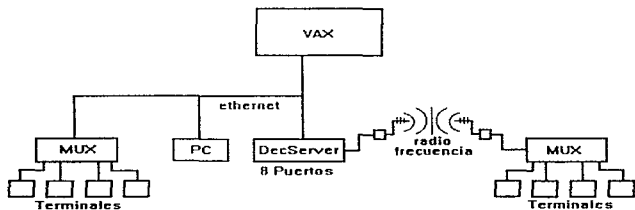
Los módulos de información integrados en el Sistema Integral de Demanda Ciudadana serán:

- a) Catálogo de Colonias y Manzanas (Zonificación de Uso de Suelo).
- b) Catálogo de Servicios por Área Operativa.
- c) Catálogo de Asociaciones y Dependencias.
- d) Catálogo de Equipamiento Urbano de la Delegación.
- e) Cartografía para presentación gráfica de la información.
- f) Recepción de la demanda ciudadana.
- g) Dictamen de servicio y generación de orden de trabajo.
- h) Seguimiento a la orden de trabajo.

De acuerdo al interés de mostrar la funcionalidad de una base de datos y un lenguaje de cuarta generación integrados con una herramienta para manejar cartografía, se seleccionaron varias consultas representativas de los módulos para implementar como prototipo.

6.3 Arquitectura

Se cuenta con un servidor VAX 4100 con 64 MB de memoria principal y sistema operativo OpenVMS V5.5.2. Para el acceso a nivel local se cuenta con equipos PC con software para emulación de terminal y terminales VT420. El esquema planteado originalmente contemplaba la utilización de terminales gráficas a nivel local y remoto para el despliegue de mapas cartográficos, aunque se descartó esta posibilidad debido al alto costo de la memoria para las Xterminales. A nivel local se cuenta con un controlador DEC SERVER 380 de 8 puertos, con el cual se pretende en un futuro tener puertos remotos a través de radio frecuencia utilizando varios DecMux 300, aunque todavía no se define como se implementará. También se contempla la adquisición de un plotter de alta resolución para la impresión de los mapas, aunque se estudiará el costo/beneficio en comparación a otras opciones. A continuación se muestra un diagrama de la configuración que se desea a futuro:



Para la creación del prototipo se utilizará una red ethernet que incluye al servidor VAX 4100 y equipos PC 486/33 DX con PATHWORKS DECNET 4.1a para servicios de red y Windows 3.1 como ambiente gráfico.

6.4 Recursos de Hardware

En base a los recursos existentes en la Delegación y al alcance del prototipo se decidió que en esta etapa se utilizaría sólo el equipo necesario para mostrar la funcionalidad de los productos y poder realizar demostraciones a los funcionarios de la Delegación.

El hardware que se contempló para esta etapa incluye el siguiente equipo:

Servidor VAX 4100
64 MB RAM
3 Discos de 470 MB

Equipo PC 486/33 Mhz DX
16 MB RAM
Disco 420 MB
Tarjeta de red Intel EtherExpress
Impresora HP Deskjet 500C

La segunda etapa de este proyecto contempla la puesta en marcha de los módulos más importantes del Sistema Integral de Demanda Ciudadana con un grupo de usuarios que se seleccionará de acuerdo a los objetivos del proyecto, para lo cual será necesario la adquisición de más equipos PC así como contar con más recursos en el servidor.

6.5 Recursos de Software

El software de sistema existente en el hardware disponible en la Delegación Tlalpan para este proyecto incluye:

VMS V5.5.2
DECNET PATHWORKS V4.1a
MS DOS 5.0
Windows 3.1
ADABAS V2.1.2
NATURAL V2.1.5

En base a las políticas de informática existentes, infraestructura de sistemas actual y tipo de funcionalidad requerida se decidió que era conveniente aprovechar la experiencia del personal de sistemas de la Delegación en el manejo de la base de datos y el lenguaje de programación, por lo cual se pensó en un producto que se pudiera integrar con este software. El producto que permite lo anterior es NATURAL GEOGRAPHIC, que además cuenta con la ventaja de ser fabricado por el mismo proveedor de la base de datos y el lenguaje de cuarta generación existentes.

En cuanto a comunicaciones se decidió utilizar el software de conectividad propio de Digital con el objeto de evitar el instalar un nuevo protocolo de comunicación.

A continuación se presentan las características del software de desarrollo utilizado:

Base de Datos: ADABAS

Es un DBMS creado por Software AG, el cual ha sido comercializado desde 1971 con bastante éxito a nivel mundial. ADABAS es reconocida como una de las bases de datos más versátiles y con mejor performance, por lo cual es utilizada en ambientes grandes, la mayoría mainframe, en sistemas de misión crítica.

Inicialmente como producto para plataforma IBM mainframe, ha sido portada a muchas otras plataformas, entre las que se encuentran UNIX, Windows NT y OpenVMS. A pesar de que la base de datos se distingue por su performance, también ha crecido en capacidades. Originalmente como un DBMS de listas invertidas, puede implementar diversos modelos de

datos, de ahí el nombre de Adaptable DATA BASE, es decir, ADABAS es una base multimodelo.

ADABAS es un RDBMS basado en listas invertidas. Puede ser accedido utilizando SQL con el producto ADABAS SQL Server, lo cual amplía las capacidades del manejador al soportar estándares tipo ODBC. ADABAS en esencia es una base de datos multimodelo, lo cual permite implementar gran diversidad de aplicaciones, incluyendo sistemas de información cartográfica.

En cuanto a performance, ADABAS se ha distinguido por su rendimiento en aplicaciones de tipo transaccional. Lo anterior se debe a la avanzada implementación de sus estrategias de indexación y el manejo eficiente de buffers. A diferencia de otras bases de datos con arquitectura de Btrees, las listas invertidas de ADABAS proporcionan un mecanismo rápido para almacenamiento y recuperación de datos.

La utilización de CPU es bastante eficiente debido a que la base de datos se diseña en forma individual para cada plataforma soportada. Utiliza tipos de datos nativos e implementa algoritmos eficientes para compresión de datos. En la mayoría de los casos ADABAS usa funciones de bajo nivel y en muchas ocasiones es capaz de evitar servicios de alto nivel del sistema operativo, lo cual hace que esta base de datos sea muy eficiente. ADABAS soporta llamadas directas, lo cual permite evitar la sobrecarga de CPU que causan los optimizadores.

La arquitectura de listas invertidas empleada por ADABAS genera menos I/O a disco que los Btrees utilizados por la mayoría de las bases de datos relacionales. Las listas invertidas son estructuras de datos muy flexibles lo cual las hace propicias para ambientes transaccionales y de consulta. Adicionalmente, las listas invertidas proporcionan un mejor mecanismo de contención de tal forma que los candados a nivel página o tabla de otros sistemas. A continuación se explican con mas detalle estas estructuras.

Listas Invertidas

Una lista invertida es un término genérico que describe un método de almacenamiento y recuperación de datos. En cualquier DBMS el almacenamiento de datos se puede realizar de distintas maneras, aunque la más común es la de tener los datos agrupados en entidades construidas en forma similar. Estas entidades son conocidas como tablas o archivos. Cada elemento se denomina renglón (o registro, en caso de que la entidad se denomine archivo). Cada parte del renglón se conoce como columna (o campo, si éste pertenece a un registro).

La técnica de acceso utilizada por cada fabricante de bases de datos es el elemento que hace a un DBMS relacional (cuando soporta el modelo relacional y SQL, por ejemplo), jerárquico (cuenta con "padres" e "hijos") o red (utiliza "navegación"), etc. Prácticamente todos los DBMS utilizan algún tipo de índices. Los índices pueden estar ocultos para el programador, o pueden ser manipulados, pero en ambos casos el DBMS tiene que utilizarlos para obtener datos, de lo contrario tendrá que hacer un barrido de todos los registros, lo cual utilizará bastantes recursos. Esta es la razón por la cual todos los DBMS utilizan algún tipo de índice, algunos utilizando cierto tipo de "índices invertidos".

Las listas invertidas son una técnica de indexación en donde una columna puede ser consultada de tal forma que sea posible determinar que valor o valores han existido para cada renglón en dicha columna. Valores duplicados o no son permitidos de acuerdo al diseño del sistema.

Una técnica de creación de índices es la inversión, de aquí el término "índice invertido". Consultar el índice permite determinar rápidamente si un renglón contiene un valor en particular. No es necesario leer los renglones almacenados en el archivo físico, ya que es suficiente con el índice. Si éste además se mantiene ordenado (en forma ascendente, por ejemplo) entonces resulta más rápido el obtener los datos en un orden predeterminado.

El problema con esta técnica es que sólo permite determinar si un valor específico es encontrado, pero no indica cual es el renglón exacto que contiene dicho valor. Esto se debe a que no se almacena en el índice información sobre la localización física o el identificador del renglón.

Para solucionar el problema anterior, se han implementado diversas variaciones de inversión. Una de las más exitosas ha sido la implementada dentro de ADABAS. En este DBMS cada lista invertida contiene los valores existentes en los renglones e identifica cuales son los renglones en donde se encuentran dichos valores. Una vez que un renglón se identifica de acuerdo al criterio de selección, entonces es extraído por el DBMS y las columnas adicionales pueden ser presentadas. Adicionalmente, los identificadores de los renglones hacen posible el combinar y unir índices para satisfacer búsquedas complejas.

Si un DBMS no incluye identificadores de renglón en sus índices entonces tendrá problemas al tratar de resolver criterios de selección complejos. Podrá determinar cual de los índices involucrados contiene el menor número de valores (típicamente empleando una utilería para contar los registros y almacenar el número de valores que existen en cierto momento), pero posteriormente deberá traer cada renglón (potencialmente se tiene una lectura física) para determinar si los criterios

adicionales se satisfacen. Los DBMS de este tipo comúnmente cuentan con "optimizadores de búsquedas", cuya principal función es la de tratar de reducir esta sobrecarga .

Otros DBMS utilizan un identificador físico en los índices, pero esto puede causar problemas si los datos se mueven a un nuevo lugar físico dentro de la base de datos. Todas las referencias en los índices necesitan ser reconstruidas en determinado momento con una utilidad. Este proceso se conoce como reorganización. Tiene la desventaja que consume mucho tiempo y recursos; frecuentemente requiere que la base de datos, o al menos una tabla, no esté en servicio mientras se realiza la reorganización.

ADABAS soluciona este problema utilizando un identificador lógico en lugar de uno físico. La posición física de un renglón se almacena sólo dentro de una parte dentro de ADABAS: En una tabla llamada "address converter". Si un renglón necesita ser movido de un lugar a otro, sólo el address converter necesita ser actualizado (no todos los índices). La actualización del address converter se realiza en forma dinámica, evitando poner al DBMS fuera de servicio para reorganizarlo.

La arquitectura de ADABAS permite implementar el modelo de datos necesario de acuerdo a los requerimientos de la aplicación. Es importante señalar que ADABAS maneja las matrices necesarias, en lugar de requerir que las aplicaciones soporten los "detalles" de un diseño relacional de la estructura. Esta arquitectura permite la utilización de este DBMS para aplicaciones de negocio que requieran el manejo de datos no estructurados, como es el caso de los sistemas con contenido geográfico.

4GL: NATURAL

NATURAL es un lenguaje procedural diseñado para aplicaciones de misión crítica. La sintaxis de NATURAL se apega a los conceptos de programación estructurada y cuenta con interfaces a rutinas escritas en un 3GL. También soporta la inclusión de SQL ANSI así como las extensiones para DB2, Oracle, Rdb, Sybase y ADABAS.

El código NATURAL casi siempre es compilado. La excepción es cierto tipo de señales o marcas (tokens) necesarios para ciertas funciones del DBMS. Software AG vio que era complicado el manejar cierto tipo de procesos, como es el caso de los arreglos, cuando el producto era solo compilado.

Las aplicaciones run-time NATURAL son bastante eficientes debido a que Software AG optimiza sus productos para cada plataforma. El manejo de

buffer y optimización de I/O dan como resultado ganancias en performance en ambientes multiusuario.

El ambiente de desarrollo NATURAL soporta distintos tipos de ambientes, como es el caso de interfaces gráficas (GUI), interfaces modo caracter así como programas tipo batch. Los programadores utilizan el mismo lenguaje y el mismo ambiente para definir los componentes de una aplicación sin importar el tipo de interface a utilizarse. El ambiente de desarrollo incluye opciones para manejo de seguridad, versiones, ayuda en línea, varios editores, entre otras.

Existen otros componentes adicionales al lenguaje que proporcionan mayor soporte para el ciclo de vida de un sistema. Estos componentes incluyen un repositorio operacional y diccionario de datos activo llamado PREDICT; generador de código, NATURAL CONSTRUCT; herramienta CASE para análisis y diseño llamada NATURAL ENGINEERING WORKBENCH o NEW.

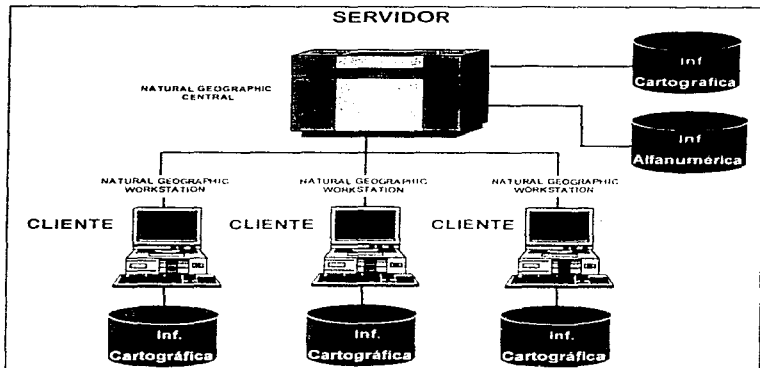
NATURAL GEOGRAPHIC

Comúnmente dentro de las bases de datos corporativas se encuentra información que tiene una relación geográfica. El poder contar con un modelo geográfico basado en esta información se convierte en un apoyo muy importante para la toma de decisiones.

NATURAL GEOGRAPHIC es una extensión al lenguaje NATURAL que permite crear Sistemas de Información Geográfica (GIS) con los cuales es posible manejar datos de origen cartográfico asociándolos a información de tipo alfanumérico manteniéndolos en un DBMS común.

Un ejemplo de las aplicaciones que se pueden crear con estas herramientas son sistemas para manejo de rutas de transporte, organización de servicios de emergencia (hospitales, policía, bomberos, etc.), investigación de mercado, consultas y modificaciones de Catastro, organización de redes de venta, redes de distribución de agua, gas, electricidad, etc.

La interactividad gráfica se resuelve mediante un proceso distribuido que incluye al servidor, que es donde reside la base de datos central, con las estaciones de trabajo como se muestra en el siguiente diagrama:



Las estaciones de trabajo pueden ser de varios tipos, desde equipos PC a estaciones gráficas UNIX. En las estaciones de trabajo se realizan dos procesos:

- Manejo de la interacción gráfica requerida por el usuario para realizar vistas, zooms, identificación de elementos en forma gráfica, etc.
- Conexión en línea con el servidor emulando una terminal estándar.

En el servidor este último proceso conecta la estación de trabajo a un conjunto de programas NATURAL que aseguran que el sistema funcione en tiempo real. Las estaciones de trabajo cuentan con varias utilerías que permiten el manejo de la información gráfica, como es la edición, impresión, definición de simbología, edición de diccionarios de códigos, etc.

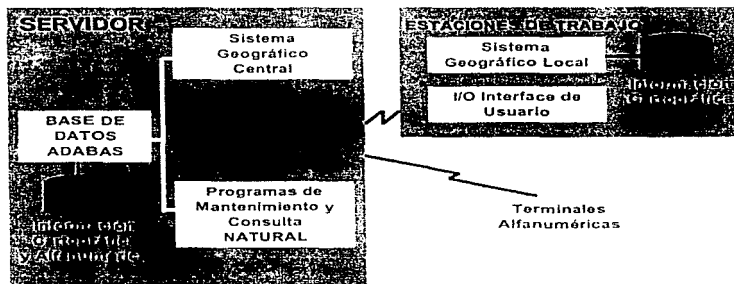
El consumo de recursos de CPU en el servidor es el mismo que para cualquier terminal alfanumérica ya que las operaciones gráficas en las estaciones de trabajo se realizan sin intervención del servidor. Para

lograr lo anterior se mantiene en el cliente una copia de la información gráfica. Esta copia se actualiza simultáneamente en la estación de trabajo y en el servidor después de cualquier modificación.

La interface de usuario está basada en menús configurables, de tal forma que se pueden adaptar a funciones específicas. La utilización de estos menús se realiza en un ambiente gráfico, lo cual permite aprovechar las ventajas de este tipo de interface, como es el manejo de mouse, ventanas, etc.

La arquitectura básica de NATURAL GEOGRAPHIC se muestra en la siguiente figura:

ARQUITECTURA BASICA DE NATURAL GEOGRAPHIC

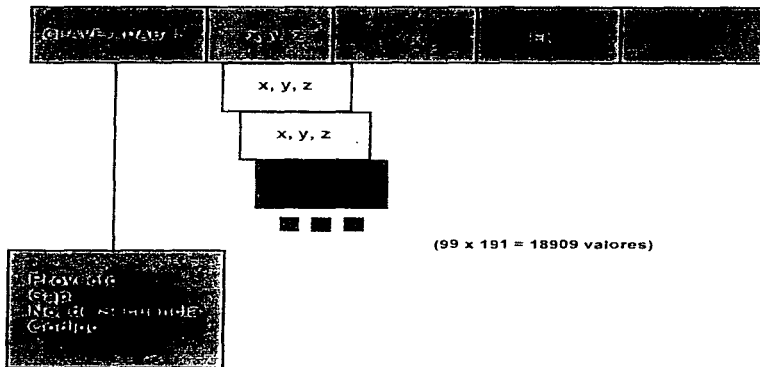


Las funciones aplicativos están escritas en NATURAL, lo cual proporciona todas las ventajas de contar con un 4GL; el diseño físico de ADABAS permite manejar la información gráfica como cualquier información, solucionando el problema del almacenamiento de elementos complejos (relación de elementos) y compuestos (con información alfanumérica implícita). Debido a que los datos se mantienen en registros ADABAS, el sistema proporciona las siguientes ventajas:

- Los datos geográficos son accesibles con la misma facilidad que cualquier otro dato y están disponibles para cualquier otro usuario, local o remoto.
- El manejo de los mismos se puede hacer mediante programas NATURAL que accedan a los campos que contienen las coordenadas de los distintos elementos cartográficos de base y a los demás campos del registro de NATURAL GEOGRAPHIC, de modo que el desarrollo de aplicaciones de tipo técnico (por ejemplo, calcular el área de una manzana o la pendiente de una carretera) se pueda hacer con la misma facilidad que otro tipo de aplicaciones.
- El tamaño de los registros es reducido debido a la compresión de datos que realiza ADABAS.
- Existe bastante seguridad en el manejo y control de la información.
- Opera en tiempo real ofreciendo integridad transaccional.
- Al poder definir campos múltiples (campos de un registro con múltiples ocurrencias) es posible utilizar el mismo diseño de registro de ADABAS para almacenar elementos singulares o con texto (elementos de un sólo punto), así como elementos lineales (con un número indeterminado de puntos).
- La integración de datos que se logra con NATURAL GEOGRAPHIC hace que se puedan manejar en forma global desde cualquier área que esté conectada a la base de datos que utiliza el sistema.

La siguiente figura muestra como se pueden representar estructuras complejas de datos dentro de ADABAS:

REGISTRO BASICO ADABAS



NATURAL GEOGRAPHIC soporta varias opciones para la entrada y salida de datos, tanto de tipo alfanumérico como gráfico. La información gráfica se introduce al sistema a partir de mapas en papel, fotografía aérea, trabajos topográficos de campo así como diversos formatos gráficos.

Cuando los datos de partida son mapas de papel, éstos se digitalizan y se utiliza un modelo de captura masiva mediante archivos de código que permiten la clasificación de toda la información recogida. Además, con el módulo de calibración es posible considerar las posibles deformaciones (no lineales) del papel en el mapa y/o la descalibración de la mesa digitalizadora al capturar los datos, de tal forma que cada mapa se puede calibrar en forma particular reduciendo errores.

Si los datos se originan con fotografías aéreas del terreno, NATURAL GEOGRAPHIC cuenta con un módulo de hardware que permite conectarse directamente a cualquier restituidor fotogramétrico (stereo comparator).

Para capturar trabajos topográficos de campo, existen varias funciones que permiten la generación directa del mapa a partir de un croquis de campo. Adicionalmente es posible utilizar datos cartográficos de otros sistemas; existen interfaces para los formatos IGES y SIF de salida, DXF de Autodesk Inc. entre otros. También es posible incluir información a partir de la digitalización de documentos o cualquier imagen tipo raster.

Es posible obtener mapas de precisión al utilizar las funciones de impresión a plotter. Este módulo contempla aspectos como el formato de hoja, escala de representación, tipos de línea compuestos, simbología, etc.

Existe un módulo editor de símbolos y un formato específico para la definición de líneas.

Cartografía Urbana Digitalizada

Inicialmente se consideró la opción de digitalizar un mapa en papel albanene de la zona fraccionada de la Delegación Tlalpan. Sin embargo se evaluaron otras alternativas, específicamente el SCINCE de INEGI así como Guía Roji. Al analizar la información de ésta última se detectó que no se conseguía una resolución aceptable al realizar acercamientos o "zoom" con el producto debido a que Guía Roji utiliza información tipo raster, es decir, digitalizada con un scanner, lo cual hace que un elemento (por ejemplo, una línea en escala inicial) se vaya deformando conforme se "agrandan" la imagen. Por otro lado, la información del SCINCE de INEGI es de tipo vectorial (en términos de tipo de imagen) lo cual significa que al hacer una manejo cartográfico no existen deformaciones ni aproximaciones por lo que es posible hacer un manejo muy preciso de todo tipo de elementos cartográficos.

La cartografía digitalizada utilizada entonces es la creada por el INEGI. Esta es un producto del XI Censo Nacional de Población y Vivienda 1990, que además de obtener información sobre las características demográficas, educativas y económicas de la población residente en el país sirvió para generar diversos productos cartográficos bajo un esquema de planeación, análisis, consulta y experimentación, que permitió garantizar la calidad y oportunidad de los resultados.

Los productos cartográficos son resultado de un proceso de cuatro años de acopio, elaboración y actualización en campo de cada uno de ellos con el trabajo de una estructura estatal a nivel nacional especializado en cartografía.

La cartografía no solo sirvió para la realización de las etapas de planeación, levantamiento y procesamiento de la información, sino que su calidad permite que se pueda divulgar como un resultado adicional del censo.

Los planos urbanos son un elemento indispensable en la elaboración de planes y programas de desarrollo urbano, así como en la realización de estudios específicos tales como análisis geoelectorales.

La integración de la cartografía se realiza a partir de los planos de las localidades urbanas cuya información hace posible conocer la ubicación geográfica de cada una de ellas y sus principales características como nombre de las calles, configuración de las manzanas e identificación de los servicios básicos de cada localidad. Adicionalmente, esta cartografía muestra la división geográfico-estadística usada en el censo de 1990.

La digitalización de 4384 localidades urbanas consumió 30 meses con 90 personas en 3 turnos. Esta operación se realizó de manera descentralizada.

Marco Geoestadístico Nacional

El marco geoestadístico nacional es un sistema de referencia geográfica que para fines de generación estadística ha creado el INEGI. Consiste en una división de todo el territorio nacional en áreas de fácil identificación en campo y extensión adecuada para las actividades de captación de información. Dichas áreas se denominan "áreas geoestadísticas básicas (AGEB)". En relación con la división político-administrativa del país, el marco geoestadístico se ajusta en lo posible a los límites de los municipios (delegaciones para el caso del distrito federal) y las entidades federativas.

Definición de AGEB

El AGEB es un espacio geográfico delimitado mediante rasgos naturales o culturales permanentes y reconocibles en el terreno, con extensión convencional al interior de cada municipio del país y que facilita la captación y referencia geográfica de información.

Dadas las diferencias de densidad de la población, se utilizó la dicotomía rural-urbana a fin de distinguir al menos dos tipos. Las AGEB urbanas delimitan una parte o el total de una localidad de 2500 habitantes o más conforme al catálogo de localidades de 1980, o una cabecera municipal independientemente del número de habitantes. Como resultado de la

diversidad de actividades económicas y usos del suelo en las áreas urbanas, este tipo de AGEB puede cubrir tanto espacios habitacionales como de otras características (áreas recreativas, establecimientos económicos, reservas territoriales, etc.). A diferencia de las urbanas, las AGEB rurales cubren una superficie donde no existen localidades de 2500 o más habitantes ni cabeceras municipales.

La cartografía se conforma de mapas y planos con distintos niveles de cobertura geográfica y escalas de representación. La planeación y ejecución del censo nacional de población y vivienda 1990 implicó la actualización de mapas y planos, así como la reproducción y distribución oportuna de 1.7 millones de documentos cartográficos para el levantamiento censal.

Características técnicas de la cartografía

El acceso a los archivos gráficos es posible con cualquier paquete que permita la importación y manejo de datos en formato DXF (Draw Extend Format).

La cartografía está digitalizada en formatos que tienen un tamaño en coordenadas geográficas equivalente a 2' de longitud por 180'' de latitud y cada formato constituye un archivo independiente con su propia clave de identificación y tira marginal.

6.6 Identificación de Requerimientos

En base a los antecedentes y a entrevistas realizadas con el personal de sistemas de la Delegación Tlalpan, se determina que el uso de suelo es uno de los puntos más importantes en el Sistema Integral de Demanda Ciudadana, por lo que se desea implementar en el prototipo algunas funciones relacionadas.

De acuerdo a la ubicación se tiene un tipo o uso de suelo, por ejemplo H1, en donde H1 significa "Habitacional. Hasta 100 habitantes por hectárea". Por colonia y manzana es como se controla el uso de suelo, por lo cual se requiere el poder realizar consultas que a partir de una ubicación (colonia y manzana) se pueda conocer en forma gráfica y alfanumérica los servicios válidos en esa zona. Otro tipo de consulta sería aquella en la cual se pregunta en que lugares es legal construir una escuela o un restaurante, por ejemplo.

También se requieren consultas gráficas que muestren elementos del equipamiento urbano, por ejemplo ver todos los hospitales o escuelas existentes en la delegación.

Actualmente ya se cuenta con el Catálogo de Colonias y Manzanas dentro de ADABAS, cuya descripción y DDM se incluyen. El objeto de este catálogo es el de conocer que calles delimitan a cada manzana al norte, sur, oriente y poniente. Incluye información como los tipos de suelo de la manzana, por ejemplo.

6.7 Metodología Empleada

De acuerdo a las necesidades del proyecto se determinó que lo mejor sería utilizar un esquema de trabajo que permitiera la máxima interacción con el usuario de tal forma que se pudieran dar resultados reales en corto tiempo.

El utilizar alguna metodología tradicional no era la mejor opción ya que se trataba de sólo una parte del sistema. Sin embargo, fue necesario el considerar algún tipo de estrategia que permitiera construir un prototipo inicial de acuerdo a lo planteado pero que tuviera la capacidad de evolucionar para posteriormente incluir mayor funcionalidad.

Existen diversas ideas y conceptos distintos en relación a los prototipos. De la variedad de métodos, el prototyping rápido estructurado o Structured Rapid Prototyping o SRP fue seleccionado.

Un prototipo rápido de software es un modelo visual dinámico que funciona como herramienta de comunicación entre cliente y desarrollador siendo más efectivo que la narración o los modelos visuales estáticos. Ha sido descrito como:

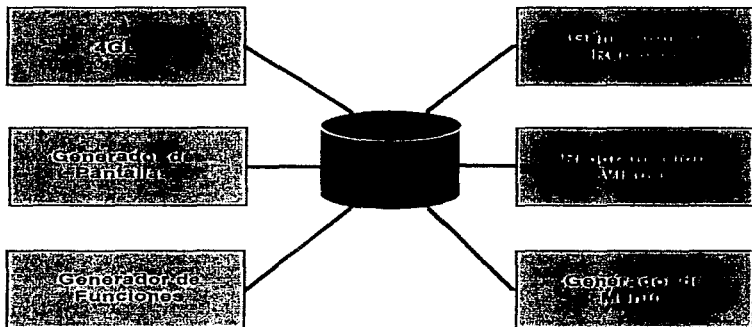
- Funcional después de una cantidad mínima de esfuerzo.
- Un medio de proporcionar a los usuarios de la aplicación una representación física de las partes claves del sistema antes de la implementación de éste.
- Flexible. Las modificaciones requieren poco esfuerzo.
- No necesariamente representa un sistema completo.

Las técnicas del prototyping evolucionable pueden aplicarse perfectamente a todas las fases del ciclo de vida del software ya que los prototipos aprobados por el usuario siempre son candidatos para afinarse y evolucionar a sistemas de producción a diferencia de otras estrategias en donde el prototipo es desechado, por lo cual utilizando prototyping en muchas ocasiones se tienen ciclos de desarrollo más cortos y por lo tanto costos menores en comparación con el desarrollo tradicional u otros esquemas para el desarrollo de prototipos.

El presentar las diversas estrategias para desarrollo de prototipos sale del objetivo de este trabajo, sin embargo vale la pena mencionar que a diferencia de simplemente "sentarse a programar" el SRP incluye análisis y diseño detallados, por lo cual las técnicas de especificación y desarrollo estructurado son parte integral de la estrategia de SRP. Las técnicas para prototipos evolutivos son aplicadas concurrentemente con análisis

estructurado al inicio del proyecto. Las mismas técnicas se aplican junto con diseño y programación estructurada durante la fase de afinación. Puede parecer confuso, pero se debe tener presente que estamos hablando de un proceso iterativo y no secuencial. SRP utiliza datos reales, sin embargo no debe perder la facilidad de modificarse, ya que el volumen de datos es grande y el proceso de entrada de datos es manual. Las herramientas adecuadas para el prototyping proporcionan a los desarrolladores la forma de modificar rápidamente las estructuras de almacenamiento conservando la información existente.

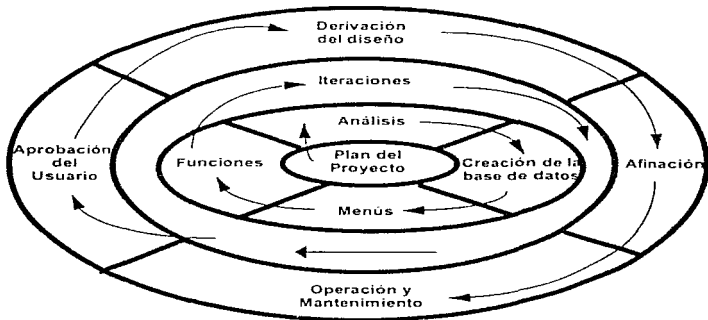
Las técnicas de prototyping involucran el uso de nuevas herramientas como lenguajes de cuarta generación y bases de datos relacionales, por lo cual es necesario que las personas relacionadas con el desarrollo de un prototipo comprendan y tengan una actitud abierta con respecto a la utilización de nuevas técnicas y herramientas. Es muy importante considerar este factor humano ya que el personal técnico puede ver a las nuevas técnicas como una crítica a la forma en que hacen las cosas, lo cual puede causar que se minimicen o no se les de importancia a conocimientos con los que no están familiarizados.



Ambiente Ideal para Prototyping

El proceso iterativo de revisión y demostración del prototipo puede continuar para siempre si no existe un control adecuado del proyecto. Conforme los usuarios ven que sus requerimientos son satisfechos puede existir una tendencia a que aumente la lista de cosas a incluirse en el prototipo hasta que el enfoque del proyecto queda fuera del estudio de factibilidad. Es indispensable crear un plan del proyecto como primera actividad y revisarlo continuamente. Si el usuario lo desea y los desarrolladores cuentan con los recursos, este plan puede ser extendido infinitamente. La importancia de este plan es el poder comunicarle al cliente que conforme los elementos a entregar aumentan o la complejidad de las funciones se incrementa también lo hacen el plan de trabajo y el costo asociado.

El siguiente diagrama muestra un modelo del proceso para la creación rápida de prototipos estructurados:



A continuación se presentan los beneficios más importantes de utilizar las técnicas de SRP:

Control de Calidad

SRP puede ser considerado como una técnica para verificación de calidad o Quality Assurance, ya que el beneficio primario y objetivo principal es aumentar el nivel de satisfacción del cliente con respecto a la funcionalidad del software entregado. Esto se debe a que este software contendrá menos defectos funcionales al contar con la involucración del usuario desde la fase de análisis.

Acomodar nuevos requerimientos del usuario

Muchas veces el usuario no puede comprender y detallar claramente los requerimientos funcionales antes de tener la oportunidad de experimentar interactivamente con las opciones. Esto causa que con esquemas tradicionales de desarrollo sea más probable que el usuario quiera cambiar ciertas características del sistema en una etapa avanzada del ciclo de desarrollo, que es cuando tiene oportunidad de ver el sistema funcionando por primera vez. Muchas veces la realidad es muy distinta a los conceptos vistos y creados en papel. También es común que al ver una función en operación se identifique alguna necesidad de contar con funcionalidad adicional. SRP proporciona la oportunidad de manejar estos cambios, que básicamente representan ideas creativas del usuario en las fases iniciales del ciclo de vida del sistema y por lo tanto a bajo costo.

Evolución

El prototipo puede ser evolucionado a un sistema más completo debido a que las técnicas utilizadas (DFD, ERD) permiten modificaciones y extensiones a la primera implementación. De igual forma, éstas se pueden integrar al resultado de proyectos paralelos que formen un sistema global.

6.8 Proceso de creación del sistema

Como se mencionó previamente, las tareas que se realizan al utilizar una metodología de prototyping son un poco distintas y tienen una secuencia diferente que cuando se desarrolla un sistema empleando la metodología tradicional.

Las tareas a realizarse, aunque se presentan en forma secuencial, empezando por la planeación del proyecto hasta la presentación del modelo funcional al usuario para revisión, ocurren en su mayoría simultáneamente.

El modelo inicial intencionalmente no está completamente especificado y es parcialmente desarrollado con el objeto de que el agregar o modificar funciones resulte más eficiente que un total reemplazo de éstas. De esta manera el desarrollo del prototipo y su evolución incluirá solamente la funcionalidad necesaria.

La primera actividad es la especificación del plan del proyecto, el cual también evolucionará posteriormente. La utilidad de este plan es el mostrar el objetivo y beneficios del proyecto, su alcance, herramientas a utilizarse, resultados y plan de trabajo inicial.

Los siguientes puntos son los que se mostraron a la Delegación Tlalpan (usuario) y se utilizaron como plan del proyecto:

Justificación del Proyecto

Es de suma importancia llevar un control efectivo del uso de suelo en la Delegación, ya que actualmente no se conoce con certeza si el equipamiento urbano y servicios en determinadas ubicaciones van de acuerdo al tipo de uso de suelo permitido en dichas ubicaciones, lo cual afecta en distintos aspectos como pueden ser la seguridad ciudadana o bien el cobro de impuestos.

Objetivo

Mostrar la funcionalidad del producto NATURAL GEOGRAPHIC junto con ADABAS y NATURAL con la creación de un prototipo que sea de utilidad para el control del equipamiento urbano y uso de suelo de la Delegación Tlalpan y que tenga la capacidad de evolucionar para cubrir o complementar a los demás módulos del Sistema Integral de Demanda Ciudadana.

Alcance

El prototipo que se propone incluirá los módulos de Zonificación de Uso de Suelo y Equipamiento Urbano de la Delegación así como la parte cartográfica necesaria para la representación gráfica de la información. Estos módulos forman parte del Sistema Integral de Demanda Ciudadana.

De acuerdo al planteamiento de la Delegación se acordó que el prototipo abarcaría información de tres colonias de la Delegación Tlalpan (Colonia Isidro Fabela, Colonia Toriello Guerra y Colonia Ampliación Isidro Fabela). Inicialmente sólo se considerarán inmuebles como hospitales, escuelas, mercados, restaurantes, hoteles, etc. pero que en un futuro se incluirán otro tipo de objetos como luminarias y alcantarillas, por ejemplo.

Herramientas de desarrollo

Se utilizará la base de datos ADABAS y el lenguaje de cuarta generación NATURAL con NATURAL GEOGRAPHIC como herramienta para incluir y manejar información cartográfica. Como fuente de información (mapas) digitalizados se utilizará la del SCINCE (INEGI).

Se crearán los siguientes diagramas: DFD (Data Flow Diagram), ERD (Entity-Relationship Diagram) y CFG (Control Flow Diagram).

Responsabilidades del usuario

El usuario participará en la creación de los diagramas DFD, ERD y CFG. Participará en las revisiones del sistema, proporcionará datos para prueba así como toda la información necesaria acerca de los catálogos ya existentes. También proporcionará los recursos necesarios en el equipo servidor y cliente para la creación del sistema. Finalmente revisará y dará la aprobación final del sistema.

Resultados a obtenerse

El sistema contará con una interface gráfica creada con NATURAL GEOGRAPHIC que permita la interacción y uso de programas NATURAL accediendo en el servidor a la base de datos ADABAS. Se entregará documentación técnica y de usuario. El prototipo a entregarse permitirá:

- Capturar una solicitud de demanda ciudadana.
- Consultar en forma gráfica o alfanumérica una demanda ciudadana a partir del folio.

- Consultar en forma gráfica o alfanumérica información de la zonificación del uso de suelo.
- Localizar una Entidad Federativa (D.F.).
- Localizar un municipio (Delegación Tlalpan).
- Presentar la carta urbana.
- Localizar una manzana.
- Proporcionar información alfanumérica de una manzana seleccionada.
- Localizar equipamiento urbano.
- Proporcionar información alfanumérica del equipamiento seleccionado.
- Impresión de vista cartográfica.

A continuación se muestra el plan de trabajo del proyecto:

Descripción de la Tarea	Abril				May				Jun				Julio		
	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10
1 Análisis Requisitos															
2 Creación y carga de la base de datos															
3 Creación de los programas NATURAL															
4 Probar funcionalidad de los programas															
5 Modificar y mejorar el sistema															
6 Especificación detallada de requisitos															
7 Especificación del diseño preliminar															
8 Creación de menús en la interfaz gráfica															
9 Documentación															
10 Pruebas y entrega															
11 Especificación del diseño final															
12 Pruebas finales															
13 Entrega y validación															

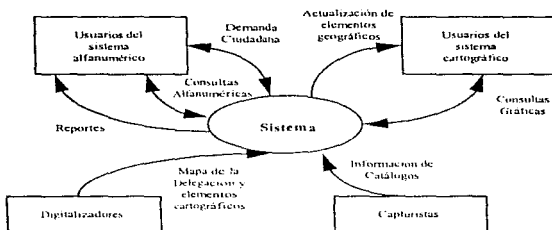
Análisis

A partir de distintas entrevistas y sesiones con los usuarios se obtuvieron los diagramas necesarios para el sistema.

Existen dos tipos de usuarios que utilizan información del SIDC: los que pueden utilizar la parte alfanumérica del sistema o los que pueden realizar consultas alfanuméricas y gráficas. La información sobre manzanas y equipamiento urbano es capturada por personal dedicado.

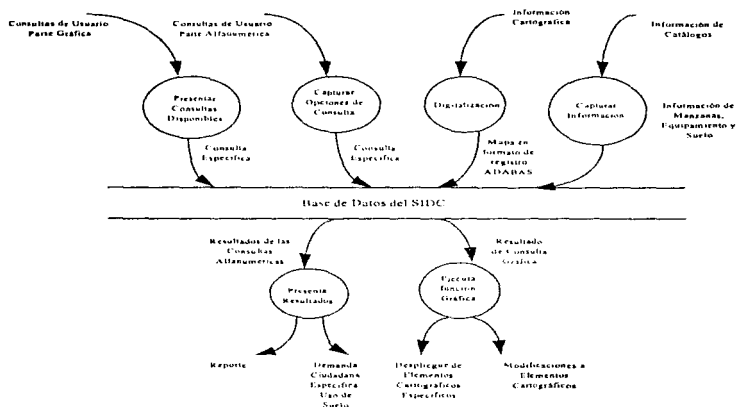
La digitalización ya está hecha, por lo cual sólo es necesario utilizar una utilidad que permite convertir archivos con formato .dxf a formato registro de ADABAS y cargar dicha información a la base de datos.

El DFD de contexto muestra con que y con quien interactuará el sistema, que básicamente son los usuarios, digitalizadores y capturistas. A continuación se muestra el diagrama de contexto del sistema:



De acuerdo al diagrama, los usuarios de la parte alfanumérica del sistema pueden realizar consultas, obtener reportes así como colocar una demanda ciudadana; los usuarios de la parte gráfica pueden hacer consultas en el mapa y recibir actualización de los elementos cartográficos en caso de que hayan sido modificados; el digitalizador se encarga de convertir la información cartográfica en un formato que sea manejado por el sistema; los capturistas se encargan de introducir la información sobre manzanas y equipamiento urbano.

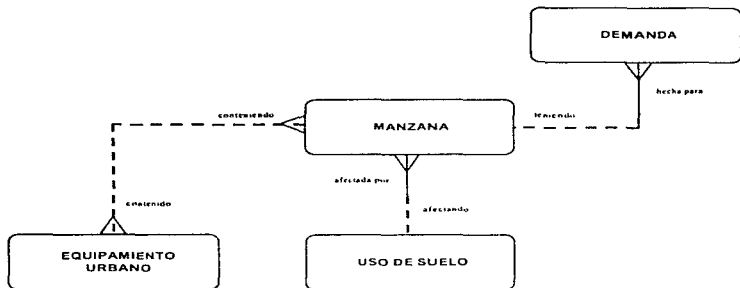
El DFD de contexto proporciona una visión general del sistema, sin embargo es necesario mayor nivel de detalle, por lo cual este diagrama se partirá en un DFD de funciones esenciales, el cual se muestra a continuación:



Este diagrama muestra la información que fluye hacia la base de datos, es decir, las consultas específicas, datos sobre nuevas demandas ciudadanas, el mapa de la delegación en formato de registro ADABAS así como la información sobre manzanas, equipamiento urbano y uso de suelo. A partir de esta información se obtienen los reportes o resultados de las consultas, despliegue gráfico de los elementos solicitados así como las modificaciones a los elementos cartográficos en caso de que existan.

Por cuestiones físicas asociadas a la creación del prototipo, se dibuja un depósito de datos del cual todas las funciones toman o guardan información. Este depósito representa a la base de datos que se diseñará.

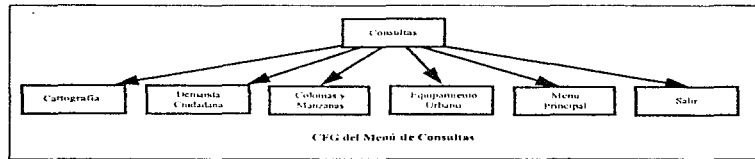
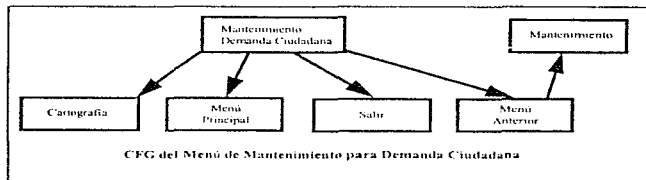
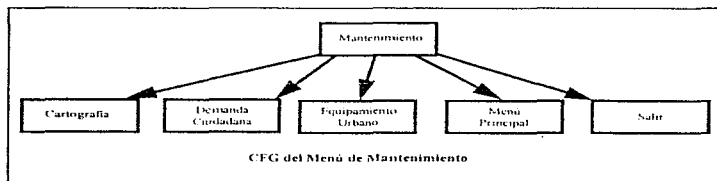
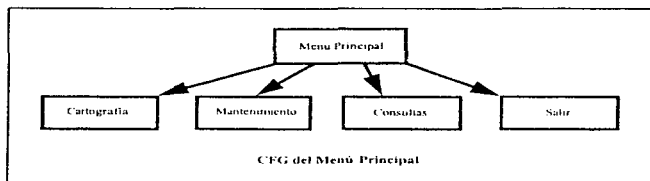
El diagrama ERD refleja los datos que serán almacenados y utilizados por el sistema. Como ya se mencionó, algunas tablas corresponden a catálogos ya existentes en otros sistemas. El ERD nos proporciona otra manera de describir al sistema:

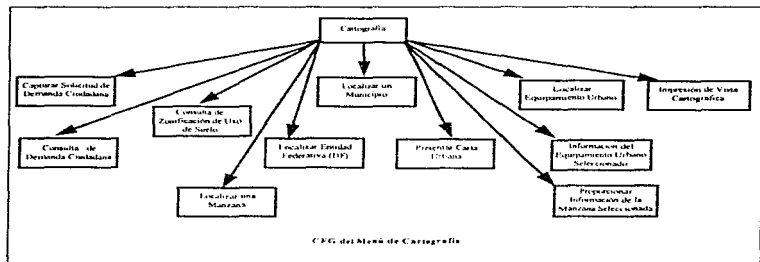
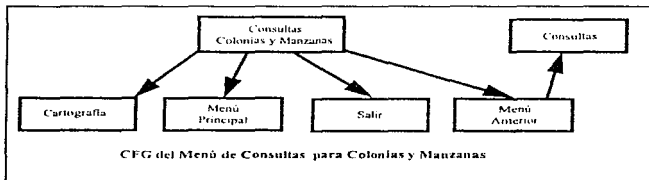
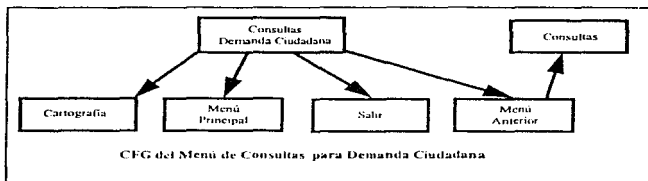


El diagrama puede ser considerado como una especificación sin ambigüedades de los requerimientos de información del usuario. El diagrama anterior nos indica:

- Una manzana puede estar conteniendo uno o más equipamientos urbanos.
- Un equipamiento urbano puede estar contenido en una y solo una manzana.
- Una manzana puede estar teniendo una o más demandas.
- Una demanda debe estar hecha para una y solo una manzana.
- Una manzana debe estar afectada por uno y solo un uso de suelo.
- Un uso de suelo puede estar afectando a una o más manzanas.

Antes de empezar a construir el primer modelo del sistema es necesario crear un diagrama que represente los eventos y el flujo del control en el sistema. Un CFG (Control Flow Graph) representa como el usuario verá al sistema desde su estación de trabajo. El CFG es la forma de especificar la secuencia de eventos de acuerdo a la necesidad del usuario. De acuerdo a este diagrama se crearán las pantallas y menús necesarios. A continuación se muestran los CFGs para este prototipo:





Es necesario mencionar que los diagramas mostrados son las versiones revisadas por el usuario, ya que cada diagrama tuvo en promedio tres ciclos de revisión y ajuste. Esto no significa que estos diagramas tengan que ser la versión final del sistema, sino que puedan reflejar los requerimientos reales del usuario antes de empezar a escribir programas y crear la base de datos. Gracias a la utilización de técnicas estructuradas estos diagramas pueden evolucionar a las especificaciones más detalladas que son necesarias en una metodología estructurada tradicional. Al ser necesaria mayor funcionalidad se pueden agregar procesos a los DFDs; el ERD puede expandirse para incluir más entidades; nuevos módulos pueden agregarse al CFD.

Desarrollo del prototipo

Después de la etapa de análisis se comenzó la construcción del prototipo. Los objetivos inmediatos fueron:

- Obtener un modelo experimental de los aspectos claves del sistema frente al usuario y de manera rápida.
- Asegurarse de que este modelo es tan fácil de modificar como los modelos en papel producidos por las especificaciones estructuradas convencionales.
- Incluir en el prototipo datos con los que el usuario esté familiarizado de tal forma que le sea fácil verificar los requerimientos funcionales y de información.
- Utilizar una estructura de almacenamiento de datos que sea fácil de modificar conforme los requerimientos de información evolucionen durante las iteraciones del prototipo.

Los pasos a seguir para lograr estos objetivos fueron:

1. Creación de una base de datos.
2. Creación de tablas en esa base de datos.
3. Cargar con datos existentes las tablas, si es posible. De lo contrario crear datos de prueba.
4. Crear pantallas y menús.
5. Crear módulos funcionales de alto nivel que puedan ser llamados desde los menús.

Creación de la base de datos y tablas

Se generó la base de datos NATGEO. En base al diagrama ER se crearon las siguientes tablas en la base de datos:

Nombre Físico	Nombre Lógico
ME-CATALOGO-MANZANAS	Manzana
ME-CATALOGO-CIUDADANA	Demanda
ME-EQUIP-URBANO	Equipamiento Urbano
ME-USO-SUELO	Uso de Suelo

Las siguientes tablas forman parte del módulo servidor de NATURAL GEOGRAPHIC:

NG3-DES-CURVAS
NG3-DES-POLIGOS
NG3-DES-PUNTOS
NG3-DES-RASTER
NG3-DES-TEXTO
NG3-DES-USER

El detalle y atributos de cada una de estas tablas se incluye en la documentación técnica.

Carga de la base de datos

A partir de información ya existente en archivos de los catálogos de Tipos de Bienes Inmuebles (Equipamiento Urbano), Uso de Suelo, Colonias por Zona y Manzanas por Colonia, se cargaron con información de la Delegación las tablas Equipamientos, Suelos y Manzanas.

Creación de pantallas, menús y módulos funcionales

El desarrollo de los distintos programas y pantallas se hizo utilizando el ambiente integrado de desarrollo de NATURAL.

En lenguaje NATURAL se denomina mapa a una pantalla de información. El ambiente de desarrollo incluye un editor especial para la creación de mapas. Las pantallas o mapas se van creando en forma visual de tal forma que resulta muy sencillo agregar campos o texto descriptivo. Una vez que se tiene el mapa como se desea es necesario compilarlo para crear un objeto que pueda ser usado por un programa.

La codificación de programas se hace utilizando el editor de programas. Cada programa puede definir su área de datos, que es donde se definen las variables que se usarán; es posible crear módulos independientes que definan variables para después ser incluidos en los programas.

Un programa hace referencia a los mapas de acuerdo al tipo de función que se requiera. Después de incluir los objetos necesarios dentro del programa es necesario compilarlo para verificar que no existan errores de sintaxis y así tener un objeto tipo programa que a su vez puede ser llamado desde otro programa.

A continuación se muestran los nombres de los programas y mapas desarrollados con la descripción de su función:

Programa	Mapa	Descripción
MEXP0001	MEXM0001	Menú Principal
MEXP0002	MEXM0002	Menú de Consultas
MEXP0004	MEXM0007	Consulta Alfanumérica de Manzanas
	MEXM0005	Resultado de la Consulta Alfanumérica de Manzanas
MEXP0005	MEXM0004	Menú de Mantenimiento
MEXP1004	MEXM0007	Consulta Gráfica de una Manzana
MEXP4000	MEXM0014	Captura de Demanda Ciudadana
MEXP4001	MEXM0015	Consulta Alfanumérica de Demanda Ciudadana
	MEXM0013	Resultado de la Consulta de Demanda Ciudadana
	MEXM0006	Despliega información de Equipamiento Urbano

El listado de los programas y las pantallas anteriores se incluye en la documentación técnica.

Presente y Futuro

"Es posible que las computadoras del futuro no pesen arriba de 1.5 toneladas"

-Mecánica Popular, 1949.

Los analfabetos del siglo 21 no serán aquellos que no puedan leer y escribir sino más bien quienes no puedan aprender, desaprender y volver a aprender.

Alvin Toffler.

7.1 Introducción

Actualmente el avance tecnológico tanto en hardware como en software lleva una velocidad impresionante que hace prácticamente imposible para la mayoría de las personas, e inclusive a gran cantidad de empresas, el poder seguir de cerca todos los acontecimientos que suceden alrededor de las nuevas tecnologías. En muchas ocasiones esto provoca un efecto de "congelamiento" en el cual las decisiones sobre nuevos proyectos se atrasan debido a largos procesos de evaluación en donde los parámetros más importantes de decisión siguen siendo las características individuales de cada producto, especialmente en el caso del software. Sin embargo en la actualidad un enfoque correcto del negocio, así como la selección adecuada de tecnología significan la permanencia o desaparición de muchas compañías.

El objeto de este capítulo es mostrar la evolución y tendencias más importantes en cuanto a los conceptos que se mostraron al inicio de este trabajo, indicando que tipo de tecnología podría ser utilizada en caso de que se desarrollara de nuevo gran parte del sistema presentado o que éste tuviera algún tipo de extensión a nivel funcional.

7.2 Perspectiva

La dinámica en el ámbito de los negocios y la economía, el surgimiento de nueva tecnología así como la evolución de las capacidades en las plataformas existentes ha dado como resultado la redefinición del rol de cada una de éstas, lo cual forma parte de un nuevo paradigma en computación. A continuación se muestran las características más importantes de las opciones existentes cuando se inició el proyecto y su situación actual.

Mainframes

A pesar de que ya no son percibidos como tecnología de punta, muchos sistemas basados en mainframe todavía continúan en operación después de dos décadas. La mayoría son utilizados en aplicaciones de misión crítica como es el manejo de transacciones bancarias y reservaciones en aerolíneas.

El problema con esta plataforma sigue siendo el largo ciclo para implementar cambios a las aplicaciones. Nuevas técnicas para desarrollo de sistemas así como plataformas aparentemente menos costosas fueron vistas en su momento como la solución. Sin embargo existen varias razones por las cuales el incremento en la productividad no fue tan espectacular como se esperaba. A pesar de las desventajas de estos ambientes, los sistemas basados en mainframe funcionan bien y de manera confiable.

Las características operacionales pueden garantizar una disponibilidad mayor al 99.9 por ciento gracias a la madurez de los componentes de software y herramientas de administración. Es común encontrar ambientes con más de 2000 usuarios gracias a la utilización de monitores de teleproceso,

El crecimiento y avance de esta plataforma cada vez es más lento en comparación con las demás, por lo cual en muchas ocasiones los requerimientos de usuario exceden las capacidades de los ciclos por CPU que estos equipos pueden proporcionar. Las alternativas han sido repartir el trabajo en más CPUs o bien migrar ciertas cargas de trabajo a otras plataformas, aunque el crecimiento horizontal es en muchas ocasiones complicado y costoso. Sin embargo la nueva generación de mainframes basados en tecnología CMOS proporciona una alternativa interesante de solución al problema de poder de proceso del CPU, si bien

hay que asegurarse de que el código de las aplicaciones (algunas con más de 20 años) es capaz de aprovechar arquitecturas basadas en múltiples procesadores.

Las mejoras funcionales son difíciles de implementar. Un cambio simple como el agregar un campo nuevo a una pantalla o cambiar un campo de entrada de datos puede llevar varias semanas para realizarse. La incapacidad de responder rápidamente a los cambios en el negocio puede ser motivo de frustración.

Adicionalmente a lo anterior existe el problema latente relacionado con el cambio de milenio, que a pesar de ser un problema que afecta múltiples plataformas, será ésta la más afectada debido a la dificultad que representa el corregir código escrito en muchos casos hace más de 25 años y que en ocasiones ni siquiera se cuenta con los programas fuente.

La interface de usuario es muy limitada. Se puede mejorar utilizando un emulador 3270 en una PC o bien algún software que proporcione una interface gráfica. Sin embargo, para este tipo de productos, la distribución y control del código son problemas importantes.

El costo de estos ambientes es elevado e insensible a las mejoras en cuanto a precio/rendimiento de la tecnología, principalmente por los costos fijos en cuanto a soporte, personal para administrar el ambiente y mantenimiento de las instalaciones en conjunto con hardware de alto costo que debe ser amortizado a lo largo de varios años.

Cliente/Servidor

Uno de los principales motivadores para esta arquitectura fue la disponibilidad de una gran variedad de aplicaciones comerciales con mucha funcionalidad y que podían ser implementadas más rápido y con menor costo que en mainframes. Adicionalmente, la adopción de estándares de sistemas abiertos significó la existencia de múltiples opciones de hardware y bases de datos en el mercado. Típicamente a nivel operativo estos ambientes pueden soportar entre 1000 y 1500 usuarios como máximo en un sólo servidor. Sin embargo estas plataformas son menos confiables que un mainframe debido a la inmadurez de la tecnología y la existencia de herramientas de administración incompletas.

Aparentemente el crecimiento de un ambiente cliente/servidor es más sencillo que en un ambiente mainframe. Nuevos clientes y servidores más

potentes pueden ser agregados fácilmente, asumiendo que se implementa una configuración estándar de hardware y software con todos los componentes probados previamente y con un sistema eficaz para el control de versiones. Teóricamente, el poder en los servidores puede crecer horizontalmente al dividir la carga entre varios equipos siempre y cuando se utilice tecnología de bases de datos distribuidas.

El incluir mejoras funcionales es bastante sencillo, ya que la separación a nivel lógico y físico entre la lógica de la aplicación y los datos hace que cambios simples al código se puedan realizar en minutos o en horas.

La interface de usuario es excelente, ya que se pueden utilizar los GUIs más comunes con muchas herramientas de desarrollo para crear sistemas completos con ayudas y documentación en línea.

El costo de estos ambientes ha sido considerado menor al de los mainframes, sin embargo muchas veces se excluyen los costos de mantenimiento o bien éstos se diluyen entre los diversos departamentos de una organización. Sin embargo, existe mayor sensibilidad a las mejoras en cuanto a precio/rendimiento de la tecnología. Gartner Group realizó un estudio comparativo de los costos de cliente/servidor contra mainframe. Al tener un análisis detallado que incluía los "pequeños" costos por concepto de tener a los propios usuarios como administradores de sus sistemas, los costos reales del almacenamiento en PCs (por ejemplo, espacio desperdiciado al no borrar versiones viejas de software) y los costos por concepto de no realizar respaldos que son implícitos en la operación de un centro de datos, el esquema cliente/servidor resultó ser más costoso en un periodo de cinco años de operación.

A pesar de las ventajas de estos ambientes en comparación con los mainframes, todavía existen varios problemas. Una limitación es la administración de configuraciones y el control de versiones. No hay duda de que es extremadamente complicado administrar más de 1000 estaciones de trabajo remotas. El esfuerzo necesario para validar que cumplen con las especificaciones y que están conectadas en forma correcta es bastante considerable. Se requiere algún tipo de automatización para estas funciones, especialmente para sincronizar las versiones de software que existen en cada PC.

La distribución de software es otra limitante. Es un reto a nivel logístico el distribuir nuevos cambios en las aplicaciones o parches en el software a más de 1000 puestos de trabajo remotos. Adicionalmente, debe existir cierto grado de certidumbre de que el nuevo software sea instalado en cada equipo.

La administración del sistema es otro punto delicado. En los mainframes existe un número pequeño de "cajas" como son los procesadores de comunicaciones, controladores de terminales y el mainframe en sí. En un sistema cliente/servidor grande existen más de 1000 máquinas. El software necesario para administrar un sistema de este tipo garantizando altos niveles de disponibilidad debe ser suficientemente completo; es bastante cuestionable el hecho de contar en la actualidad con herramientas maduras para realizar esta labor.

El manejo del rendimiento es otro problema debido a que un sistema cliente/servidor maneja más información a través de la red que un mainframe, ya que éste sólo envía o recibe mensajes o imágenes de pantalla que no varían mucho en tamaño, lo cual permite optimizar la red para manejar estos volúmenes de tráfico de datos. En un sistema cliente/servidor cada transmisión puede variar en el número y tipo de peticiones que se realizan al servidor. El tráfico en la red también varía en función de la herramienta utilizada para el desarrollo de las aplicaciones y del software utilizado para las funciones de conectividad entre clientes y servidores. Los patrones de tráfico de la red se vuelven mucho más complejos y difíciles de controlar.

La escalabilidad del servidor puede ser una restricción. Este es un problema común de los sistemas cliente/servidor que requieren más de 1000-1500 clientes. Cada cliente requiere por sesión un proceso o thread en el servidor y, en algunos casos, al menos 1.5MB de memoria no compartida. El punto de corte en cuanto al número soportado de usuarios concurrentes es la incapacidad del servidor para crear más procesos o bien cuando se llega al límite de memoria disponible.

Ambientes híbridos mainframe y cliente/servidor

La mayoría de los sistemas basados en mainframe fueron desarrollados a lo largo de mucho tiempo, dando como resultado aplicaciones con una cantidad considerable de funcionalidad pero implementados en forma monolítica. Típicamente estas aplicaciones fueron escritas como un gran trozo de código y no divididas en módulos predefinidos con interfaces documentadas. Para complicar el escenario, en algunos casos el código fuente ya no existe. Una combinación de esta situación y una mentalidad de "si funciona para que arreglarlo" hace que mucha gente decida no tocar los sistemas existentes y quiera desarrollar sistemas para satisfacer los nuevos requerimientos utilizando plataformas cliente/servidor. La conectividad entre la plataforma mainframe y sistemas abiertos ha

mejorado considerablemente en los últimos años. En muchos casos el nuevo software tipo cliente/servidor debe tener interfaces batch hacia aplicaciones existentes en plataformas mainframe. Esto ha ocasionado que aumente la demanda del poder de procesamiento en el mainframe conforme más transacciones son solicitadas al sistema. Esta estrategia proporciona una oportunidad de realizar una transición lenta del mainframe a otras plataformas sin sacrificar la confiabilidad de los sistemas existentes. Hoy día se abre la posibilidad de adoptar una estrategia similar utilizando tecnología basada en la Web permitiendo una rápida evolución.

7.3 Cambio de paradigma.

La Web se ha convertido en uno de los cambios más interesantes de esta década no sólo por el aspecto tecnológico sino por el impacto que está teniendo o que tendrá en la mayor parte de las organizaciones. Sin embargo, a pesar de las ventajas que presenta este esquema en comparación con los modelos anteriores, todavía existen varios aspectos que deberán ser solucionados para que pueda convertirse en una opción viable en la implementación de aplicaciones de misión crítica. A continuación se muestran los principales factores que han frenado la implantación masiva del modelo cliente/servidor así como las principales ventajas de la Web con éste último.

Problemas del modelo Cliente/Servidor

La gran expectativa que se tuvo alrededor de la arquitectura cliente/servidor en los últimos cuatro años y las experiencias en la implantación de sistemas basados en este modelo han dado como resultado interesantes conclusiones.

Realmente no existe una definición comúnmente aceptada para cliente/servidor a pesar de que el mercado está lleno de herramientas, bases de datos, redes y aplicaciones que hacen referencia a esta arquitectura, la cual generalmente se asocia con cómputo distribuido y un front-end típicamente basado en una plataforma Windows.

Con esta definición mínima de cliente/servidor se pueden hacer las siguientes afirmaciones en relación a los problemas creados por este modelo:

1. Cliente/Servidor resulta costoso en comparación con sistemas centralizados basados en terminales "tontas".
2. La solidez de este esquema es relativa debido a la existencia de múltiples proveedores de tecnología.
3. El rendimiento es problemático debido al ancho de banda de las redes.
4. La administración del sistema y la seguridad son mucho más difíciles de manejar.
5. Las pruebas del sistema tienen que incluir plataformas y configuraciones variables.
6. Aspectos políticos: En muchas ocasiones no está claramente definido quien es el responsable del cliente.

7. El desarrollo de sistemas en ambientes GUI basados en eventos es mucho más complicado.
8. La tecnología cliente/servidor generalmente no ha madurado y en ocasiones no cuenta con capacidades análogas a productos para administración disponibles en otras plataformas.

Costos Ocultos

Desafortunadamente muchos proveedores de tecnología han promovido la percepción de que cliente/servidor es una alternativa mucho más económica comparada con sistemas basados en mainframe. La idea de que se ahorra dinero debido a que las aplicaciones corren en servidores y PCs de bajo costo es falsa, simplemente por el hecho de que el hardware sólo representa un pequeño porcentaje del costo total de una implementación cliente/servidor. Las investigaciones han revelado que el hardware representa alrededor del 20% del costo asociado a un sistema cliente/servidor mientras que el 80% restante aplica al soporte, software y mantenimiento.

Aunque cliente/servidor proporciona algunas ventajas, resulta costoso y en muchos casos no reduce el presupuesto global de tecnología informática en una empresa. El verdadero costo sólo puede ser calculado cuando han sido identificados todos los gastos de operación e implementación. Comparado con ambientes centralizados, una gran proporción del presupuesto asignado a un proyecto cliente/servidor es para capacitación y soporte que para hardware y software. Aproximadamente un 70% del presupuesto se destina a mano de obra en contraste con un 30% necesario en ambientes host.

Muchas organizaciones han emprendido proyectos cliente/servidor con la creencia de que una vez adquirido el hardware y software adecuados la implementación será sencilla y sin mucho esfuerzo. Este es uno de los aspectos menos comprendidos de la tecnología cliente/servidor y tiene serias implicaciones en como las personas justifican proyectos y asignan presupuestos. Simplemente se puede tomar como ejemplo un software como Windows 95; suponiendo un costo de \$90 dólares, el hecho de migrar a 1000 usuarios y sus aplicaciones a esta versión de sistema operativo puede llegar a costar más de dos millones de dólares, en donde la mayoría serán costos asociados a mano de obra.

En muchos casos se opta por una estrategia cautelosa al comprar un pequeño servidor y una herramienta para desarrollo GUI. Esto puede funcionar para pequeñas aplicaciones cliente/servidor a nivel

departamental, pero es demasiado arriesgado para aplicaciones más grandes o críticas para el negocio. También existen muchos costos ocultos que no pueden ser anticipados. A continuación se muestran algunos ejemplos:

- **Limitaciones de las redes de área local**

El costo de sobrepasar los límites de una LAN debe ser considerado antes de usar la red para cualquier operación crítica o de toma de decisiones. No sólo la administración del sistema es mucho más laboriosa, sino que muchas de las primeras herramientas creadas para este fin no son robustas. En una LAN no se tienen las capacidades de automatización de procesos que son algo común en ambientes mainframe. Los administradores locales (costo adicional) deben respaldar y recuperar el sistema en caso de falla, llevar el control de las licencias de software así como realizar actualizaciones de éste.

- **Interconectividad de LANs**

Un proyecto cliente/servidor grande comúnmente involucra varias LANs. El costo de integrar ambientes heterogéneos con diferentes DBMSs, redes, sistemas operativos y aplicaciones ha tomado por sorpresa a muchas organizaciones. El papel de ésta se convierte en el de un integrador de sistemas; conectar diversos componentes y mantener esta conexión es complicado y consume tiempo. Frecuentemente las empresas subestiman el tiempo y recursos necesarios para resolver problemas típicos de integración de ambientes. Adicionalmente, características de cliente/servidor como replicación de datos o two phase commit incrementan el tráfico en la red, lo cual implica actualizaciones costosas a ésta.

- **Distribución de software**

Otro costo oculto se presenta con el problema de distribuir software desde un punto central o de coordinación hacia todos los usuarios en la red. La distribución automática de software es la respuesta, aunque en la actualidad presenta problemas de costo y dificultad de mantenimiento.

- **Usuarios finales**

Las organizaciones también deben tener en cuenta el tiempo que invierten los usuarios en corregir problemas con sus PC como son la configuración de impresoras o bien problemas con el sistema operativo. Este fenómeno es uno de las principales causas de gasto de recursos en una empresa.

- **Pruebas de las aplicaciones**

La nueva generación de herramientas para desarrollo gráficas y basadas en objetos definitivamente incrementará la productividad. Sin embargo, un aspecto que frecuentemente es subestimado es el costo de las pruebas. Realizar pruebas de aplicaciones gráficas y basadas en eventos puede llegar a consumir tres veces más tiempo y esfuerzo que si se probaran en un esquema tradicional en base a procesos. Este costo impacta y opaca las ganancias obtenidas en productividad.

- **Capacitación**

Adicionalmente al incremento en el costo de operación y soporte, cliente/servidor requiere que la mayoría del personal de sistemas se capacite ampliamente en nuevos conceptos, procesos y herramientas. Se estima un costo promedio de 30 mil a 60 mil dólares (incluyendo el costo de la productividad perdida) para capacitar a un administrador de sistemas en el manejo y control de una infraestructura cliente/servidor compleja. Los usuarios finales también necesitan mucha capacitación ya que la implantación de un ambiente cliente/servidor lleva consigo la introducción de nuevas aplicaciones, típicamente con una interface gráfica. Sin esta capacitación los usuarios finales no obtendrán el máximo beneficio de las aplicaciones y los datos que tendrán a su disposición.

Las implementaciones cliente/servidor son más costosas de lo que parecen a primera vista, por lo cual la reducción de costos no deberá ser el motivador para iniciar un proyecto de este tipo.

La administración de un ambiente cliente/servidor es una tarea compleja que se debe atacar desde varios puntos. El desarrollo de sistemas cliente/servidor utilizando técnicas tradicionales orientadas a mainframe estará destinada a fallar. Muchas empresas han comenzado proyectos de este tipo sin tener plena conciencia del ciclo de vida completo de un sistema cliente/servidor, desde el análisis y diseño hasta la

implementación distribuida y el mantenimiento. Existe una necesidad creciente de metodologías y herramientas flexibles para el manejo de este tipo de proyectos.

El valor real de cliente/servidor es el permitir crear una nueva generación de aplicaciones como datawarehousing y workflow. El problema ha sido que en muchos casos cliente/servidor ha sido visto como la forma de utilizar nueva tecnología en la creación de aplicaciones viejas.

Con el surgimiento de lo que hoy conocemos como Web, las posibilidades para la creación de sistemas distribuidos se incrementan dramáticamente y de alguna forma se puede ver como una evolución o redefinición del modelo cliente/servidor.

Web

En los últimos años han aparecido gran cantidad de herramientas y tecnologías para Internet que son los browsers, servidores web, el lenguaje HTML, Java y el protocolo HTTP. Las organizaciones están descubriendo que este mismo conjunto de tecnologías pueden ser utilizadas para crear poderosos sistemas de información y colaboración en la forma de Intranets.

Los servidores web pueden integrarse con bases de datos corporativas, servidores de archivos y repositorios de documentos. Esto proporciona a los usuarios una gran cantidad de información utilizando una interface única en forma de un browser web y que, en muchos casos, puede ser utilizado sin costo.

La capacidad de las Intranets para ser conectadas directamente a tecnología ethernet, FDDI o ATM las convierte en un ambiente ideal para aplicaciones multimedia que generalmente requieren gran ancho de banda, espacio y rendimiento.

Los componentes clave para la instalación de una Intranet son:

- **Browser o visualizador Web.**

Un browser es software que traduce HTML a un formato atractivo y entendible por el usuario. Los dos browsers predominantes en el mercado son el Navigator de Netscape y el Internet Explorer de Microsoft. Los browsers puede desplegar la información como simple texto, gráficos o

bien ligas de hipertexto. Mediante las últimas innovaciones en esta tecnología se pueden representar objetos más sofisticados como sería el caso de las aplicaciones de realidad virtual. La liga de hipertexto permite al usuario ver un nuevo documento o página web almacenado en cualquier servidor de Internet, o bien saltar a un diferente lugar o posición del documento que se está desplegando.

- **Servidor Web**

La función de un servidor web es la de atender peticiones enviadas desde browsers que requieren determinado documento. El servidor toma el documento del disco y lo manda al browser utilizando el protocolo HTTP. También realiza otras funciones como es la validación de peticiones de documentos con seguridad y que sólo pueden ser utilizados por usuarios autorizados.

- **Herramientas de Authoring**

El primer paso para el desarrollo de una Intranet es la colocación de documentos HTML en el servidor web. La opción más viable para la creación de éstos es la utilización de editores HTML, los cuales evitan el tener que conocer el lenguaje y escribir todas las sentencias directamente.

- **TCP/IP**

Los datos son transmitidos a través de la redes utilizando el protocolo TCP/IP. Cada computadora conectada a la red cuenta con una dirección IP que la identifica. Todos los paquetes de información que tengan una dirección IP en particular llegarán a la computadora que tenga ese IP en la red. Cuando el paquete es enviado a otra Intranet éste es ruteado mediante dispositivos llamados ruteadores, que son quienes reciben cada paquete de información y deciden a donde enviarlo en base a tablas de ruteo. Si parte de la red no está disponible u ocupada, el ruteador puede direccionar paquetes de información utilizando distintos caminos hasta el destino.

- **Firewalls**

Las organizaciones que actualmente están utilizando una red corporativa eventualmente desearán conectarse a Internet y de manera selectiva permitir el acceso desde Internet a la red corporativa. Existe confusión y temor al respecto debido a que es realmente difícil tener a la red libre de virus o intrusos. El adoptar una estrategia en donde todo el tráfico es ruteado hacia un proceso de validación antes de entrar o salir de la Intranet es una solución simple y efectiva. Sin embargo en algunas ocasiones no es algo viable debido a que eventualmente puede convertirse en un cuello de botella.

Es claro que un simple programa de validación no es suficiente. El software cuya tarea es decidir quien tiene o no acceso debe tener cierto grado de inteligencia si se requiere que funcione de manera efectiva y segura. De esta forma, el asunto de seguridad requiere una estrategia bien planeada para garantizar la operación deseada.

El método más común y efectivo de proteger a una red es mediante un firewall, que es un conjunto de componentes colocados entre dos redes y que deben cumplir con las siguientes propiedades:

- Todo el tráfico de adentro hacia afuera y de afuera hacia adentro debe pasar por el firewall.
- Sólo el tráfico autorizado, definido por una política de seguridad, podrá pasar.
- El firewall por si mismo debe ser inmune a penetración.

Un firewall se diseña para reducir el área de riesgo para un sistema o bien un ambiente multi-sistemas. Es un sistema que bloquea y checa el tráfico entrante a la red permitiendo sólo a los usuarios autorizados el acceso y transmisión de información privilegiada.

Existen dos tipos de implementaciones de firewalls:

- Los que permiten todo lo que expresamente no está prohibido
- Los que prohíben todo lo que expresamente no está permitido

Pareciera que la etapa de las redes de área local, e inclusive de las WAN junto con cliente/servidor, ha pasado. Actualmente las redes LAN ya no son tan locales; en cualquier compañía de tamaño mediano a grande la red generalmente está ligada a otras y a la Web, por lo que puede ser accesada desde cualquier parte. A pesar de que el concepto cliente/servidor prometía mucho como teoría, en la práctica no era fácil de implementar y por lo tanto no fue tan adoptado como se esperaba.

Existe una posibilidad de solución en forma de Intranets, en donde se utilizan los conceptos de los estándares Internet para crear aplicaciones abiertas. Los principios básicos para la construcción de intranets son muy similares a los de cliente/servidor excepto por el hecho de que la función y la carga de proceso en el cliente se reduce significativamente. Al igual que el paradigma de cliente/servidor, los datos se almacenan en forma centralizada utilizando bases de datos mientras que el control de las aplicaciones y de los ambientes operativos se realiza en el servidor.

La transición y movimiento de un ambiente basado en el cliente a un ambiente de desarrollo basado en la red está cobrando fuerza. Una nueva generación de aplicaciones centralizadas en la red está siendo desarrollada utilizando las siguientes tecnologías:

- HTML (HyperText Markup Language).
- Java y JavaScript
- Internet Foundation Classes
- IIOP (Internet Interoperability Protocol)
- Protocolos de comunicación como HTTP; NNTP; SMTP; IMAP4 y POP3.
- SSL (Secure Sockets Layer)

Intranets

Internet, en particular la World Wide Web, ha sido denominada como la aplicación de red ideal. Sin embargo, mientras Internet está siendo utilizada principalmente como un medio para marketing y ventas, las Intranets permiten a las empresas distribuir información para uso interno con la protección de tecnología firewall.

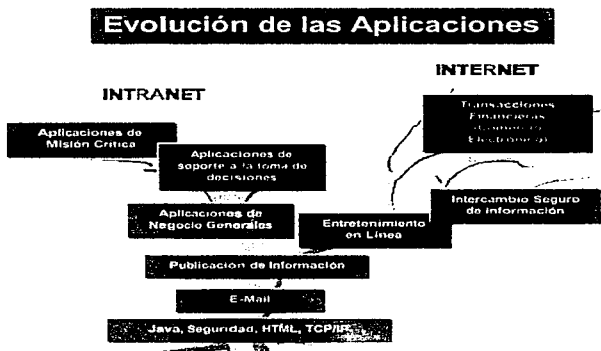
Las Intranets permiten la distribución de información y software de una manera sencilla. Son baratas, fáciles de usar y la protección que proporcionan los firewalls es suficiente para la mayoría de las aplicaciones.

Hace un par de años no existían Intranets, sin embargo se estima que el mercado potencial a nivel mundial excede los 450 millones de dólares, mientras que el negocio alrededor de los servidores para Intranet puede llegar a los 8 billones. El número de sitios web internos actualmente sobrepasa a los externos aproximadamente en una proporción de 10 a 1.

Uno de los mayores éxitos de las Intranets ha sido la capacidad de proporcionar un mecanismo de organizar y guardar información de manera que se pueda utilizar en el momento en que se necesite de forma sencilla y flexible.

Actualmente el acceso a documentos es uno de los usos principales de las Intranets. Se están desarrollando aplicaciones interactivas o dinámicas que permitirán el acceso a datos y aplicaciones existentes, lo cual ampliará las posibilidades de utilización de una manera nunca antes vista.

La siguiente figura muestra el crecimiento con respecto a utilización así como las principales diferencias entre Internet e Intranet:



Las Intranets permiten presentar información de la misma forma en plataformas diferentes combinando distintas aplicaciones y datos, logrando la integración de ambientes heterogéneos en un sólo sistema que proporciona la facilidad de utilizar información sin importar donde resida ésta, algo largamente prometido por proveedores de hardware y software pero que nunca fue completamente cumplido.

Web vs. Cliente/servidor

El Web puede ser utilizado como un servidor centralizado de aplicaciones y datos en donde la estación de trabajo remota simplemente proporciona los servicios de presentación. Todos los problemas de distribución de software, control de versiones y administración del sistema desaparecen sin perder la importante característica de contar con una interface intuitiva. Los clientes remotos pueden acceder al servidor mediante una red privada o bien utilizando Internet con una conexión tipo SLIP o PPP.

La administración del crecimiento llega a ser más sencilla si pensamos que los servidores HTTP cuentan con la escalabilidad necesaria para satisfacer los requerimientos de aplicaciones de misión crítica.

A pesar de que la plataforma Web puede ser una alternativa interesante comparada con la implementación cliente/servidor tradicional, todavía necesita evolucionar para ser el medio ideal para aplicaciones que manejan un alto volumen de datos. La disponibilidad continua no es todavía una de sus características; la tecnología Internet permite el ruteo alternativo del tráfico, pero el concepto de recuperación de transacciones después de una falla del servidor HTTP no existe. Alguna implementación semejante a un monitor de transacciones se requiere para este ambiente, similar a la mayoría de los sistemas actuales en donde existe la posibilidad de realizar commit a las transacciones o regresarlas a su estado original sin alterar el estado de los datos a los que hacen referencia.

La predicción o estimación en cuanto al número de transacciones es impredecible debido a la naturaleza no determinística del Web. En ocasiones se encuentran inmediatamente los URLs; a veces es necesario intentar repetidamente. Es común sufrir alteraciones en el tiempo de respuesta de la red ya que surgen fallas o se incrementa el tráfico en ésta, lo cual puede resultar inaceptable para cierto tipo de aplicaciones.

La seguridad de las transacciones es una preocupación en estos ambientes. Las violaciones a los sistemas se han convertido en algo muy

publicado por lo cual es comprensible que exista cierto nerviosismo al respecto, sobre todo si se trata de aplicaciones comerciales. La falta de estándares tampoco promueve la creación de aplicaciones de misión crítica. En los Estados Unidos es posible la encriptación utilizando 128 bits, pero actualmente esta tecnología no se puede exportar.

La tecnología Web proporciona una perspectiva muy interesante en cuanto al potencial de utilización de ésta. Todavía existen factores comerciales como es la lucha entre los principales proveedores de software y de hardware por lograr posicionarse en el terreno de los estándares de jure y de facto. Sin embargo el ciclo de innovación alrededor de estas tecnologías es mucho más dinámico, lo cual permitirá que en pocos meses ésta llegue a un nivel de maduración y consolidación aceptable.

Existe otro factor sumamente importante (y frecuentemente subestimado) en la implantación de nueva tecnología que es la cultura existente en cuanto al uso de ésta. En ocasiones esto puede significar el fracaso de un proyecto sin que importe que tan sólido sea en cuanto a tecnología.

Debido a que la proyección que tiene la tecnología Web no es sólo hacia un grupo reducido o privilegiado de personas sino que al contrario, está enfocada para alcanzar al máximo número de individuos u organizaciones que sea posible, todo parece indicar que el período de aceptación no será tan largo como con otras tecnologías. Sin embargo también hay que tener en cuenta que la tecnología Web no es la panacea para la solución de los problemas de creación de aplicaciones y que no todas requieren este tipo de implementación, por lo cual hay que cuidar el exceso de optimismo o de promesas alrededor de esta tecnología.

Data warehousing

La tecnología de la información (IT, por sus siglas en inglés) ha sido sinónimo de automatización en muchas organizaciones. En los últimos treinta años se han automatizado procesos de negocio de tal forma que éstos puedan ser ejecutados de manera más eficiente y rápida. Existe una función aún no tan desarrollada y explotada de la IT fuera del ámbito de automatización de procesos de negocio. Este papel es el de proveer información.

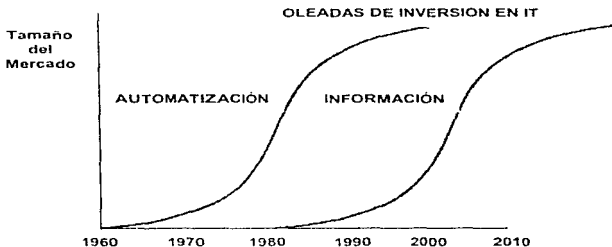
Paradójicamente muchos usuarios de la IT piensan que actualmente están utilizando sistemas de información. Con la proliferación de equipos PC muchos usuarios son capaces de manipular grandes cantidades de datos

y realizar cálculos complejos sobre éstos. Sin embargo en muchos casos este tipo de manipulación de datos no es efectiva o bien es incompleta.

En este contexto la automatización tiene que ver con la eficiencia de realizar tareas repetitivas y que se pueden predecir. No es capaz de indicar por sí misma si determinado proceso es adecuado y si éste puede evolucionar.

La información se encuentra un nivel arriba de la automatización y se deriva fundamentalmente de una acción de comparación. Por ejemplo, el generar un reporte mensual de ventas es una tarea automatizada a menos que se incluya algún proceso comparativo (comparar el reporte actual contra el pronosticado). La información siempre implica una acción.

Con la automatización la computadora es vista como una calculadora gigante. Al aplicarla a la información la computadora se convierte en una máquina lógica. Esta área no se ha comprendido del todo y es por eso que hasta la fecha no se ha explotado el potencial de la IT con una orientación hacia la información de negocios. Parece ser que data warehousing y las tecnologías asociadas pueden cambiar esto.



La mayoría de las organizaciones se han dado cuenta de los beneficios derivados de la automatización, por lo cual ahora la atención y esfuerzos se empiezan a concentrar en otras formas de obtener una ventaja competitiva. Data warehousing representa el primer movimiento hacia el aprovechamiento de la información.

Hasta ahora la justificación de inversión en proyectos de IT ha sido más o menos directa debido a está orientada principalmente al ahorro en costos. Por el contrario, desde el punto de vista de data warehousing no existen fórmulas o mecanismos directos que permitan calcular una reducción de costos, lo cual es quizás uno de los principales problemas para las empresas que desean utilizar esta tecnología.

La tecnología necesaria para un proyecto data warehousing varía en función de las fuentes de datos involucradas y de la complejidad del análisis de información requerido, ya que se pueden utilizar desde simples hojas de cálculo hasta sofisticadas herramientas de data mining.

Una data warehouse requiere la combinación de diversas tecnologías dentro de una estructura. Contiene elementos de:

- **Hardware** - el servidor, ya sea MPP/SMP o uno de gran capacidad.
- **DBMS - RDBMS** tradicionales optimizados o no para consultas.
- **Middleware** - Es el "pegamento" que une a las bases de datos para crear una vista única de los datos corporativos.
- **Herramientas para usuarios finales** - Data mining, DSS utilizando conceptos de inteligencia artificial, sistemas expertos, redes neuronales o alguna técnica estándar de análisis.
- **Herramientas para administración del warehouse** - organizan y reorganizan los datos, crean nuevas vistas, crean o borran tablas sumarizadas, etc.
- **Servicios** - proporcionados por los proveedores para ayudar en la construcción e implementación incremental del warehouse desde la fase piloto hasta la implantación a nivel corporativo.

Data warehousing está ligado con el esquema cliente/servidor, incluyendo la integración de tecnología web, por lo cual se debe ver como un recurso dentro de la infraestructura con una interface que reside en el middleware. Los beneficios que se derivan de una estrategia data warehousing bien diseñada e implementada son:

- Mayor exactitud mediante la eliminación de redundancias.
- Incremento en la eficiencia debido al acceso oportuno a la información.
- Toma de decisiones más efectiva.

Es importante resaltar que la tecnología juega un papel importante en un proyecto data warehouse al permitir el manejo y visualización de la información de una manera distinta y enfocada al análisis de ésta. Sin embargo, es indispensable adoptar una metodología especializada para este tipo de proyectos de tal forma que se pueda implementar en forma ordenada y estructurada, de lo contrario estará destinado a fracasar.

Tecnología de Objetos

Indudablemente la tecnología de objetos jugará un papel importante en la forma en que las empresas utilizan la tecnología informática. Los objetos prometen mayor reutilización y extensibilidad; en un ambiente ideal basado en objetos los desarrolladores serán capaces de resolver un problema de negocio simplemente ensamblando o extendiendo las capacidades de componentes de software reutilizables. Sin embargo los objetos por sí mismos no ofrecen una solución completa. Deben residir en un ambiente abierto y distribuido que permita su utilización en un esquema tipo "plug and play".

El poder real de la tecnología de objetos es la posibilidad de crear aplicaciones que sean modulares, flexibles, escalables y que puedan incluir tecnología existente y futura. En vez de crear aplicaciones monolíticas con una gran cantidad de interdependencias, los objetos permiten escribir aplicaciones en forma de pequeños componentes que pueden ser ensamblados. De esta manera es posible la creación de aplicaciones grandes y complejas que puedan utilizarse a lo largo de la empresa bajo un enfoque modular.

Inicialmente se vio la aplicación de esta tecnología en las herramientas para programación GUI, sin embargo existe un gran mercado para tecnología de objetos por parte de compañías que están creando aplicaciones sofisticadas para aplicaciones críticas de negocio. La tecnología existente basada en programación procedural ha llegado a su límite en muchos casos, por lo que las empresas comienzan a aceptar a los objetos como una forma de reconstruir sus sistemas.

7.4 Tendencias en software

Como se mencionó anteriormente, la tecnología de objetos comienza a convertirse en un aspecto clave para el desarrollo de aplicaciones críticas debido a características claves como la reutilización del código.

La reutilización del software es una de las estrategias más discutidas pero menos implementadas para mejorar la calidad y productividad de los sistemas de una organización. Los conceptos son muy simples: No reinventar lo que ya ha sido inventado; construir nuevos sistemas y mejorar los existentes utilizando bloques o componentes ya probados.

Componentes

El desarrollo en base a componentes o "componentware" es la parte no académica del movimiento de orientación a objetos, en donde típicamente se involucra una estrategia de desarrollo de abajo hacia arriba.

Los conceptos de orientación a objetos como la herencia múltiple y el polimorfismo en muchas ocasiones no han tenido gran seguimiento por parte de los desarrolladores, quienes más bien lo que desean y esperan al emplear tecnología OO es reutilización. El "componentware" es la implementación en el ámbito comercial de este concepto y sus beneficios asociados, como es el incremento de la productividad de los desarrolladores, incremento en la calidad del software, mejora en las capacidades de mantenimiento así como alto grado de adaptación. Todo esto relacionado con la capacidad de ensamblar aplicaciones en base a software reutilizable.

Se estima que aproximadamente un 75% de las organizaciones que desarrollan aplicaciones operan al nivel uno del modelo SPCMM (Software Process Capability Maturity Model) creado por el instituto de ingeniería de software de Carnegie Mellon. En el nivel uno la organización no cuenta con una metodología de desarrollo o procedimientos formales entre proyectos, por lo que el presupuesto se realiza por proyecto.

Existen conceptos erróneos alrededor de la reutilización del software. En muchos casos se plantea la premisa de que es un asunto técnico e inevitable una vez que se resuelven asuntos como la plataforma a utilizarse, la selección de herramientas y la integración de sistemas. La reutilización sólo sucederá si se fomenta la actitud necesaria para lograrlo.

Debido a que la metodología apalanca a la arquitectura, es indispensable adoptarla, desarrollarla e implementarla para que los proyectos de reutilización sean exitosos. Gartner Group estima que el uso consistente de una metodología de reutilización incrementa la productividad en el desarrollo de aplicaciones hasta en un 30%.

Se debe tener en cuenta que adicionalmente a la selección adecuada de tecnologías y metodologías es necesario vencer barreras sociales y culturales dentro de la organización, así como sincronización y comunicación entre equipos de trabajo.

Actualmente existen dos estándares para la implementación de aplicaciones a partir de componentes. Uno de ellos es CORBA (Common Object Request Broker Architecture) auspiciado por el OMG (Object Management Group) y ActiveX, promovido por Microsoft y basado en la arquitectura DCOM (Distributed Component Object Model) desarrollada junto con Software AG. Existe mucha polémica en relación a estas dos estrategias; los puristas se inclinan a favor de CORBA argumentando que cumple con los cánones que marca la teoría de orientación a objetos y haciendo énfasis en la naturaleza no propietaria de CORBA. Sin embargo ActiveX representa el enfoque práctico que ha caracterizado a Microsoft, inclusive el manejo y definición futura de esta tecnología estará coordinada por el Open Group con la intención de que rápidamente se convierta en un estándar y una alternativa viable para la creación de aplicaciones distribuidas basadas en componentes.

Como resultado las organizaciones ya no estarán restringidas a adquirir aplicaciones (que pueden no cubrir todas sus necesidades de negocio) o bien a desarrollar sistemas desde cero. Para muchas empresas la opción de ensamblar aplicaciones en base a componentes de software que han sido probados, pueden ser modificables y reutilizables ofrece la solución óptima para sus necesidades de sistemas. Independientemente de que estándar y tecnología se utilice para la creación de estas aplicaciones, es claro que el papel del repositorio para efectos de control y manipulación de los componentes jugará un papel clave.

La capacidad de reutilizar objetos requiere un repositorio sólido y extensible. La mayoría de los proveedores de herramientas han identificado esta necesidad y ofrecen algún tipo de solución dentro de su catálogo de productos, incluyendo herramientas para la administración del repositorio. Adicionalmente al control de versiones y funcionalidad para la administración de la configuración, se requieren herramientas

para organizar y catalogar componentes, de lo contrario la verdadera reutilización estará todavía lejos de implementarse.

Java

Actualmente Java es una de las tecnologías dominantes en el mundo Web. Según datos de JavaSoft y varias organizaciones de investigación de mercado, se calcula que el número de desarrolladores Java en la actualidad es de 200,000 con una expectativa de crecimiento a 1 millón en los próximos dos o tres años.

En 1990 Sun Microsystems crea un proyecto llamado Green con la idea de desarrollar sistemas avanzados para diversos dispositivos de red. En esos años, el lenguaje C++ y las metodologías de objetos tenían auge entre la comunidad de investigadores.

Después de intentar adoptar a C++ para este proyecto se decide crear uno nuevo que se basara en las mismas ideas y conceptos del mundo de objetos y lenguajes como Smalltalk, Eiffel, Objective-C, Cedar/Mesa y el mismo C++, pero con un enfoque hacia redes y el desarrollo de aplicaciones distribuidas.

Surge la idea de un lenguaje y su ambiente de desarrollo en 1991. Se le bautiza inicialmente como Lenguaje Oak y en 1992 se crea el proyecto Star 7*. En 1994 el grupo de desarrollo se enfoca a la tecnología WWW, en específico a generar un browser, el cual sería bautizado como HotJava. En Mayo de 1995 surge oficialmente Java y HotJava.

En la actualidad, Java ha tenido una gran aceptación, por lo que es de esperarse que pronto termine de establecerse como estándar en el mundo Internet. Varias de sus características, como el hecho de ser un lenguaje orientado a objetos, lo hacen ideal para poder crear diversos tipos de aplicaciones e inclusive hardware o chips Java.

Dentro de las aportaciones de Java al entorno de desarrollo se encuentra el hecho de hacer transparente para el desarrollador la interface gráfica así como el sistema operativo, lo cual hace que sea independiente de la arquitectura. El código puede ejecutarse sin cambio alguno en cualquier máquina que cuente con un ambiente Java. Este lenguaje permite ejecutar aplicaciones en forma concurrente ya que soporta el concepto de multithreading, lo cual hace que Java pueda ser utilizado para generar aplicaciones de simulación.

Java es un lenguaje de programación interpretado. Cuenta con un recolector de basura que permite optimizar el uso de la memoria, soporte para ambiente run-time y un conjunto de clases integrado para desarrollos multi-plataforma. Estas características hacen que Java sea un lenguaje adecuado para ambientes heterogéneos.

Las aplicaciones Java en el contexto del WWW han sido denominadas Applets. Actualmente el diseño de la Máquina Virtual Java, soportada por algunos browsers, requiere que el sistema operativo soporte multithreading, por lo cual en ambientes DOS no es posible tener Java al menos como se conoce ahora.

El código Java no se compila en código de máquina normal. Este es traducido a código en formato de máquina normal y posteriormente a código en formato único para una máquina virtual. En la máquina en donde se tiene el sistema Java Virtual Machine, también se tiene un intérprete Java, el cual ejecuta el código. En este esquema no existe la etapa de ligado, ya que el intérprete liga en forma dinámica las clases conforme se requieren.

Sin embargo, como nueva tecnología Java tiene varias características que lo hacen inadecuado para la programación de aplicaciones robustas o de misión crítica. Muchas empresas están conscientes del potencial del lenguaje, sin embargo consideran que todavía no es algo maduro. El desempeño es actualmente uno de los puntos débiles de Java; otro punto en contra es la falta de herramientas para el manejo de los applets, ya que al ejecutarlos desde un browser no es posible controlar el registro y la interpretación de éstos

Una de las aportaciones de Java a la ingeniería de software es la flexibilidad y potencial que proporciona la existencia y disponibilidad de applets, los cuales permiten crear aplicaciones en menos tiempo y con menor inversión, con la ventaja de distribuirse de una manera mucho más sencilla. Al ser un lenguaje orientado a objetos, Java ha permitido aprovechar en forma práctica metodologías para análisis, diseño y construcción orientada a objetos que han evolucionado durante la última década.

JavaScript

La tecnología de los browsers ha permitido el surgimiento de aplicaciones independientes de plataforma y fáciles de implementar, en comparación con el tardado y costoso proceso de instalación y actualización en las

implementaciones cliente/servidor tradicionales. Una página Web puede ser creada para funcionar como un cliente "delgado". Como se trata de una página, la distribución puede hacerse a través de una red interna (Intranet) o de una red externa (Internet). Una aplicación sencilla sería una simple forma para captura de datos. Esta forma podría ser almacenada en un servidor Web en lugar de implementar esta aplicación en una herramienta tradicional como Visual Basic, C o Delphi. La forma podría ser utilizada por cualquier cliente que tuviera un browser. Sin embargo, las limitantes de HTML hacen que una aplicación tan simple como ésta sea algo complicado de programar totalmente en este lenguaje.

Los motores para scripts se diseñaron para resolver el problema de no contar con código de aplicación en el cliente. Como el script está incluido en el código HTML, éste se descarga cada vez que la página es accesada. La actualización de los clientes simplemente requiere un cambio de archivos en el servidor.

La tecnología Web redefine el esquema cliente/servidor ya que permite tener un balance entre lo que anteriormente eran sólo terminales "tontas" del equipo host o bien varios equipos PC conectados en red pero en la mayoría de los casos asilados de los demás ambientes. La tecnología Web permite crear aplicaciones en donde el servidor "descarga" pequeños programas en el cliente, el cual a su vez ejecuta dicho código. Este nuevo paradigma aprovecha las fortalezas de un modelo centralizado, como son el mantenimiento y control. También permite al equipo cliente realizar los procesos más adecuados para estas plataformas como son la verificación de los datos de captura y las funciones relacionadas con la interface de usuario.

La mayor parte de la atención se encuentra alrededor de Java, sin embargo la funcionalidad común y corriente puede ser cubierta por JavaScript o VBScript, a pesar de que una máquina todavía requiere una cantidad considerable de recursos (en plataforma Windows, 16MB de RAM y procesador Pentium como mínimo) para ejecutar una aplicación Java de manera aceptable. Los mecanismos de scripts son una extensión mucho más simple a la tecnología de browsers. Todo indica que Java se establecerá como una nueva plataforma totalmente distribuida, pero los mecanismos de scripts pueden proporcionar la mayoría de las funciones clave necesarias para que una aplicación sea ejecutada en un cliente "delgado".

Los dos mecanismos para scripts disponibles son JavaScript y VBScript. JavaScript no tiene muchas cosas en común con Java, pero al igual que

éste, es una versión simplificada de C++. VBScript está basado en el lenguaje Visual Basic.

Los lenguajes basados en scripts permiten la creación de funciones y su inserción en código HTML, lo cual proporciona lógica estándar como son operaciones matemáticas, condiciones y ciclos. Los dos mecanismos permiten crear HTML en forma dinámica, lo cual significa que un programa puede crear una página HTML en base a la selección del usuario o a los datos capturados, sin que esto requiera un programa en el servidor Web para construir la página necesaria.

Adicionalmente a las capacidades de programación, ambos lenguajes pueden hacer llamadas a recursos fuera de la lógica del programa. Por ejemplo, VBScript puede ejecutar objetos ActiveX o applets Java almacenados en el cliente. JavaScript también permite manipular applets Java. Esta capacidad permite extender las capacidades de la aplicación hacia otras tecnologías al actuar como "pegamento" en un sistema basado en Web.

Entre las ventajas de utilizar applets Java está la disponibilidad en varias plataformas y el poder bajarlos conforme se necesitan. Si un browser no cuenta con un applet utilizado por alguna página en particular, el servidor Web puede automáticamente "descargarlo" al cliente. Esto proporciona la capacidad de distribución dinámica. Sin embargo actualmente los applets Java no se caracterizan por su rendimiento.

El diseño de la máquina virtual Java permite ejecutar un applet Java en cualquier plataforma que cuente con un motor de máquina virtual. Esta característica requiere que cada instrucción pase a través de este mecanismo de middleware para simular su ejecución.

Los lenguajes basados en scripts tienden a ser mucho más rápidos debido a que utilizan código compilado en la plataforma destino listo para ejecución. Esto significa que un comando ya sea de JavaScript o VBScript ejecuta un conjunto de instrucciones compiladas en forma nativa en vez de llamar a una rutina que debe simular el procesamiento de cada instrucción (Java), lo cual implica una degradación en el rendimiento debido a la simulación de un procesador instrucción por instrucción.

A pesar de las desventajas de los componentes plug-in (Plug-ins de Netscape o controles Active X) en ocasiones es necesario utilizarlos para algunas aplicaciones debido a que proporcionan acceso a los recursos nativos del sistema como drivers de video, impresoras y tarjetas de sonido, por ejemplo. En contraste, la misma tecnología que proporciona a

los applets Java una independencia de plataforma evita el poder procesos muy específicos u optimizados.

Las características de cada lenguaje de scripts los hace adecuados o no para cierto tipo de aplicaciones o ambientes. Sin embargo, Netscape Communications Corp. a pesar de haber creado JavaScript, promueve el uso de Java para el desarrollo de aplicaciones, lo cual explica de alguna forma la falta de un ambiente poderoso de desarrollo. Inclusive el soporte y difusión del lenguaje en la Web por parte de Netscape es reducido. Por otro lado, Microsoft está tratando de promover fuertemente su tecnología Active X utilizando a VBScript como uno de los componentes claves en su estrategia.

Problemas con estas tecnologías

Durante el desarrollo de cualquier sistema, uno de los mayores problemas es la existencia de bugs dentro de los propias herramientas de desarrollo. Por la naturaleza dinámica de la Web, los ciclos de creación y liberación para herramientas para desarrollo se ha colapsado en forma dramática, lo cual no siempre es para bien. Cada mes se liberan nuevas versiones con nuevas características y, desafortunadamente en muchos casos, con más bugs.

Aplicaciones

El tipo de aplicaciones que se pueden implementar con estas tecnologías tienen la característica de clientes "ligeros". La captura de datos es una de éstas, ya que las pantallas de datos puede interactuar con el usuario y automáticamente verificar la información capturada y realizar cálculos simples. Esto puede disminuir en gran medida el tráfico en la red, en contraste con realizar estas operaciones en el servidor. Cuando se requiere cambiar el formato de captura de datos, simplemente se modifican los archivos HTML que residen en el servidor y los usuarios inmediatamente están actualizados. Este formato de captura también puede ser utilizado desde varias plataformas sin necesidad de realizar una versión especial para cada una de éstas.

Típicamente las aplicaciones que hacen uso intensivo de pantallas o ventanas no son buenas candidatas para implementarse en el Web, ya que debido a la misma independencia de plataforma en muchos casos es casi imposible predecir el comportamiento exacto de la presentación en la

pantalla y garantizar que una pantalla se verá igual en cualquier sistema, monitor e impresora.

Finalmente, a pesar de los avances en tecnología Web, el acceso es unidireccional. Por ejemplo, utilizando un browser no es posible enviar un archivo gráfico al servidor a menos que se implemente un control especial que realice la transferencia utilizando FTP.

Sin embargo hay que tener en cuenta que Java es un lenguaje que tiene año y medio de creado en comparación con C++ que tardó más de 10 años en evolucionar.

Todo indica que estas tecnologías se estabilizaran durante 1997, aunque siempre es recomendable crear un prototipo para realizar una prueba de concepto ya que de lo contrario un proyecto se puede ver detenido por características todavía no implementadas o fallas en los productos utilizados.

GIS

Analistas de la industria como DataQuest han reportado que se ha incrementado la utilización de tecnología GIS para mejorar el servicio al cliente y ayudar a reducir costos, lo cual ha motivado a que se considere que el mercado para este tipo de sistemas crecerá de \$862 millones dls (1995) a \$1,700 millones en el año 2000. Esto muestra que esta tecnología sigue en crecimiento y se espera atraiga la atención de los principales líderes en bases de datos y herramientas CAD.

Organizaciones como servicios públicos, transporte, marketing, entre otras, que utilizan información con algún contenido geográfico son las que han contribuido al gran crecimiento de la industria de los GIS.

La maduración de la tecnología también ha jugado un papel importante; el software existente ha evolucionado proporcionando mayores capacidades para análisis de la información, facilidad de uso y variedad de aplicaciones en donde puede ser integrado. El abaratamiento del poder de cómputo ha ocasionado que en la actualidad sea posible la manipulación de mapas en tiempo real. La utilización de nuevas tecnologías para recopilar datos geográficos así como la disponibilidad de información resultado de censos también ha tenido un impacto positivo en la implementación de aplicaciones GIS.

Muchas empresas buscan en este tipo de sistemas el contar con herramientas que desplieguen mapas y permitan analizar los datos asociados con el objeto de apoyar en la toma de decisiones, lo cual es el mismo objetivo de proyectos como son los de Datawarehousing.

La tecnología para el manejo de mapas ha evolucionado al aprovechar los conceptos de orientación a objetos así como la administración distribuida de versiones. Actualmente es posible el manejo de datos en formato vector y raster, lo cual evita el tener que digitalizar mapas completos y permite emplear un scanner para capturar en forma electrónica cualquier mapa y manejarlo en formato raster. Sólo se captura la información que se considera crítica en formato de vector para su manipulación.

Futuro

Algunos proveedores de tecnología GIS trabajan actualmente en el desarrollo de extensiones a sus productos para que permitan la implementación de aplicaciones integradas con tecnología Web. Se espera que durante los siguientes 18 meses se extenderá el uso y desarrollo de sistemas GIS que puedan ser utilizados en ambientes Internet o Intranet.

El funcionamiento de estos productos plantea la existencia de componentes servidor en donde se podrá instalar el motor para el manejo de mapas, las aplicaciones GIS y su correspondiente conexión a la base de datos, con lo cual estas aplicaciones podrán ser utilizadas desde cualquier equipo cliente que cuente con un browser. Esto significa una reducción en lo requerimientos de hardware y software en el equipo cliente así como un mejor balance entre la utilización de los recursos y capacidades de los equipos servidores y clientes.

7.5 Tendencias en hardware

Procesamiento en paralelo

Hace algunos años el procesamiento en paralelo era una tecnología relativamente reservada, sin embargo en la actualidad es un término cada vez más utilizado. No sólo está asociado a arquitecturas de multiprocesamiento simétrico (SMP ó Symmetric Multiprocessing), paralelismo masivo de procesadores (MPP ó Massively Parallel Processor) o cluster, sino también incluye a las bases de datos que soportan funcionamiento en paralelo.

SMP vs Clusters vs MPP

Cada vez es más difícil clasificar las tecnologías SMP, Cluster o MPP ya que la división entre cada una de éstas comienza a hacerse difusa. Todas se refieren a decisiones de diseño hechas por el fabricante del hardware en donde se especifica la forma en que múltiples procesadores son incorporados en una sola computadora.

SMP

Tipicamente en una arquitectura SMP la máquina cuenta como máximo con una o dos docenas de procesadores en donde cada uno comparte todos los recursos de hardware del sistema (por ejemplo memoria, discos y bus). Como cada procesador tiene acceso a toda la memoria disponible resulta sencillo hacer que "vea" los resultados generados por otro procesador, por lo cual la comunicación es directa.

Clusters

Generalmente los clusters son el resultado de algunas máquinas SMP ligadas entre sí por una interconexión, que es un mecanismo escalable y con un gran ancho de banda que en forma simultánea rutea mensajes entre nodos de múltiples procesadores. Los mensajes fluyen a través de esta red desde la memoria del que envía hasta la memoria del que recibe, sin intervención del procesador. Un protocolo generado por hardware para verificación de mensajes asegura la entrega confiable de éstos. Una configuración de este tipo permite agrupar el poder de proceso de

múltiples máquinas SMP o nodos como una sólo unidad. El problema en este esquema es que dentro de cada cluster cada nodo tiene su espacio privado de memoria, lo cual hace que la comunicación entre procesadores se convierta en algo más complejo. Un procesador en un nodo no puede ver directamente los resultados generados por otro procesador en otro nodo. Es necesario implementar un mecanismo de comunicación en donde se envían mensajes y datos a través de la interconexión.

MPP

Al soportar múltiples nodos en donde cada uno cuenta con su propia memoria y se comunican mediante un sistema de mensajes a través de una interconexión, los sistemas MPP son conceptualmente similares a los clusters. Sin embargo la principal diferencia radica en que la interconexión es mucho más sofisticada. El ancho de banda de ésta se diseña para poder incrementarse conforme se agregan más nodos a la configuración; también se utilizan esquemas de conexión más avanzados. Como resultado, las plataformas MPP pueden soportar varias decenas de nodos. Sin embargo la coordinación entre todos estos nodos puede resultar compleja de implementar debido al mecanismo de comunicación en base a mensajes utilizado.

Existen varias opciones en lo que respecta a las arquitecturas de hardware para procesamiento en paralelo. A nivel general, SMP es más fácil de manejar que MPP, pero éste último es más escalable. Sin embargo la plataforma ideal dependerá del uso que se requiere. Aunque todavía estas tecnologías están en proceso de maduración e implantación, la siguiente tabla muestra que aplicaciones serían en la vida real las más apropiadas para cada arquitectura:

	SMP	MPP
Aplicación	Misión Crítica u OLAP	Análisis complejo o sistemas DSS grandes
Crecimiento de la aplicación	Lento y continuo. Menos de 20% anual	No predecible. Más del 50% anual
Cantidad de datos	10 a 100 Gb	Más de 200 Gb

Existe poca información sobre los sistemas cluster ya que principalmente se utilizan como un mecanismo para agregar un segundo sistema SMP a

uno existente, en la mayoría de los casos como una configuración con tolerancia a fallas.

En cuanto al futuro del hardware para procesamiento en paralelo es muy probable que los fabricantes de equipo SMP agreguen algunas características tipo MPP, lo cual hará que los fabricantes de hardware MPP tengan una presión y competencia en lo que respecta a precio y funcionalidad, ya que actualmente un cluster formado por equipo SMP es capaz de soportar bases de datos mayores a un terabyte. Sin embargo para aplicaciones que requieran bases de datos de múltiples terabytes es más conveniente una configuración MPP, que puede estar formada por nodos SMP.

Hace un par de años las capacidades de los sistemas en paralelo excedían las necesidades de los usuarios y aplicaciones existentes. En la actualidad la demanda en cuanto a poder realizar distintos tipos de operaciones con gran cantidad de datos, como es el caso de sistemas DataWarehouse y aplicaciones de minería de datos (Data Mining) ocasionalmente llegan a sobrepasar las capacidades del hardware.

Network Computer

Se estima que en una arquitectura de red basada en clientes "delgados" los costos asociados (incluyendo el hardware y software) por estación de trabajo son de \$2500 a \$3000 dls. Comparando estos números con el costo de cada PC, que según el estimado de Gartner Group es de \$11,900 o con el estimado por Intel de \$8000 por puesto de trabajo, la diferencia es notoria.

A pesar de parecer un concepto simple, existen todavía diversas definiciones de lo que será un cliente "delgado" en forma de una NC (Network Computer). La especificación común de la NC plantea un equipo con 4Mb de RAM, sin disco duro y cualquier rango de procesadores desde un Pentium hasta un chip Risc de 32 bits de bajo costo (\$20 dls.) El software de sistema para la NC consiste en un sistema operativo microkernel, una máquina virtual Java para presentar aplicaciones, una capa multimedia para soporte de audio y video, un browser y un sistema de archivos para soportar memoria cache. El almacenamiento fijo se realiza en el servidor. Se estima que el costo de la NC será de alrededor de \$800 dls.

Sin embargo, a pesar de las aparentes ventajas que tiene la NC sobre los equipos PC tradicionales, existen una serie de consideraciones e interrogantes al respecto. Por un lado existe cierta confusión en el mercado por el hecho de que originalmente se había anunciado que el costo de estos dispositivos sería de alrededor de \$500 dls. Actualmente se estima que el costo promedio de la NC será de unos \$1000 dls. Adicionalmente, existen ciertos costos ocultos en forma de servidores, ruteadores, switches y equipo de red asociados a un ambiente basado en NC. El costo de desarrollar aplicaciones que funcionen bajo este ambiente es otro factor que se debe tener presente.

Lo anterior no significa que la NC no sea una opción interesante, sino que definitivamente hay que tomar con cautela las expectativas creadas en el mercado alrededor de este concepto. La NC no sustituirá directamente a las PCs, sino más bien a las "terminales tontas" ya que proporcionará la misma funcionalidad en un esquema basado en Java, aunque eventualmente podrán posicionarse muy bien en nichos donde una PC es demasiado hardware para la operación requerida.

Como respuesta a la expectativa que se tiene de las NC, Microsoft e Intel recientemente anunciaron la especificación de la NetPC bajo un programa llamado "Iniciativa de Cero Administración para Windows" y que automatizará tareas como la instalación de aplicaciones y actualizaciones al sistema. La NetPC sigue siendo un cliente "gordo" comparado con una NC ya que contará con un disco duro, procesador Pentium y 16Mb de memoria como mínimo, sin embargo resulta una opción muy atractiva por el hecho de reducir los gastos de administración, mantener cierto poder de cómputo en el cliente, no depender totalmente de la red y por estar apoyada por fabricantes de PCs como Compaq, HP y Dell. El costo de este equipo será de \$1000 dls. aproximadamente.

La aparición de la NetPC cuestiona aún más el impacto que puedan tener las NC como en un inicio se promovió. Parece claro que no desplazarán a los equipos PCs, sino que más bien estarán confinadas a determinados ambientes de hardware y software.

7.6 Cambios propuestos al sistema

El diseño del sistema se basó en una arquitectura cliente/servidor de acuerdo a la situación tecnológica en ese momento. Por esta razón sufre algunos de los problemas relacionados con esta arquitectura, principalmente los altos requerimientos de hardware en el cliente para el procesamiento y despliegue de los mapas de acuerdo a los estándares de hace algunos años, en donde una PC con procesador 486 DX y 16MB en RAM era un equipo con muchos recursos en comparación con el estándar existente que eran equipos con procesadores 286 o 386 y 2 o 4 MB de RAM. Sin embargo, a pesar de haber sido construida cuando la tecnología cliente/servidor estaba en su máximo apogeo, el software base fue diseñado para evitar al máximo el tráfico en la red ya que ejecuta toda la lógica de los programas en el servidor y sólo realiza el procesamiento gráfico en el cliente. Con los requerimientos actuales de hardware para un sistema operativo como Windows 95 o Windows NT 4.0, la aplicación no tendría ningún problema en lo que a consumo de recursos se refiere.

Si se realizara el sistema de nuevo, definitivamente debería basarse en una infraestructura Web, de tal forma que el costo de implementación por concepto de comunicaciones sea mucho menor en comparación a tener una infraestructura de red dedicada que, además de ser costosa, limita el acceso a un número reducido de usuarios. En contraste, al habilitar esta aplicación al Web prácticamente cualquier persona que tenga acceso al Intranet o a Internet podría aprovechar la información.

Se debe cuidar que la arquitectura y tecnología a utilizarse para este sistema que, aunque no es de misión crítica, debe contemplar factores como performance, confiabilidad, escalabilidad e integración con los datos y sistemas existentes.

Conclusiones

La aplicación construida permitió a la Delegación Tlalpan evaluar las ventajas de contar con un sistema de este tipo y poder determinar cual es la infraestructura de cómputo más apropiada de acuerdo a la implementación que se busca.

Los puntos de más interés para la Delegación Tlalpan fueron:

- Poder ver de manera diferente la información existente.
- Contar con una herramienta capaz de integrarse totalmente con la plataforma de software de la Delegación.
- Tener la capacidad de crear una extensión a las aplicaciones ya desarrolladas dentro de la Delegación.
- Poder utilizar software y aplicaciones en español.
- Posibilidad de estandarizar su uso en las demás Delegaciones.

Se cubrieron los objetivos planteados ya que ahora la Delegación Tlalpan tiene una idea mas concreta del potencial de uso de sistemas GIS y la aplicación de estos para la solución de problemas específicos de la Delegación.

Fue importante utilizar una metodología de desarrollo ya que aunque se requiere mayor esfuerzo en lo que respecta a documentación y análisis, el resultado de seguir un método permitió la creación de un sistema realmente basado en las necesidades del usuario y que en cualquier momento pueden ser ampliado o modificado en forma controlada si las necesidades de información lo requieren.

Este proyecto permitió conocer el potencial que tienen este tipo de herramientas y sistemas, los cuales proporcionan la información que el usuario necesita en formatos que anteriormente estaban reservados para grandes corporaciones u organizaciones militares y que formaban parte de sistemas con un diseño muy específico y poco flexible.

Con la aparición de herramientas como las utilizadas es posible desarrollar aplicaciones tipo GIS a partir de sistemas tradicionales, con lo cual el costo de implementación se reduce enormemente ya que se aprovecha la inversión que se tiene en sistemas anteriores, información y capacitación. Así mismo, no es necesario contar con hardware propietario y excesivamente costoso para tener una aplicación de este tipo.

En la actualidad el esquema cliente/servidor ha evolucionado de tal forma que con el uso del Web y las tecnologías asociadas sería posible extender la funcionalidad del sistema de manera que éste pudiera ser utilizado por un mayor número de personas. El aprovechar las facilidades de tecnologías como Java y DCOM permitiría ampliar las capacidades del sistema, con lo cual se podría manejar la lógica adicional de la aplicación como objetos independientes de la plataforma y del protocolo de comunicaciones.

Bibliografía

Open Systems: A Practical How-To Guide
Byron Vranas Jr.
Software AG, 1994.

CASE*Method
Entity Relationship Modelling
Richard Barker
Addison-Wesley, 1991

Downsizing for Cost-Effective Enterprise Computing
Computer Technology Research Corp. 1993.

Client/Server Technology. The New Direction in Computer Networking
Computer Technology Research Corp. 1993.

4GLs. An Evaluation and Comparison.
Philip Howard & Carl Potter
ButlerBloor Ltd. 1993.

Fundamentos de Bases de Datos
Henry F. Korth y Abraham Silberschatz
Mc. Graw Hill, 1988.

Mainstream Objects: An Analysis and Design Approach for Business
De Yourdon, Katharine Whitehead, Jim Thomann, Karin Oppel & Peter
Nevermann
Prentice Hall, 1995.

Object Oriented Design with Applications.
Grady Booch, 1991

Paradigm Shift: The New Promise of Information Technology.
Don Tapscott & Art Caston
Mc. Graw Hill, 1993.
Ingeniería de Software
Richard Fairley
Mc. Graw Hill, 1988.

Análisis Estructurado de Sistemas
Chris Gane & Trish Sarson
El Ateneo Editorial, 1982.

Natural Engineering Workbench Define V1.1 Concepts and Facilities
Manual
Software AG, 1993

Extended Relational Analysis Workshop V5.1
Relational Systems Corporation, 1985

LAN TIMES Guide to SQL
James R. Groff
Paul N. Weinberg
McGraw-Hill, 1994

Managing for Results
Peter F. Drucker
Harper & Row, 1964

DBMS Magazine
Diciembre 1993

http://nsdi.usgs.gov/nsdi/pages/what_is_gis.html

Fundamentos Cliente/Servidor
Centro Europeo de Soluciones Cliente/Servidor de IBM
Documentos Computerworld
IDG Communications, S.A.

ENTIRE Technology Glossary
Software AG, 1992.
Diccionario de Computación V5.2
Alan Freedman
The Computer Language Company Inc. 1981-1993

Managing Client/Server Development
Management Guide. Strategies and Technologies
Butler Group, Enero 1996.

CIO Magazine
Marzo, 1997

Data Warehouse
Management Guide. Strategies and Technologies
Butler Group, Enero 1996.

Glosario

A

ANSI American National Standards Institute. Instituto Americano de Normas Nacionales. Organización de afiliados privados sin fines de lucro, fundada en 1918, que coordina el desarrollo de normas nacionales voluntarias en Estados Unidos tanto en el sector privado como en el público. Es el miembro internacional de Estados Unidos en International Standards Organization (ISO) (Organización Internacional de Normas) y en la Internacional Electrotechnical Commission (IEC) (Comisión Internacional Electrotécnica).

API Application Programming Interface. Es un punto de entrada específico para cada producto que permite utilizar la funcionalidad de otros programas o bien comunicarse con estos.

APPC Advanced Program-to-Program Communication. Es una comunicación de propósito general entre programas. Originalmente APPC era sinónimo de LU 6.2, protocolo que forma parte de la arquitectura SNA (Systems Network Architecture) de IBM, sin embargo el término es usado actualmente para cualquier forma de comunicación punto a punto entre programas. Esto se debe a que LU6.2 sólo ha sido implementado en pocas plataformas, la mayoría IBM.

ATM Asynchronous Transfer Mode. Modo de Transferencia Asíncrona. Red estándar para transmitir a alta velocidad por medio de fibras ópticas. Utiliza un paquete de 53 bytes de longitud fija para datos.

Azimuthal Es una proyección en un plano que toca la superficie terrestre en un punto. Las posiciones medidas en el mapa a partir del punto de contacto serán correctas, es decir, serán las mismas que las medidas en el globo.

B

Back End Parte de la lógica de una aplicación que generalmente reside en el host. En términos de Cliente/Servidor, al servidor comúnmente se le denomina back end.

Bearing (1) Es un ángulo medido en la dirección de las manecillas del reloj a partir del norte. (2) La dirección horizontal de un punto terrestre a otro, normalmente medido en la dirección de las manecillas del reloj a través de 360 grados.

Bit (Binary digit) Dígito binario. Un dígito simple de un número binario (1 o 0). En la computadora, un bit es físicamente una celda de memoria (constituida por transistores o un transistor y un condensador), un punto magnético en un disco o una cinta, o un pulso de alto o bajo voltaje viajando a través de un circuito. Conceptualmente, un bit se puede pensar como una lámpara eléctrica, encendida o apagada. Grupos de bits forman unidades de almacenamiento en la computadora, llamadas caracteres, bytes o palabras, que son tratados como un grupo. La unidad de almacenamiento más común es byte, constituido por ocho bits y equivalente a un carácter alfanumérico.

Browser También conocido como navegador, es un programa cliente (software) utilizado para ver distintos tipos de recursos en Internet. Estos recursos son proporcionados por el Web en la forma de documentos HTML. El browser es capaz de interpretar estos documentos y presentarlos en un formato entendible para el usuario.

Bug bicho, error. Un error persistente en el software o el hardware. Si existe en software, puede corregirse modificando el programa. Si existe el hardware, deberán diseñarse nuevos circuitos. El término fue acuñado en los años 40 cuando se encontró una polilla aplastada entre los contactos de un relé electromecánico en el Mark I.

Bus ducto, colector. Un canal o ruta común entre dispositivos del hardware, ya sea internamente entre componentes de la computadora o externamente entre estaciones de una red de comunicaciones.

Byte La unidad común de almacenamiento en computación, desde computadoras personales hasta macrocomputadoras. Se compone de ocho dígitos binarios (bits). Puede agregarse un noveno como bit de paridad, para comprobación de errores.

C

CA Computer Associates International, Inc. Un líder entre los fabricantes independientes de software, con más de 200 programas para desde micros hasta macrocomputadoras. Fue fundada en 1976 por Charles Wang y otros tres asociados, y su primer producto fue el CA-SORT, un programa utilitario para macrocomputadora IBM. En el campo de las computadoras personales, la compañía es conocida por el SuperCalc, una de las primeras hojas de cálculo. En 1989, los ingresos de Computer Associates International sobrepasaron los mil millones de dólares, con más de 6500 empleados en más de 100 oficinas por todo el mundo.

CASE Computer Aided Software Engineering. Es un componente de software que puede ser utilizado en el análisis y diseño de sistemas. Existe también el upper CASE, que se refiere a software que permite llenar un diccionario o repositorio mediante una interface gráfica, y lower CASE, que se refiere a la capacidad de generación de programas.

CPU Control Processing Unit. Unidad central de proceso (procesamiento). La parte de una computadora que realiza la computación. También llamada el procesador, está constituida por la unidad de control y la ALU.

D

Data Mining Es una técnica apoyada de herramientas de software orientada al usuario que típicamente no sabe con exactitud lo que está buscando, pero que necesita ver ciertos patrones o tendencias en la información. Involucra un proceso de examinar grandes cantidades de datos para producir relaciones basadas en el contenido de estos.

Data Warehouse Es la implementación de una base de datos para información utilizada para apoyo a la toma de decisiones. Típicamente es una base de datos orientada a un tema y que permite a los usuarios explorar mejor los datos existentes en los sistemas operacionales de acuerdo a las necesidades de negocio.

DBMS Database management system, sistema de administración de bases de datos. Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta pedidos de datos desde un programa de aplicación y le ordena al sistema operativo transferir los datos apropiados.

Debugger Depurador. Un programa que asiste en la depuración de un programa. Proporciona formas de detener un programa o capturar diversos datos de sistema en momentos prescritos. El depurador puede ser capaz de saltar directamente a la línea del programa fuente que originó el problema.

DEC Digital Equipment Corporation. Una marca registrada de la Digital Equipment Corporation que es utilizada en muchos de sus productos; por ejemplo, DECmate, DECnet y DECsystem. En el pasado, todo el mundo se refería a la compañía por esta sigla; no obstante, hoy la compañía incita al uso de Digital en su lugar.

Diccionario dinámico Es un diccionario de datos que puede ser accedido en tiempo real por un programa aplicativo.

DRAM El tipo más común de memoria para computadoras. La arquitectura RAM dinámica (DRAM) emplea habitualmente un transistor y un condensador para representar un bit. Los condensadores deben ser energizados cientos de veces por segundo para mantener las cargas correctas. Nótese la diferencia con static RAM, la cual es habitualmente más rápida y no requiere circuitería de refresco. A diferencia de los chips de firmware (ROM, PROM, etc), ambas variedades de RAM pierden su contenido cuando se corta el suministro de energía.

E

Equidistancia Es la capacidad de preservar las distancias cuando nos referimos a proyecciones.

F

FDI Fiber Distributed Data Interface. Interface de Distribución de Datos de Fibra óptica. Conjunto de normas de ANSI para redes de área local con fibra óptica. Se aplica a las dos capas inferiores del modelo OSI (enlace de datos y física) y transmite a 100 megabits por segundo. A esta velocidad, los gráficos de alta resolución pueden ser transmitidos rápidamente y el video digital puede ser manipulado en tiempo real.

Fire Wall Una combinación de hardware y software que separa una red LAN en varias partes con el objeto de tener seguridad.

Front End Un elemento de hardware o software que actúa como interfaz entre el usuario y el back end. En términos de Cliente/Servidor, el cliente comúnmente se denomina front end.

G

Geodetic Un mapa de este tipo se utiliza para la producción de mapas topográficos, típicamente representaciones a gran escala de pequeñas regiones de la Tierra (estados y países). En vez de meridianos y paralelos se utilizan coordenadas rectangulares en un plano. La base para estos mapas no es una esfera (como se usa en muchas proyecciones en cartografía) sino un elipsoide estándar. La posición al norte en un mapa topográfico no debe ser confundida con el norte geográfico.

Gigabyte Un millón de bytes. También GB, Gbyte y G-byte.

GOSIP Government Open Systems Interconnection Profile. Perfil de Interconexión de Sistemas Abiertos del Gobierno Mandato del gobierno de U.S.A. para que después del 15/8/90, todos los nuevos procedimientos de redex deben ajustarse a OSI. La comprobación se realiza en NIST, que contiene una base de datos de los productos comerciales conformes a OSI. (También se pueden utilizar todavía protocolos TPC/IP).

H

Host Anfitrión. La computadora central o la computadora controladora en un entorno de procesamiento en tiempo compartido o distribuido.

HTML HyperText Markup Language. Es un lenguaje tipo script usado para crear documentos con hipertexto y utilizarse en páginas Web. Los archivos HTML se crean para ser vistos desde un browser o navegador.

I

I-CASE Integrated CASE. CASE integrado. Sistemas CASE que generan código de aplicaciones directamente desde las especificaciones del diseño. Las características incluyen soporte para un prototipo rápido, para modelar los datos y el procesamiento, y capacidad de dibujar diagramas de lógica.

I/O Input/Output, entrada/salida (E/S). Una transferencia de datos entre la CPU y un dispositivo periférico. Toda transferencia es una salida de un dispositivo y una entrada en otro.

IBM International Business Machines Corporation

Ingres Interactive Graphics and Retrieval System. Sistema interactivo gráfico y de recuperación. Un sistema de administración de bases de datos relacionales de Ingres Corporation que se ejecuta en computadoras VAX y estaciones de trabajo UNIX. Incluye un lenguaje de cuarta generación, consultas por ejemplos y un sistema de administración de formularios que permite a los usuarios la creación, edición y visualización de las bases de datos como una serie de formularios.

Su extensión Object Management (administración de objetos), introducida en 1989, permite tipos de datos definidos por el usuario, y su extensión Knowledge Management (administración del conocimiento) permite la programación de una cantidad ilimitada de reglas en las bases de datos.

La Ingres Corporation era conocida anteriormente como Relational Technology, una compañía fundada en 1980 para comercializar una versión comercial de INGRES, que fue desarrollada en University of California al Berkeley, en los inicios de la década del 70.

Interface Una conexión e interacción entre hardware, software y usuario. Las interfaces de hardware son los conectores y cables que transportan las señales eléctricas en un orden prescrito. Las interfaces de software son los lenguajes, códigos y mensajes que utilizan los programas para comunicarse unos con otros, tal como entre un programa de aplicación y el sistema operativo. Las interfaces de usuario son los teclados, ratones, diálogos, lenguajes de comando y menús empleados para la comunicación entre el usuario y la computadora.

Internet Es un gran conjunto de redes interconectadas entre sí que utilizan protocolos TCP/IP y que evolucionaron de lo que fue ARPANET a finales de los '60s y principio de los 70's. Internet conecta miles de redes independientes en forma de una sola red global.

Interoperabilidad La capacidad de dos o más sistemas de software heterogéneos para trabajar juntos (por ejemplo, llamar funciones del otro sistema) en base a un protocolo determinado. Interoperabilidad abarca más que conectividad, un término que está restringido a los componentes físicos. Por ejemplo, en una conversación telefónica, la conectividad es la capacidad de enviar los sonidos a través del cable, mientras interoperabilidad es la capacidad de comprender dichos sonidos.

Intranet Dos o más redes internas (dentro de la misma empresa) o LANs conectadas entre sí. Se conocen también como una red Internet privada.

ISO International Standards Organization - Organización Internacional de Estándares. Una organización que establece estándares (normas) internacionales, fundada en 1946, con sede en Ginebra. Se ocupa de todos los campos, excepto la electricidad y la electrónica, las cuales están ya desde antes bajo la jurisdicción de la IEC (International Electrotechnical Commission - Comisión Electrotécnica Internacional), también radicada en Ginebra. Con respecto a los estándares de procesamiento de la información, la ISO y la IEC crearon recientemente la JTC1 (Joint Technical Committee - Comité Técnico Conjunto) para la tecnología informática.

J

Join En administración de bases de datos relacionales, juntar un archivo con otro sobre la base de alguna condición, creando un tercer archivo con datos de los archivos apareados. Por ejemplo, un archivo de clientes puede ser unido con un archivo de pedidos creando un archivo con registros para todos los clientes que compraron un producto particular.

L

LAN (1) Local Area Network. Red de Área Local. Red de comunicaciones que sirve a usuarios dentro de un área geográficamente limitada. (2) Red de computadoras personales dentro de un área geográficamente confinado que se compone de servidores, estaciones de trabajo, sistemas operativos de redes y un enlace de comunicaciones.

LU6.2 También llamado APPC (Advanced Program-to-Program Communications) Comunicaciones Avanzadas de Programa a Programa Sesión SNA que proporciona comunicaciones entre dos programas de aplicaciones. Cuando se utiliza en un nodo SNA Tipo 2.0 permite que PCs o computadoras de rango medio, ejecutando sus propios programas, se comuniquen con el computador central. Cuando se utiliza en un nodo SNA Tipo 2.1, mantiene comunicaciones par a par entre todas las otras computadoras de nodo Tipo 2.1 sin pasar por la central.

M

Macro (1) En programas de aplicación, una pequeña rutina, o escritura, que automatiza operaciones normalmente activadas seleccionando menús o introduciendo una orden cada vez. Los lenguajes de macros pueden incluir controles de programación comunes, como IF THEN, GOTO y DO WHILE. Para ejecutarlo, al macro se le asigna un comando de tecla que es sustituido por el macro cuando la tecla es presionada. (2) En lenguajes ensambladores, una subrutina prescrita que es llamada en varios lugares del programa. En el momento de ensamblar, las llamadas al macro son sustituidas ya sea por la subrutina completa o por una serie de instrucciones que derivan a la subrutina. El equivalente en lenguajes de alto nivel se llama función.

Mainframe Una computadora grande. A mediados de los años 60, las épocas antiguas de las computadoras, todas las computadoras eran mainframes (literalmente "bastidor principal"), ya que el término se refería al gabinete que contenía la CPU. Aunque mainframe aún significa gabinete principal, usualmente se refiere a un gran sistema de computación y toda la experiencia asociada que va con él.

Megabyte 1.000.000 o 1.048.576 bytes o caracteres. También se escribe MB, Mbyte y M-byte.

MIPS Millon Instructions Per Second. Millón de instrucciones por segundo La velocidad de ejecución de una computadora. Por ejemplo, 0,5 MIPS son 500 000 instrucciones por segundo. Una macrocomputadora grande puede realizar de 10 a 50 MIPS, mientras que un microprocesador barato puede estar en el bando de 0.05 MIPS.

Las tasas MIPS no son uniformes en todas las líneas de proveedores. Algunas medidas son las mezclas de los mejores casos y algunos son promedios. Como resultado, MIPS ha sido interpretado como "Mis Information to Promote Sales" (Desinformación para promover ventas).

La tasa MIPS, que está ligada a la velocidad de reloj de la computadora, es sólo un factor en el rendimiento total. La velocidad y el ancho de banda (8 bits, 16 bits, 32 bits) del bus y los canales, la velocidad de disco y memoria, las técnicas de administración de memoria y el software de sistema también determinan la producción total de un sistema de computación.

Mouse Un objeto semejante a una caja que se usa como un dispositivo apuntador y de dibujo. A medida que se hace rodar sobre el escritorio en cualquier dirección, el cursor o apuntador se mueve correspondientemente sobre la pantalla.

MPP Massive Parallel Processing. Es la estrategia "sin compartir nada" del cómputo en paralelo.

Multimedia La comunicación de información en más de una forma; incluye el uso de texto, audio, gráficos, animación de gráficos y video de pleno movimiento.

O

OO Object Oriented. Orientado a objetos.

Ortographic En términos de proyecciones es cuando líneas de puntos de la superficie terrestre son proyectadas desde un punto en el infinito y son, por lo tanto, perpendiculares a un plano que es tangente a la superficie terrestre, resultando en una proyección azimutal en perspectiva. Si el plano en proyección toca, por ejemplo, a la Tierra en un polo, se preserva la equidistancia a lo largo de los paralelos. Los mapas ortográficos son utilizados comúnmente para mapas de la Luna y el cielo.

OSF Open Software Foundation. Fundación de Software Abierto. Una organización de investigación y desarrollo sin fines de lucro dedicada a desarrollar y distribuir un entorno operativo abierto basado en estándares. Formada en 1988, solicita tecnologías de la industria en general e invita a la participación de los miembros para establecer directivas técnicas para sus esfuerzos de desarrollo.

OSI Open System Interconnection. Interconexión de sistemas abiertos. Un modelo de referencia que fue definido por la ISO (International Standards Organization) como un estándar para las comunicaciones mundiales. Define una estructura por implementación de protocolos en siete estratos o capas. El control es transferido de un estrato al siguiente, comenzando en una estación por el estrato de aplicación, llegando hasta el estrato más bajo, luego por el canal hasta la otra estación y subiendo nuevamente la jerarquía.

Existe una funcionalidad similar en todas las redes de comunicaciones; sin embargo, algunos sistemas no OSI existentes integran a menudo dos o tres capas funcionales en una sola.

La mayoría de los fabricantes han accedido a apoyar el modelo OSI en una forma u otra.

P

Plotter Una impresora gráfica que dibuja imágenes con plumas de tinta. Los trazados requieren datos en formato de gráficos vectoriales, de manera que una imagen se compone de una serie de líneas de punto a punto.

Prototyping La creación de un sistema con carácter de prueba para su verificación y aprobación. Con respecto al desarrollo de sistemas de información, la construcción de prototipos se ha vuelto esencial para clarificar los requerimientos de información. En el enfoque tradicional, las especificaciones funcionales, que son los planos y el diseño del sistema de información, deben concluirse y "congelarse" antes de que el

sistema pueda ser construido. Mientras que la persona con orientación analítica puede tener una imagen clara de los requerimientos de información, puede no suceder lo mismo con otras.

Empleando las capacidades de cuarta generación de un sistema de administración de bases de datos como herramienta de construcción de prototipos, los analistas de sistemas y los usuarios pueden desarrollar el nuevo sistema de forma conjunta. Se pueden crear y manipular las bases de datos de una terminal de macrocomputadora, o en una computadora personal, mientras el usuario asiste al progreso.

Cuando los usuarios ven salidas tangibles en una pantalla de presentación o en un informe impreso, pueden imaginar qué es lo que está faltando o cual sería la próxima pregunta si este fuera un sistema de producción. Si la construcción de prototipos se lleva a cabo cuidadosamente, el resultado final puede ser un sistema en marcha.

Aún si el sistema final debe ser reprogramado en otros lenguajes con motivos de estandarización o de eficiencia de máquina, el prototipo sirvió para proveer especificaciones para un sistema real en lugar de uno teórico.

R

Raster En graficación por computadora, técnica para representar una figura como una matriz de puntos. Es la contrapartida digital del método analógico usado en televisión. Contrástese con vector graphics.

RDBMS relacional Relational DataBase Management System. Sistema de administración de base de datos

Run time Tiempo de ejecución. Se refiere a la ejecución real de un programa.

S

Scanner Dispositivo que lee texto, imágenes y códigos de barras. Los scanners de texto y de código de barras reconocen las letras impresas y los códigos de barras y los convierten en código digital, tal como el ASCII. Los scanners gráficos convierten una imagen impresa en una de video (gráficos por trama) sin reconocer el contenido real del texto o las figuras.

SIMM Single In-line Memory Module. Módulo simple de memoria en línea. Panel estrecho de circuito impreso de unos 8 cm de longitud, que sostiene ocho o nueve chips de memoria. Los SIMM se enchufan en sockets especiales.

Sketchy En forma o formato de borrador.

SMP Symetrical Multi-Processing. Es la estrategia "todo compartido" del cómputo en paralelo.

SNA Systems Network Architecture. Arquitectura de redes de sistemas. Principal estrategia de IBM para el uso de redes, introducida en 1974. La SNA está compuesta por una variedad de productos de hardware y software que interactúan todos entre sí.

Sun Sun Microsystems, Inc. Fabricante de estaciones de trabajo de alto rendimiento basadas en redes, fundado en 1982.

T

TCP/IP Transmission Control Protocol/Internet Protocol. Protocolo de control de transmisiones/protocolo Internet. Conjunto de protocolos de comunicaciones desarrollado por la Defense Advanced Research Projects Agency (DARPA - Agencia de proyectos de investigación avanzada de defensa) para intercomunicar sistemas diferentes. Se ejecuta en un gran número de computadoras VAX y basadas en UNIX, y es utilizada por muchos fabricantes de hardware, desde los de computadoras personales hasta los de macrocomputadoras. Es empleado por numerosas corporaciones y por casi todas las universidades y organizaciones federales de los Estados Unidos.

Terabyte Un billón de bytes.

Throughput Caudal de procesamiento, rendimiento efectivo o específico. Velocidad con la que puede procesar datos una computadora. El caudal de procesamiento de una computadora es una combinación de las velocidades de entrada y salida de sus periféricos, de su velocidad de procesamiento interno y de la eficiencia de su sistema operativo y demás software de sistema, trabajando todo en conjunto.

Traversing Uno de los métodos comunes para obtener información. Consiste en los resultados de medir un conjunto de líneas consecutivas en lo que respecta a longitud y dirección. Existen variaciones para levantar información sobre rutas o propiedades, por ejemplo.

Trilateration Es uno de los métodos modernos para medir distancias con dispositivos electrónicos. Permiten construir triángulos al medir directamente sus lados.

U

UPS Uninterruptible Power Supply. Fuente de alimentación ininterrumpible. Energía de seguridad para un sistema de computación cuando la energía eléctrica de la línea se interrumpe o baja a un nivel de tensión inaceptable. Los pequeños sistemas UPS proveen energía de baterías por sólo unos pocos minutos; los necesarios para apagar la computadora de manera ordenada. Los sistemas más sofisticados están conectados a generadores eléctricos y pueden proveer energía durante días enteros.

URL Uniform Resource Locator. La forma estándar para indicar la dirección o localización de cualquier recurso (documento u objeto) en Internet.

V

VAR Value Added Reseller. Revendedor de valor añadido. Una organización que añade valor a un sistema y lo vuelve a vender. Por ejemplo, un VAR podría adquirir una computadora y periféricos de proveedores diferentes y un paquete de software para gráficos de otro proveedor, y empaquetar todo junto como un sistema CAD especializado.

W

Workflow Software utilizado para rutear o direccionar automáticamente eventos (trabajo) de un usuario o programa a otro bajo un ambiente cliente/servidor estructurado o no estructurado.

Workstation Estación de trabajo. (1) Micro o minicomputadora para un único usuario, de alto rendimiento, que ha sido especializada para gráficos, diseño asistido por computadora, ingeniería asistida por computadora o aplicaciones científicas. (2) En una red de área local, una computadora personal que sirve a un único usuario, a diferencia de un servidor de archivos, que sirve a todos los usuarios de la red.

WWW World Wide Web. Gran cantidad de recursos que pueden ser accedados utilizando herramientas para el Web. También se refiere al universo de servidores de hipertexto (HTTP) que permiten mezclar texto, gráficos, sonido, etc.

X

Xbase Lenguajes tipo dBASE como Clipper y FoxPro. Originalmente casi idénticos a dBASE, nuevas características y órdenes durante los últimos años solo lo han convertido en compatible parcialmente con dBASE.

Z

Zenithal Sinónimo de azimuthal.

Anexo A. Documentación de Usuario.

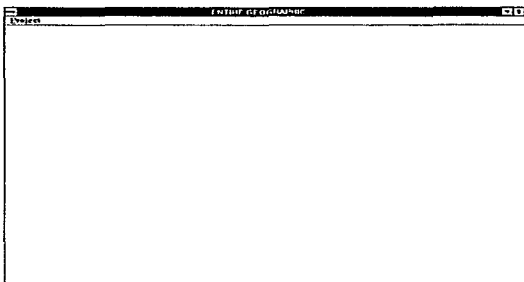
Uso de Natural Geographic

El siguiente documento tiene como objetivo mostrar el uso básico del producto Natural Geographic (ENTIRE Geographic en ambiente PC) y poder utilizar la aplicación desarrollada para la Delegación Tlalpan. Algunas opciones todavía no se encuentran disponibles, por lo que en esta guía sólo se incluyen las que son utilizables. Para mayor información por favor consultar el manual del producto, contactar al personal de sistemas o bien a algún consultor de Software AG.

1. Hacer doble click en el icono de ENTIRE GEOGRAPHIC
2. Presionar OK después del aviso de copyright

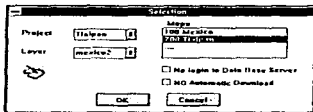


A continuación aparecerá la siguiente pantalla:

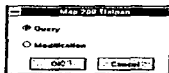


3. Seleccionar Project

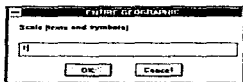
- 3.1. En el campo de Project, seleccionar Tlalpan.
- 3.2. En Layer seleccionar Mexico2.
- 3.3. En maps seleccionar 200 Tlalpan. En caso de desear actualizar la información seleccionar login to database server y automatic download.



4. Presionar OK y seleccionar la opción de query.

















5. A continuación aparece la ventana de escala. Escribir la escala deseada, por ejemplo 1. Esta escala se encuentra en cm.



6. Seleccionar el dispositivo de entrada. En este caso mouse.

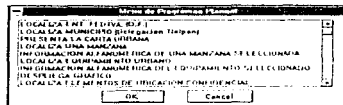


7. A continuación se verá una pantalla en negro y una sección de iconos a la derecha. Los iconos disponibles son:

Menu de aplicaciones.		Selección de un elemento del mapa.	
Creación de una nueva página (zoom).		Enviar al servidor modificaciones hechas en el mapa.	
Cambiar a la pantalla de emulación.		Deshacer.	
Imprimir la página o pantalla activa.		Menú de digitalización.	
Enviar la página activa a plotter.		Ayuda.	
Shell a DOS.		Modificar variables del sistema.	
Cambiar la forma en que se pueden seleccionar elementos (puntos, líneas, puntos y líneas).		Salir.	



8. Seleccionar el icono appl. Aparecerá una ventana donde se encuentra el menú.



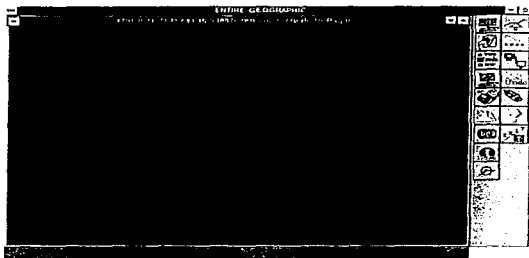
A continuación se muestra el detalle de las opciones disponibles y pantallas similares a las que deben aparecer por cada opción:

a) Localiza entidad federativa (DF).

Esta opción despliega el catálogo de entidades federativas.

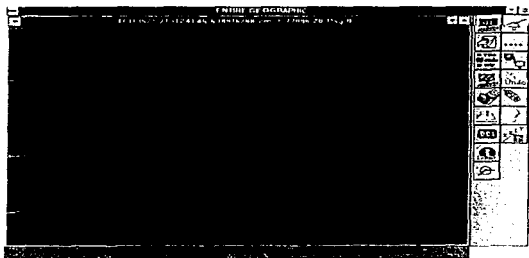


Actualmente la única entidad federativa disponible es el D.F., por lo cual es la única opción a seleccionar:



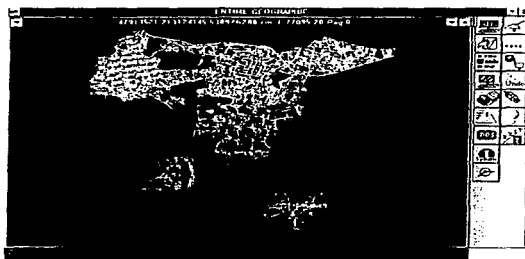
b) Localiza municipio (deleg. tlalpan)

Esta opción ilumina la parte correspondiente a la Delegación Tlalpan



c) Presenta la carta urbana

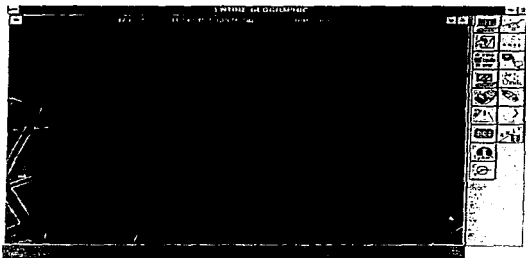
Esta opción despliega presenta una vista general.



d) Localiza una manzana

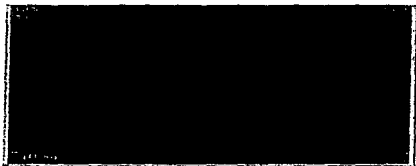
Esta opción presenta la pantalla que pregunta colonia y manzana.. Hace un zoom de una manzana específica.





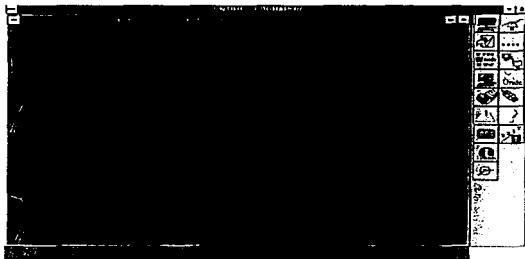
e) Información alfa. de la manzana seleccionada.

Esta opción presenta el detalle de dicha manzana, como colindancias, uso de suelo, etc.



f) Localiza equipamiento urbano.

Esta opción presenta las opciones de equipamiento urbano existentes en el correspondiente catálogo(alfanumérico) hace un zoom y despliega en forma gráfica.



g) Información alfanumérica del equipamiento urbano seleccionado.

Esta opción presenta el detalle del equipamiento urbano seleccionado en el mapa.



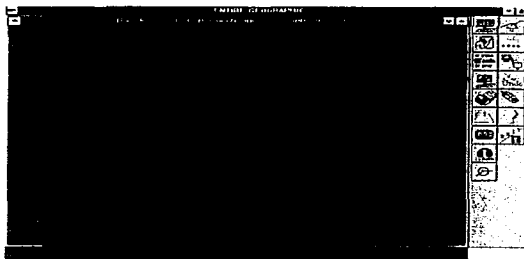
h) Despliega gráfico

Función no habilitada.

i) Localiza elementos de ubicación confidencial.

Esta opción presenta una pantalla alfanumérica de elementos de equipamiento urbano confidencial. Hace un zoom y despliega en el mapa.





j) Imprime vista.

Esta opción permite imprimir la pantalla desplegada.

Anexos

1	CM	USO-COMP2	A	5	N	
1	CN	CONLINDANCIA2	A	1	NN	
1	CO	INTENSIDAD	A	5	NN	
1	CP	CUADRANTE	U	5	NN	
1	CQ	COORDENADA-X	U	5	NN	
1	CR	COORDENADA-Y	U	5	NN	
1	CS	NO-OFFICIAL-2	A	9	NN	D
1	CT	SUP-LLAVE4	A	6		S
*	CB	COLONIA		1:3		
*	CC	MANZANA-2		1:3		

 FDT of file <ME-DEMANDA-CIUDADANA> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
	1	AA	NUM-COL-MAN	A	6		N	
	1	AB	NOM-ARE-SER	A	30		NN	
	1	AC	NOM-SER	A	30		NN	
	1	AD	FEC-REC-SOL	U	6		NN	
	1	AE	ORI-MAN	A	1		NN	
	1	AF	NOM-SOL	A	30		NN	
	1	AG	NOM-ASO-DEP	A	30		NN	
	1	AH	OFICIO	A	16		NN	
	1	AI	NUM-TEL	U	7		NN	
	1	AJ	FEC-ATN	U	6		NN	
	1	AK	DOM-SOL	A	40		NN	
	1	AL	COD-POS	U	5		NN	
M	1	AM	DESCRIPCION	A	60	5	NN	
	1	AN	NUM-FOL	U	6		NN	
	1	AO	LLAVE	A	12		NN	S
*		AA	NUM-COL-MAN		1:6			
*		AD	FEC-REC-SOL		1:6			

 FDT of file <ME-EQUIP-URBANO> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
G	1	BA	REG					
	2	BB	TIPO	A	2		NN	
	2	BC	NOMBRE	A	45		NN	
	2	BD	SECTOR	U	1		NN	
	2	BE	NO-COLONIA	A	3		NN	
	2	BF	MANZANA	A	3		NN	
	2	BG	ORIENTACION	A	1		NN	
	2	BH	NO-OFFICIAL	A	9		NN	
	2	BI	ZONA	A	2		NN	
	2	BJ	ZONA-POSTAL	A	2		NN	
	2	BK	CODIGO-POSTAL	U	5		NN	
	2	BL	TELEFONO	U	10		NN	

Anexos

2	BM	ADMINISTRADOR	A	30		N	
1	BN	SUP-LLAVE2	A	48			S
*	BB	TIPO		1:2			
*	BC	NOMBRE		1:45			
*	BD	SECTOR		1:1			
1	BO	SUP-LLAVE3	A	16			S
*	BE	NO-COLONIA		1:3			
*	BF	MANZANA		1:3			
*	BG	ORIENTACION		1:1			
*	BH	NO-OFICIAL		1:9			

 FDT of file <ME-USO-SUELO> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	AA	CLAVE-SUELO		A	5			N
1	AB	NO-PROGRESIVO		A	3			N
1	AC	DESCRIPCION1		A	60			N
1	AD	DESCRIPCION2		A	60			N
1	AE	DESCRIPCION3		A	60			N
1	AF	SUP-LLAVE		A	8			S
*	AA	CLAVE-SUELO			1:5			
*	AB	NO-PROGRESIVO			1:3			

 FDT of file <NG3-DES-CURVAS> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	BA	CAR-SUP1		A	12			N D
1	BE	CAR-STIP		U	4			N D
1	BO	CAR-DATO		A	32			N D

 FDT of file <NG3-DES-POLIGOS> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	DA	CAR-SUP1		A	12			N D
1	DE	CAR-STIP		U	4			N D
1	DL	CAR-DATO		A	32			N D

 FDT of file <NG3-DES-PUNTOS> in database <NATGEO>

Anexos

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	AA	CAR-SUP1		A	12		N	D
1	AE	CAR-STIP		U	4		N	D
1	AO	CAR-DATO		A	32		N	D

FDT of file <NG3-DES-RASTER> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	FA	CAR-SUP1		A	12		N	D

FDT of file <NG3-DES-TEXTO> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	CA	CAR-SUP1		A	12		N	D
1	CE	CAR-STIP		U	4		N	D
1	CK	CAR-TEXT		A	18		N	D
1	CR	CAR-DATO		A	32		N	D

FDT of file <NG3-DES-USER> in database <NATGEO>

typ	lev	sname	lname	form	leng	occ	comp	desc
uni								
1	HA	TUSE-CLAV		A	8		N	D
1	HB	TUSE-TIPO		A	1		N	

Programas y Mapas

A continuación se muestra el listado de los programas utilizados así como las pantallas o mapas de la aplicación:

```
0010***** //////////////////////////////////////*****
0020***** NATURAL GEOGRAPHIC *****
0030***** *****
0040***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0050***** AUTOR: SAGHEX *****
```

```
0060***** FECHA: MARZO 1994 *****
0070***** DESCRIPCION: MENU PRINCIPAL MEXP0001 *****
0080***** ////////////////////////////////////// *****
0090DEFINE DATA
0100LOCAL
011001 #MAPA (A08)
012001 #PROG (A08)
013001 #FECHA (A08)
014001 #HORA (A08)
015001 #OPC (N01)
0160END-DEFINE
0170RESET #FECHA #HORA #PROG #MAPA #OPC
0180SET KEY PF1='CA0000P0'
0190SET KEY PF3='LOGOFF'
0200MOVE *DATE TO #FECHA
0210MOVE *TIME TO #HORA
0220MOVE *PROGRAM TO #PROG
0230MOVE 'MEXP0001' TO #MAPA
0240MAP USING MAP 'MEXP0001'
0250IF #OPC LT 1 OR #OPC GT 2
0260 REINPUT 'CAPTURE UNA OPCION VALIDA'
0270END-IF
0280IF #OPC EQ 1
0290 FETCH 'MEXP0005'
0300ELSE
0310 FETCH 'MEXP0002'
0320END-IF
0330END
```

Anexos

```

0001***** //////////////////////////////////////////////////
0002***** NATURAL GEOGRAPHIC *****
0003***** *****
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0005***** AUTOR: SAGMEX *****
0006***** FECHA: MARZO 1994 *****
0007***** DESCRIPCION: MENU DE CONSULTAS MEXP0002 *****
0008***** ////////////////////////////////////////////////// *****
0009DEFINE DATA *****
0010LOCAL *****
0011 EQUIPO VIEW OF ME-EQUIP-URBANO *****
0012 2 REG *****
0013 3 TIPO *****
0014 3 NOMBRE *****
0015 3 SECTOR *****
0016 3 NO-COLONIA *****
0017 3 MANZANA *****
0018 3 ORIENTACION *****
0019 3 NO-OFICIAL *****
0020 3 ZONA *****
0021 3 ZONA-POSTAL *****
0022 3 CODIGO-POSTAL *****
0023 3 TELEFONO *****
0024 3 ADMINISTRADOR *****
0025***** CATALOGO DE EQUIP. URBANO *****
0026LOCAL *****
002701 #MAPA (A08) *****
002801 #FECHA (A08) *****
002901 #HORA (A08) *****
003001 #PROG (A08) *****
003101 #OPC9 (N02) *****
003201 #OPC8 (N01) *****
003301 #OPC1 (N01) *****
003401 #DAT (A01) *****
003501 #TIPO (N02) *****
003601 #CALLE (A40) *****
003701 #NOMBRE (A20) *****
003801 #COLONIA (A35) *****
003901 #INICIO (A48) *****
004001 REDEFINE #INICIO *****
0041 2 #ELE-I (N02) *****
0042 2 #NOM-I (A45) *****
0043 2 #SEC-I (A01) *****
004401 #FIN (A48) *****
004501 REDEFINE #FIN *****
0046 2 #ELE-F (N02) *****
0047 2 #NOM-F (A45) *****
0048 2 #SEC-F (A01) *****
0049END-DEFINE *****
0050FORMAT LS=120 *****
0051INCLUDE MEXC0001 ***** // *CONTROL DE PF'S: PF1 PF2 Y PF3
0052MOVE *PROGRAM TO #PROG *****
0053MOVE *DATE TO #FECHA *****
0054MOVE *TIME TO #HORA *****
0055MOVE *MEXM0002' TO #MAPA *****
0056MAP USING MAP 'MEXM0002' *****
0057IF #OPC8 LT 1 OR #OPC8 GT 3 *****

```

```
0058 REINPUT 'CAPTURE UNA OPCION VALIDA'
0059END-IF
0060DECIDE FOR EVERY CONDITION
0061 WHEN #OPCB EQ 1
0062 IF #DAT EQ 'A'
0063 PERFORM CONSULTA-DEMANDA
0064 END-IF
0065 WHEN #OPCB EQ 2
0066 IF #DAT EQ 'A'
0067 PERFORM CONSULTA-COL-MANZ-ALFA
0068 END-IF
0069 IF #DAT EQ 'G'
0070 PERFORM CONSULTA-COL-MANZ-GRAF
0071 END-IF
0072 WHEN #OPCB EQ 3
0073 IF #DAT EQ 'A'
0074 PERFORM CONSULTA-ALFA-EQUIPO
0075 END-IF
0076 IF #DAT EQ 'G'
0077 PERFORM CONSULTA-GRAF-EQUIPO
0078 END-IF
0079 WHEN NONE
0080 IGNORE
0081END-DECIDE
0082DEFINE SUBROUTINE CONSULTA-DEMANDA
0083FETCH 'MEXP4001'
0084END-SUBROUTINE
0085DEFINE SUBROUTINE CONSULTA-COL-MANZ-ALFA
0086FETCH 'MEXP0004'
0087END-SUBROUTINE
0088DEFINE SUBROUTINE CONSULTA-COL-MANZ-GRAF
0089FETCH 'MEXP1004'
0090END-SUBROUTINE
0091DEFINE SUBROUTINE CONSULTA-ALFA-EQUIPO
0092SET CONTROL 'WL64C17FB09/25'
0093MAP USING MAP 'MEXM0006'
0094SET CONTROL 'WB'
0095DECIDE FOR FIRST CONDITION
0096 WHEN #OPC9 EQ 1 MOVE 2 TO #TIPO
0097 WHEN #OPC9 EQ 2 MOVE 03 TO #TIPO
0098 WHEN #OPC9 EQ 3 MOVE 04 TO #TIPO
0099 WHEN #OPC9 EQ 4 MOVE 05 TO #TIPO
0100 WHEN #OPC9 EQ 5 MOVE 07 TO #TIPO
0101 WHEN #OPC9 EQ 6 MOVE 14 TO #TIPO
0102 WHEN #OPC9 EQ 7 MOVE 17 TO #TIPO
0103 WHEN #OPC9 EQ 8 MOVE 19 TO #TIPO
0104 WHEN #OPC9 EQ 9 MOVE 21 TO #TIPO
0105 WHEN #OPC9 EQ 10 MOVE 22 TO #TIPO
0106 WHEN #OPC9 EQ 11 MOVE 25 TO #TIPO
0107 WHEN #OPC9 EQ 12 MOVE 26 TO #TIPO
0108 WHEN #OPC9 EQ 13 MOVE 32 TO #TIPO
0109 WHEN #OPC9 EQ 14 MOVE 34 TO #TIPO
0110 WHEN #OPC9 EQ 15 MOVE 39 TO #TIPO
0111 WHEN #OPC9 EQ 16 MOVE 40 TO #TIPO
0112 WHEN #OPC9 EQ 17 MOVE 41 TO #TIPO
0113 WHEN #OPC9 EQ 18 MOVE 39 TO #TIPO
0114 WHEN #OPC9 EQ 19 MOVE 43 TO #TIPO
```

Anexos

```
0115 WHEN NONE IGNORE
0116END-DECIDE
0117MOVE #TIPO TO #ELE-I #ELE-F
0118MOVE ' ' TO #NOM-I
0119MOVE 0 TO #SEC-I
0120MOVE 9 TO #SEC-F
0121MOVE 'ZZZ' TO #NOM-F
0122IF #OPC9 EQ 19
0123 FIND EQUIPO WITH SUP-LLAVE3 NE '
0124 IF NO RECORD FOUND
0125 WRITE 'NO HAY ELEMENTO'
0126 END-NOREC
0127 IF TIPO EQ '43'
0128 MOVE 'TORIELLO GUERRA'TO #COLONIA
0129 MOVE ADMINISTRADOR TO #CALLE
0130 MOVE NOMBRE TO #NOMBRE
0131 DISPLAY #NOMBRE #CALLE NO-COLONIA MANZANA ORIENTACION
0132 END-IF
0133 END-FIND
0134ELSE
0135 FIND EQUIPO WITH SUP-LLAVE2 EQ #INICIO THRU #FIN
0136 IF NO RECORD FOUND
0137 WRITE 'NO HAY ELEMENTO' #INICIO #FIN
0138 END-NOREC
0139 MOVE ADMINISTRADOR TO #CALLE
0140 MOVE NOMBRE TO #NOMBRE
0141 DISPLAY #NOMBRE #CALLE NO-COLONIA MANZANA
0142 END-FIND
0143END-IF
0144END-SUBROUTINE
0145DEFINE SUBROUTINE CONSULTA-GRAF-EQUIPO
0146FETCH 'MEXP000' /*FIND-ELE
0147END-SUBROUTINE
0148FETCH 'MEXP000'
0149END
```

```

0001***** ////////////////////////////////// .....
0002***** NATURAL GEOGRAPHIC .....
0003***** .....
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO .....
0005***** AUTOR: SAGMEX .....
0006***** FECHA: MARZO 1994 .....
0007***** DESCRIPCION: CONSULTA DE MANZANAS MEXP0004 .....
0008***** ////////////////////////////////// .....
0009DEFINE DATA GLOBAL USING CA0000GO
0010LOCAL USING SAGL0001
0011LOCAL
001201 #REG-01 (A250)
001301 REDEFINE #REG-01
0014 02 #CO (A03)
0015 02 #COL (A03)
0016 02 #NOM (A30)
0017 02 #DES (A20)
0018 02 #MAN (A03)
0019 02 #MA (A03)
0020 02 #NTE (A40)
0021 02 #SUR (A40)
0022 02 #OTE (A40)
0023 02 #PTE (A40)
0024 02 #US1 (A05)
002501 #FEC (A08)
002601 #FECHA (A08)
002701 #HOR (A08)
002801 #HORA (A08)
002901 #MAP (A08)
003001 #MAPA (A08)
003101 #PRO (A08)
003201 #PROG (A08)
003301 LLAVE (A06)
00341 REDEFINE LLAVE
0035 2 LLA-C (A03)
0036 2 LLA-M (A03)
0037END-DEFINE
0038*----- CATALOGO DE MANZANAS -----
0039FORMAT LS=80
0040INCLUDE MEXC0001
0041MOVE *PROGRAM TO #PROG #PRO
0042MOVE *DATE TO #FECHA #FEC
0043MOVE *TIME TO #HORA #HOR
0044MOVE 'MEXM0007' TO #MAP
0045MOVE 'MEXM0005' TO #MAPA
0046SET KEY PF4='MEXP0002'
0047MAP USING MAP 'MEXM0007'
0048IF #CO EQ ' ' OR #MA EQ ' '
0049 REINPUT 'CAPTURA LA CVE-COLONIA Y CVE-MANZANA'
0050END-IF
0051MOVE #CO TO LLA-C
0052MOVE #MA TO LLA-M
0053DECIDE FOR EVERY CONDITION
0054 WHEN #CO EQ '060'
0055 MOVE 'ISIDRO FABELA' TO #NOM
0056 WHEN #CO EQ '061'
0057 MOVE 'AMPLIACION ISIDRO FABELA' TO #NOM

```


Anexos

0058 WHEN #CO EQ '067'
0059 MOVE 'TORIELLO GUERRA' TO #NOM
0060 WHEN NONE IGNORE
0061END-DECIDE
0062FIND MANZANAS WITH SUP-LLAVE4 EQ LLAVE
0063 IF NO RECORD FOUND
0064 REINPUT 'NO SE ENCUENTRA LA CLAVE DADA'
0065 ESCAPE BOTTOM
0066 END-NOREC
0067 MOVE MANZANAS.COLONIA TO #COL
0068 MOVE MANZANAS.MANZANA-2 TO #MAN
0069 MOVE MANZANAS.LIMITE-NORTE TO #NTE
0070 MOVE MANZANAS.LIMITE-SUR TO #SUR
0071 MOVE MANZANAS.LIMITE-ORIENTE TO #OTE
0072 MOVE MANZANAS.LIMITE-PONIENTE TO #PTE
0073 DECIDE FOR FIRST CONDITION
0074 WHEN MANZANAS.US01-DE-SUELO EQ 'H1 '
0075 MOVE 'HAB.HASTA 100 HAB/HA' TO #DES
0076 WHEN MANZANAS.US01-DE-SUELO EQ 'ES '
0077 MOVE 'EQUIPO DE SERV.ADMON.SALUD' TO #DES
0078 WHEN MANZANAS.US01-DE-SUELO EQ 'AV '
0079 MOVE 'AREAS VERDES Y ESP.ABIERTOS' TO #DES
0080 WHEN MANZANAS.US01-DE-SUELO EQ 'H2 '
0081 MOVE 'POLIGONOS EXCEPTUADOS' TO #DES
0082 WHEN MANZANAS.US01-DE-SUELO EQ 'H2B '
0083 MOVE 'HABITACIONAL HASTA 200/HA' TO #DES
0084 WHEN NONE IGNORE
0085 END-DECIDE
0086 MOVE MANZANAS.US01-DE-SUELO TO #US1
0087 MAP USING MAP 'MEXM0005'
0088 ESCAPE BOTTOM
0089END-FIND
0090FETCH 'MEXP0004'
0091END

Ancxos

```

0001***** //////////////////////////////////////////////////
0002***** NATURAL GEOGRAPHIC *****
0003*****
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0005***** AUTOR: SAMGEX *****
0006***** FECHA: MARZO 1994 *****
0007***** DESCRIPCION: MENU DE MANTENIMIENTO MEXP0005 *****
0008***** //////////////////////////////////////////////////
0009DEFINE DATA
0010LOCAL
001101 #MAPA (A08)
001201 #FECHA (A08)
001301 #HORA (A08)
001401 #PROG (A08)
001501 #OPC9 (N02)
001601 #OPC (N01)
001701 #OPC1 (N01)
001801 #TIPO (A01)
0019END-DEFINE
0020RESET #FECHA #HORA #TIPO #MAPA
0021INCLUDE MEXC0001 /*CONTROL DE PF'S: PF1 PF2 Y PF3
0022MOVE *PROGRAM TO #PROG
0023MOVE *DATE TO #FECHA
0024MOVE *TIME TO #HORA
0025MOVE 'MEXM0004' TO #MAPA
0026MAP USING MAP 'MEXM0004'
0027SET KEY PF4='MEXP0002'
0028IF #OPC LT 1 OR #OPC GT 5
0029 REINPUT 'CAPTURE UNA OPCION VALIDA'
0030END-IF
0031DECIDE FOR EVERY CONDITION
0032 WHEN #OPC EQ 1
0033 FETCH 'MEXP4000'
0034 WHEN #OPC EQ 2
0035 FETCH 'MEXP0003'
0036 WHEN #OPC EQ 4
0037 SET CONTROL 'WLJ5C13FB08/30'
0038 MAP USING MAP 'MEXM0006'
0039 SET CONTROL 'WB'
0040 IF #OPC1 GE 1 OR #OPC1 LE 5
0041 MAP USING MAP 'MEXM0004'
0042 END-IF
0043 WHEN NONE
0044 IGNORE
0045END-DECIDE
0046FETCH 'MEXP0005'
0047END

```

```
0001***** ////////////////////////////////////////////////////*****
0002***** NATURAL GEOGRAPHIC *****
0003***** *****
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0005***** AUTOR: SAGMEX *****
0006***** FECHA: MARZO 1994 *****
0007***** DESCRIPCION: CONSULTA GRAFICA DE UNA MANZANA MEXP1004 *****
0008***** ////////////////////////////////////////////////////*****
0009DEFINE DATA GLOBAL USING CA000000
0010LOCAL USING SAGL0001
0011LOCAL
001201 #REG-01 (A250)
001301 REDEFINE #REG-01
0014 02 #CO (A03)
0015 02 #MA (A03)
0016 02 #FEC (A08)
0017 02 #HOR (A08)
0018 02 #MAP (A08)
0019 02 #PRO (A08)
002001 POLIGOS VIEW OF NG3-DES-POLIGOS
0021 2 CAR-SUP1 (A12)
0022 2 REDEFINE CAR-SUP1
0023 3 CAR-PMAP (I2)
0024 3 CAR-MAPA (I4)
0025 3 CAR-NSEC (I4)
0026 3 CAR-NORD (I2)
0027 2 CAR-STIP (N04)
0028 2 CAR-DATO (A32)
0029 2 REDEFINE CAR-DATO
0030 3 CAR-COL (A03)
0031 3 CAR-MAN (A03)
00321 CLAVE-EXT (A11)
00331 REDEFINE CLAVE-EXT
0034 2 TIPO-DATO (A01)
0035 2 CVE-DATO (A10)
00361 LLAVE (A06)
00371 REDEFINE LLAVE
0038 2 COL (A03)
0039 2 MAN (A03)
00401 #CAR-SUP1 (A18)
00411 REDEFINE #CAR-SUP1
0042 2 #CAR-PMAP (N1)
0043 2 #CAR-MAPA (N6)
0044 2 #CAR-CODE (N4)
0045 2 #CAR-NSEC (N7)
0046END-DEFINE
0047** ***** CATALOGO DE MANZANAS *****
0048RESET LLAVE #CAR-SUP1
0049FORMAT LS=80
0050INCLUDE MEXC0001
0051MAP USING MAP 'MEXM0007'
0052IF #CO EQ ' ' OR #MA EQ ' '
0053 REINPUT 'CAPTURA LA CVE-COLONIA Y CVE-MANZANA'
0054END-IF
0055MOVE #CO TO COL
0056MOVE #MA TO MAN
0057FIND MANZANAS WITH SUP-LLAVE4 EQ LLAVE
```

```
0058 IF NO RECORDS FOUND
0059   STACK TOP COMMAND 'CA0000P0'
0060   STOP
0061 END-NOREC
0062 FIND POLIGOS WITH CAR-DA'IO EQ LLAVE
0063   IF NO RECORD FOUND
0064     STACK TOP COMMAND 'CA0000P0'
0065     STOP
0066 END-NOREC
0067 ASSIGN #CAR-PMAP = CAR-PMAP
0068 ASSIGN #CAR-MAPA = CAR-MAPA
0069 ASSIGN #CAR-NSEC = CAR-NSEC
0070 ASSIGN #CAR-CODE = CAR-STIP
0071 IF CAR-PMAP NE 0
0072   WRITE WORK FILE 5 #CAR-SUP1
0073*  WRITE 'CAR-SU1' #CAR-SUP1
0074   END-IF
0075 END-FIND
0076END-FIND
0077**STACK TOP COMMAND 'CA0000P0'
0078**STOP
0079END
```

```

0001***** ////////////////////////////////// //*****
0002***** NATURAL GEOGRAPHIC *****
0003***** *****
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0005***** AUTOR: SAGMEX *****
0006***** FECHA: MARZO 1994 *****
0007***** DESCRIPCION: CARGA DEMANDA CIUDADANA MEXP4000 *****
0008***** ////////////////////////////////// //*****
00010DEFINE DATA
00020LOCAL
00031 DEMANDA VIEW OF ME-DEMANDA-CIUDADANA
00042 NUM-COL-MAN
00052 NOM-ARE-SER
00062 NOM-SER
00072 FEC-REC-SOL
00082 ORI-MAN
00092 NOM-SOL
00102 NOM-ASO-DEP
00112 OFICIO
00122 NUM-TEL
00132 FEC-ATN
00142 DOM-SOL
00152 COD-POS
00162 DESCRIPCION(1:10)
00172 NUM-FOL
0018***2 LLAVE
0019LOCAL
002001 #REGISTRO (A250)
002101 REDEFINE #REGISTRO
0022 02 #FEC (A8)
0023 02 #HOR (A8)
0024 02 #PROG (A8)
0025 02 #MAP (A8)
0026 02 #NOM (A40)
0027 02 #FER (N06)
0028 02 #ARE (A30)
0029 02 #CMA (A06)
0030 02 #COP (N05)
0031 02 #FOL (N06)
0032 02 #OFI (A16)
0033 02 #NAD (A30)
0034 02 #ORI (A01)
0035 02 #SER (A10)
003601 #DES (A40/1:05)
003701 #DOM (A40)
003801 #RES (A01)
003901 #CONT (N02)
0040END-DEFINE
0041**
0042FORMAT LS=80
0043MOVE *DATE TO #FEC
0044MOVE *TIME TO #HOR
0045MOVE *PROGRAM TO #PROG
0046MOVE 'MEXM0014' TO #MAP
0047** ***** CARGA DEMANDAS CIUDADANA *****
0048INCLUDE MEXC0001
0049INPUT USING MAP 'MEXM0014'

```

0050SET KEY PF4='MEXP0005'
0051MOVE #NOM TO NOM-SOL
0052MOVE #FER TO FEC-REC-SOL
0053MOVE #ARE TO NOM-ARE-SER
0054MOVE #CMA TO NUM-COL-MAN
0055MOVE #COP TO COD-POS
0056MOVE #FOL TO NUM-FOL
0057MOVE #OFI TO OFICIO
0058MOVE #NAD TO NOM-ASO-DEP
0059MOVE #ORI TO ORI-MAN
0060MOVE #SER TO NOM-ARE-SER
0061MOVE #DES(*) TO DESCRIPCION(1:5)
0062MOVE #DOM TO DOM-SOL
0063 STORE DEMANDA
0064 END TRANSACTION
0065 ADD 1 TO #CONT
0066FETCH 'MEXP4000'
0067END

Anexos

```

0001***** //////////////////////////////////////////////////
0002***** NATURAL GEOGRAPHIC *****
0003*****
0004***** SISTEMA: PROTOTIPO NATGEO-MEXICO *****
0005***** AUTOR: SAGMEX *****
0006***** FECHA: MARZO 1994 *****
0007***** DESCRIPCION:CONSULTA ALFANUM.DEM. CIUDADANA MEXP4001 *****
0008***** //////////////////////////////////////////////////
00010DEFINE DATA
0002LOCAL
000301 DEMANDA VIEW OF ME-DEMANDA-CIUDADANA
00040 1 NUM-COL-MAN
00050 2 NOM-ARE-SER
00060 2 NOM-SER
00070 2 FEC-REC-SOL
00080 2 ORI-MAN
00090 2 NOM-SOL
00100 2 NOM-ASO-DEP
00110 2 OFICIO
00120 2 NUM-TEL
00130 2 FEC-ATN
00140 2 DOM-SOL
00150 2 COD-POS
00160 2 DESCRIPCION(1:10)
00170 2 NUM-FOL
00180***2 LLAVE
00190LOCAL
0020001 #REGISTRO (A250)
0021001 REDEFINE #REGISTRO
00220 02 #FECHA (A8)
00230 02 #FEC (A8)
00240 02 #HORA (A8)
00250 02 #HOR (A8)
00260 02 #PROG (A8)
00270 02 #MAP (A8)
00280 02 #MAPA (A8)
00290 02 #NOM (A40)
00300 02 #FER (N06)
00310 02 #ARE (A10)
00320 02 #CMA (A06)
00330 02 #COP (N05)
00340 02 #POL (N06)
00350 02 #NUMF (N06)
00360 02 #OFI (A16)
00370 02 #NAD (A10)
00380 02 #ORI (A01)
00390 02 #SER (A10)
0040001 #DES (A10/1:05)
0041001 #DOM (A40)
0042001 #RES (A01)
0043001 #CONT (N02)
00440END-DEFINE
00450**
00460FORMAT LS-80
00470MOVE *DATE TO #FEC #FECHA
00480MOVE *TIME TO #HOR #HORA
00490MOVE *PROGRAM TO #PROG

```

```
00500MOVE 'MEXM0013' TO #MAP
00510MOVE 'MEXM0015' TO #MAPA
00520** ***** CARGA DEMANDAS CIUDADNA -----
00530INCLUDE MEXC0001
00540MAP USING MAP 'MEXM0015'
00550IF #NUMF EQ 0
00560 REINPUT 'CAPTURE UN DATO VALIDO'
00570END-IF
00580FIND DEMANDA WITH NUM-FOL EQ #NUMF
00590 IF NO RECORD FOUND
00600 WRITE 'NO EXISTE REGISTRO' #NUMF
00610 END-NOREC
00620 MOVE NOM-SOL TO #NOM
00630 MOVE FEC-REC-SOL TO #FER
00640 MOVE NOM-ARE-SER TO #ARE
00650 MOVE NUM-COL-MAN TO #CMA
00660 MOVE COD-POS TO #COP
00670 MOVE NUM-FOL TO #FOL
00680 MOVE OFICIO TO #OFI
00690 MOVE NOM-ASO-DEP TO #NAD
00700 MOVE ORI-MAN TO #ORI
00710 MOVE NOM-ARE-SER TO #SER
00720 MOVE DESCRIPCION(1:5) TO #DES(*)
00730 MOVE DCM-SOL TO #DOM
00740 ADD 1 TO #CONT
00750 MAP USING MAP 'MEXM0013'
00760 INCLUDE MEXC0001
00770 SET KEY PF4='MEXP0002'
00780END-FIND
00790FETCH 'MEXP4001'
00800END
```


Mapa: MEXM0001
Descripción: Menu Principal

NATURAL															
Natural Geographic															
DELEGACION TLALPAM															
- Menu Principal -															
1. Mantenimiento															
2. Consultas															
Opcion: >															
PF1-Cartografia PF3-Salida															
Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>

Mapa: MEXM0002
Descripción: Menu de Consultas

NATURAL																
Natural Geographic																
DELEGACION TLALPAM																
CONSULTAS																
1. Demanda Ciudadana																
2. Colonias y Mancomunales																
3. Equipamiento Urbano																
Opciones:																
Grafica(G) AIFanumerica(A):																
PF1=Cartografia					PF2=Menu Principal					PF3=Salida						
Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>

Mapa: MEXM0004
Descripción: Menu de Mantenimiento

NATURAL

Natural Geografic

DELEGACION TLALPAM

- Mantenimiento -

1. Demanda Ciudadana
2. Colonias y Manzanas
3. Serv. X Area Operativa
4. Equipamiento Urbano
5. Asociaciones y Dependencias

Opcion:

PF1-Cartografia PF2-Menu Principal PF3-Salida

Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>
-------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	----

Mapa: MEXM0005
Descripción: Consulta de Manzanas

NATURAL																
Natural Geographic																
Delegación Tlalpan																
Consulta																
Cve. Colonia:	Nombre:															
Cve. Manzana:	Uso-Suelo:															
Calle-Nro:	Calle:															
Calle-Dte:	Calle:															
Calle-Pte:	Calle:															
PF1-Cartografía PF2-Menu Inicial PF3-Salida PF4-Menu Anterior																
Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>

Mapa: MEXM0006

Descripción: Equipamiento Urbano

--- NATURAL ---

Equipamiento

Urbano:

- | | |
|-----------------------|---------------------------|
| 1. Jardín de Niños | 10. Banco |
| 2. Primaria | 11. Inst. Religiosa |
| 3. Secundaria | 12. Centro Asistencia |
| 4. Preparatoria | 13. Auditorio |
| 5. CEN | 14. Restaurant |
| 6. Hospital | 15. Inmueble Ser. Publico |
| 7. Centro Comercial | 16. Laboratorio |
| 8. Parque y Jardín | 17. Fabrica |
| 9. Casa de la Cultura | 18. Oficina |
| | 19. Casillas Electorales |

Opcion: |

Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>
-------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	----

Mapa: MEXM0007

Descripción: Entrada de datos para consulta de manzanas

NATURAL														
Natural Geographic														
DELEGACION TLALPAM														
- Consultas -														
Clave-Colonia: <input type="text"/>														
Clave-Manzana: <input type="text"/>														
PF1-Cartografia PF2-Menu Principal PF3-Salida														
Enter	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	>>

Mapa: MEXM0014

Descripción: Entrada de datos para consultas de demanda ciudadana

NATURAL

1/1

Natural Geographic

Delegación Tlalpan

Demanda Ciudadana

Nombre del solicitante: | _____ Fol: _____
Asociación/Dependencia: | _____ Col-Man: | _____
MUN. Origen: | _____ Fecha-Recop.: | _____ D.P.: | _____
Domicilio: | _____
Orientación: | Servicio: | _____
Nombre del área de servicio: | _____
Descripción: | _____
| _____
| _____
| _____

PF1-Cartografía: PF2-Menu Principal PF3-Salida PF4-Menu Anterior

Enter Test Test Test Test Test Test Test Test Test Test Test Test Test Test Test >>

Mapa: MEXM0015

Descripción: Captura de folio para consultas.

NATURAL																
Natural Geographic																
Delegación Tlalpan																
- Consultas -																
Capture Folio:																
PF1-Cartografía PF2-Menu Principal PF3-Salida PF4-Menu anterior																
Enter	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	Test	>>

Guía de Instalación y Configuración

Este documento es una guía de los procedimientos de instalación del producto Natural Geographic, al cual se hará referencia como Nat-Geo. Cabe mencionar que para mayores detalles y definiciones puede ser necesario el consultar la documentación del producto. Contempla la instalación en un entorno IBM-PC así como VAX-PC.

IBM-PC

Para la instalación en el equipo IBM (con sistema operativo VSE o MVS), se cuenta con el manual del producto (Reference Manual), capítulo de instalación en mainframe. Este entorno tiene como pre-requisito hardware y software de emulación 3270 en la PC, el cual debe estar configurado y funcionando.

Las recomendaciones a este respecto son:

1. Cuidar que el proceso de instalación deje cargados los archivos de la cartografía identificados con los números:

- a) 186 (archivo para las curvas de nivel).
- b) 187 (archivo para los "centroides").
- c) 188 (archivo de poligonales y digitalización).

2. Deberá estar instalado el producto Natural Connection

3. Una vez instalado el producto, debe tenerse en cuenta que el Usuario o la sesión del usuario que use la aplicación de Nat-Geo, deberá entrar directamente a la biblioteca en la que se encuentran ubicados los programas que utilizan el producto y que han sido escritos por personal de sistemas. Además se deberá tener definida como "steplib" la biblioteca del producto NG1311 o bien NGE311.

4. Antes de utilizar el producto y empezar a definir el ambiente de trabajo, es necesario correr el programa CAINPL, ubicado dentro de la biblioteca de la aplicación (y teniendo como STEPLIB la NG1311 en este caso y siempre). Este programa lo que hace es cargar el Usuario Administrador de Nat-Geo conocido como CARTO-A.

5. Una vez que esta etapa se ha realizado exitosamente, se puede continuar tomando como referencia el manual identificado como "N/Geo 1/5".

De manera general se indican los pasos a seguir antes de iniciar la carga y descarga de la cartografía entre el Mainframe y la PC:

- 5.1 Definición de usuarios CARTO-P y CARTO-U
- 5.2 Definición del Proyecto
- 5.3 Definición de Subtipos
- 5.4 Definición de la(s) Capa(s) de Código(s)
- 5.5 Definición del Mapa (cartográfico)

6. La etapa de ejecución del "Upload" y "Download" deberán de ser realizadas habiendo cargado previamente el área de datos global, esto es, ejecutar el programa CAMENUP0 y conectándose con la sesión del usuario CARTO-U.

Nota: Solo para el caso de "Upload" y "Download", los parámetros de CARTO-U deben contemplar la opción de (M) Modificación del mapa definido previamente. Por otro lado, se debe cuidar la trayectoria de todos los programas que son activados desde cada uno de los procedimientos de Up/Download. También hay que verificar en "Option"... "Profile"... "Com"... que las opciones "OEM" y "EABS", se encuentren seleccionadas.

Nota muy importante: En caso de tener que regenerar un mapa en el servidor y borrar el existente, se debe correr el programa CABORMAP (ojo, en el servidor), el cual pide como parámetros el proyecto y el mapa.

Para el caso de la instalación en el lado de la PC:

7. Modificar el archivo `Intw\home\config.nc` como sigue:

- 7.1 La cláusula "COMBOARD" debe tener el parámetro "IRMA"
- 7.2 verificar los paths de CONFIG.NC y .NCPs

8. Insertar en el drive correspondiente el disco identificado como "SCINCE", teclear "scince" y presionar "Enter".

8.1 Para llegar a la información cartográfica, seleccionar información por Entidad Federativa.

8.2 Seleccionar (para el caso inicial) la Entidad Distrito Federal, y apuntar a la Delegación Tlalpan.

8.3 Utilizar la opción "Exportar a formato DXF".

8.4 El paso anterior genera el archivo de la cartografía. Este archivo será transformado con utilerías de Nat-Geo.

8.5 Cuando se despliega el mapa en Nat-Geo de la PC, es necesario tomar nota de las coordenadas de las esquinas extremas del mismo (estas serán requeridas en el servidor para la generación del mapa de ese lado).

9. El producto cuenta con un disco de instalación, que lo instalará bajo el directorio GISG.

10. Es necesario realizar algunas modificaciones al archivo variable.gis (en el directorio GISG):

10.1 El parámetro NATURALCONNECTION debe apuntar al lugar en que Natural Connection ubica todos sus procedimientos tipo ".ncp"

10.2 En la sección de "definición de variables para las pantallas" se debe tener en cuenta lo correspondiente al caso de IBM 3270.

10.3 En la sección de "definición de posiciones de variables en los Mapas de Natural de lectura de mapas", también tener en cuenta la sección de IBM 3270.

11. En el mismo directorio (GISG), se encuentran los archivos b_x.nat, c_x.nat, a_r.nat, a_s.nat, a_n.nat, a_t.nat, m_r.nat, m_s.nat, m_n.nat y m_t.nat... en todos ellos se debe verificar que la ejecución de su correspondiente archivo .ncp tenga la ruta exacta de los procedimientos de Natural Connection.

12. En el directorio "\\GISG\TRABAJOS\proy\lcapal" se haya el archivo denominado "mexico2.ncp", que por requerimientos del producto (aún y cuando es un procedimiento .ncp), debe residir a este nivel. El contenido de este archivo tiene que ver con el procedimiento automatizado de conexión a la sesión de trabajo con la VAX.

13. "Debajo" de "\\GISG\TRABAJOS\proy\l" se encuentran los directorios:

- 13.1 \\GISG\TRABAJOS\proy\lcapa**
- 13.2 \\GISG\TRABAJOS\proy\lmapa**
- 13.3 \\GISG\TRABAJOS\proy\lPLANNAT**

donde:

proy = número de proyecto (1-9).

capa = nombre de la capa de código.
mapa = directorio con la cartografía digitalizada.

14. Una vez que se ha instalado el Entire Geographic (que es como se le conoce al producto del lado de la PC) se hará la depuración de la cartografía para que pueda ser cargada en el servidor y replicada en la PC.

15. Generar un directorio \GISG\TRABAJOS\proy\mapa, en donde mapa será sustituido por el número de mapa que se va a generar, ubicando aquí nuestro archivo de la cartografía renombrado como "RESTO".

16. Entrar a Geographic de la PC y seleccionar la opción "project".

En la primer ventana seleccionar el nombre de la capa de código.

En la segunda ventana seleccionar el numero o nombre del mapa, que en este caso es nuevo (no existe).

En la tercer ventana, marcar la opción de los puntos suspensivos

Cuando se presiona "Enter", nos pide el número de mapa que va a ser generado, así como una descripción.

Al presionar "Enter", aparecerá un panel con los campos que contendrán las coordenadas de las esquinas del mapa que vamos a trabajar.

17. Con la secuencia anterior se generan automáticamente en el \GISG\TRABAJOS\proy\mapa_recien_definido... los archivos:

- esquinas
- nivel
- indice.niv
- resto
- indice.res
- texto
- simbol

Exceptuando "esquinas" y "resto" todos son generados con cero registros.

18. Cambiar hacia el directorio \GISG\ y escribir los siguientes comandos:

- XLIM num_proyecto num_mapa

Ejemplo: XLIM 6 100

O bien, (de no existir el programa XLIM)

- GENMAPA -p 6 -m 200 -f filepath -l filimite
- GESTADIS 6 200
- GESRANGO 6 200

Donde:

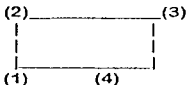
- 6 = Número de proyecto
- 200 = nuevo número de mapa(*)
- filepath = Nombre del archivo que contiene la ruta del directorio con nuestro mapa(traido de INEGI).
- filimite = Nombre del archivo que contiene las coordenadas UTM de las esquinas del mapa cartográfico. La estructura de este es como sigue:

(Desde la posición 2 en todas las líneas)

```
U
0047405671 0212015575
0047405671 0213508450
0048947575 0213508450
0048947575 0212015575
```

Notas: La "u" debe ser mayúscula e indica que se están utilizando coordenadas UTM (Universal Transversa de Mercator).

El orden de los valores de las coordenadas es de 1 a 4 considerando que al observar el mapa de frente, se tiene el siguiente orden:



Muy importante: En la PC, los valores de las coordenadas son CENTIMETROS. En el servidor, y al momento de definir el mapa, estos se consideran METROS, esto es:

```
474056.71 2120155.75
474056.71 2135084.50
489475.75 2135084.50
489475.75 2120155.75
```


Se recomienda usar un mapa(*) nuevo en caso de llegar a requerir el uso de estas utilerías ya que el GENMAPA sobre-escribe en el mapa ya existente.

El directorio tendrá el siguiente contenido:

- esquinas
- nivel
- indice.niv
- resto
- indice.res
- texto
- simbol
- estadis
- doc
- rangos
- hojas
- desdate

VAX-PC

En el caso de la instalación en un entorno VMS-Windows(PC).

1. Instalar en el equipo VAX como sigue:

1.1 Situarse dentro del directorio en el que se va a instalar.

1.2 Dar el comando `BACKUP/LOG t8xw1z$mia6:*.*/sa *.*`

1.3 Lo anterior deja entre otros archivos, uno denominado README.1ST el cual contiene el detalle de la forma de instalar cada uno de los archivos descargados.

1.3.1 En el caso del archivo NATGEO-FILES.EXP debe tenerse en consideración la versión de ADABAS existente en el ambiente, ya que de no coincidir con la de los archivos 186, 187 y 188, estos no se podrán descargar. Aun en el caso de construir estos en forma manual, se llegan a presentar problemas al momento de inicializar la aplicación.

1.3.2 En el caso de la carga del archivo de errores, se debe asignar el Natural Work File número 1 (NATWORK01) y ejecutar la utilidad ERRLODUS:

```
$ define natwork01.dat uld-err.dat
$ nat

INPL
```

1.3.3 Si es el caso, que se está transportando un aplicación desarrollada en otra plataforma (por ejemplo IBM), se sugiere la herramienta "SYSTRANS", tanto para mover de IBM hacia una PC como para hacer la opción de carga en el ambiente nuevo desde la PC o un Diskette.

1.3.4 En caso de tener problemas para la ejecución del programa CAINPL, verificar los parámetros de sesión del usuario que esta tratando de acceder al Nat-Geo (NATPARM). Hay que hacer lo mismo si el problema lo ocasiona el programa CAMENU0.

Para la instalación de Entire Geographic en la PC conectada a la VAX:

1. Editar el archivo \ntw\home\config.nc como sigue:

- COMBOARD debe tener el parámetro "VAX"
- ENV "VAX PATHWORKS"
- Verificar los paths de CONFIG.NC y .NCps

2. Verificar el archivo \GISG\variable.gis y cambiar el grupo de parámetros que corresponden a un entorno VAX/UNIX denominado como:

"Definición de variables para las pantallas de salida para VAX/UNIX"

"Definición de posiciones de variables en Mapas-Natural VAX/UNIX"

Si están activos los correspondientes a IBM 3270, hay que cambiarlos.

3. Instalar el producto Pathworks para PC. Esta instalación debe realizarse para la tarjeta de emulación que está montada en la PC.

4. Por otro lado hay que hacer modificaciones a los archivos autoexec.bat y config.sys como sigue:

```
***** AUTOEXEC.BAT *****
@c:\windows\lad_wrap.com
c:\windows\lad-dos.com
@echo off
lh c:\windows\smartdrv.exe A- B- C
loadhigh c:\dos\keyb sp,437,c:\dos\keyboard.sys
loadhigh c:\dos\doskey
prompt $p$g
set path=c:\;c:\dos;c:\windows;c:\ntw;c:\decnet
call \decnet\startnet
if not %_sysd%x==x path %path%;%_sysd%mswinv30
set path=%path%;i:\;i:\decnet;i:\pcapp
i:\mswinv30\rplmem
c:\decnet\lcterm
latcp define scb 2
c:\decnet\lat
set nvtwin=1
c:
cd \
```

```
***** CONFIG.SYS *****
DEVICE=c:\windows\himem.sys
DOS=high,umb
DEVICE=c:\windows\emm386.exe noems
SHELL=c:\dos\command.com c:\dos\ /E:400 /P
FILES=50
BUFFERS=12
COUNTRY=34,437,C:\DOS\COUNTRY.SYS
DEVICEHIGH=C:\DOS\ANSI.SYS
STACKS=9,256
DEVICE=\DECNET\PROTMAN.SYS //:C:\DECNET
DEVICE=\DECNET\EXP16.DOS
lastdrive=z
```

5. Se utilizarán las utilerías para transferencia de archivos propias del ambiente VAX.

5.1 Todos los procedimientos del tipo ".ncp", deben modificarse de manera tal que para efecto de mover información entre la PC y la VAX se utilizará la utilería NFT. Los procedimientos de Natural Connection se modificarán tomando como referencia los siguientes ejemplos:

```
***** PROCEDIMIENTO "UPLOAD.NCP" *****
set response 2000
set display on
input #parm1 1 'Project'
input #parm2 6 'Map'
set #parm5 " diti\manuel sistemas\::disco2:[ortiz]"
set #parm4 " c:\gisg\trabajos\ #parm1 \ \ #parm2 \texto."
set #parm3 " copy /image" #parm4 #parm5 "texto."
==> os NFT #parm3
type 'cartexbi ' cr
type #parm2 cr
perform wait
set #parm5 " diti\manuel sistemas\::disco2:[ortiz]"
set #parm4 " c:\gisg\trabajos\ #parm1 \ \ #parm2 \simbol."
set #parm3 " copy /image" #parm4 #parm5 "simbol."
==> os NFT #parm3
type 'carpunbi ' cr
type #parm2 cr
perform wait
set #parm5 " diti\manuel sistemas\::disco2:[ortiz]"
set #parm4 " c:\gisg\trabajos\ #parm1 \ \ #parm2 \nivel."
set #parm3 " copy /image" #parm4 #parm5 "nivel."
==> os NFT #parm3
type 'carcurbi ' cr
type #parm2 cr
perform wait
set #parm5 " diti\manuel sistemas\::disco2:[ortiz]"
set #parm4 " c:\gisg\trabajos\ #parm1 \ \ #parm2 \resto."
set #parm3 " copy /image" #parm4 #parm5 "resto."
==> os NFT #parm3
type 'carresbi ' cr
type #parm2 cr
perform wait
type 'ca0000p0 ' cr
exit
:wait
pause 1000
wait *screen eq 'ca0000p0' return
```

***** PROCEDIMIENTO "SHOWPLOT.NCP" *****

```
set #parm1 "dlt\manuel sistemas':"
set #parm2 "disco2:[ortiz]dato.dat"
set #parm3 "c:\gisgldato.dat"
set #parm4 "copy /image " #parm1 #parm2 #parm3
===> os NTF #parm4
type 'mexp0001' cr
```

Notas: El parámetro "dlt\manuel sistemas':disco2:[ortiz]" corresponde a la sintaxis del comando "NFT copy /image". Donde:

dlt = nombre del nodo VAX (como se ve desde la PC).
manuel = nombre de la sesión de usuario dentro de la VAX.
sistemas = password de la sesión del usuario 'manuel'.
disco2:[ortiz] = directorio dentro de la VAX (destino).

5.2 Del paso anterior debe tenerse en cuenta que en el caso de los procedimientos ".ncp", deben sustituirse los comandos de Natural Connection "down" y "up" por sus correspondientes "NFT copy /image" (esta utilidad es propia del Pathworks de la PC).

Anexo C. Catálogos

A continuación se presenta información de los siguientes catálogos:

- Catálogo de Equipamiento Urbano**
- Catálogo de Colonias por Zona**
- Catálogo de Manzanas por Colonia**
- Catálogo de Uso de Suelo**

D E L E G A C I O N T L A L P A N

C A T A L O G O D E C O L O N I A S P O R Z O N A

ZONA	NUMERO	COLONIA
	177	COLONIA VIVEROS DE COATEILAN
ZONA	01	
	076	BARRIO EL TRIENITO
	077	BARRIO NIÑO JESUS
	059	BARRIO SAN FERNANDO
	061	COLONIA AMPLIACION ISIDRO FABELA
	173	COLONIA AMPLIACION SAN JUAN TEPEXIMILPA
	106	COLONIA ARENAL TEPEPAN (ANTES ARENAL TLALPAN)
	068	COLONIA BELISARIO DOMINGUEZ SECCION XVI
	196	COLONIA CANTERA
	064	COLONIA CANTERA PUENTE DE PIEDRA
	069	COLONIA CENTRO DE TLALPAN I
	070	COLONIA CENTRO DE TLALPAN II
	063	COLONIA CHIMALCOYOTL
	063	COLONIA COMUNEROS DE SANTA URSULA
	111	COLONIA CORRAL
	090	COLONIA CUMBRES DE TEPETONGO
	060	COLONIA ISIDRO FABELA
	079	COLONIA LA JOYA
	191	COLONIA LA NOPALERA ZONA ARQUEOLOGICA
	076	COLONIA LOS VOLCANES
	064	COLONIA MESA LOS HORNOS
	093	COLONIA MIRADOR 1a. SECCION
	094	COLONIA MIRADOR 2a. SECCION
	095	COLONIA MIRADOR 3a. SECCION
	088	COLONIA PEDREGAL DE LAS AGUILAS
	087	COLONIA PEDREGAL DE SANTA URSULA XITLA
	065	COLONIA PUEBLO QUIETO
	062	COLONIA ROMULO SANCHEZ MIRELES
	089	COLONIA SAN JUAN TEPEXIMILPA
	066	COLONIA SAN PEDRO APOSTOL
	073	COLONIA SANTA URSULA XITLA
	092	COLONIA SANTISIMA TRINIDAD
	091	COLONIA TEPETONGO
	085	COLONIA TEXCALTECO
	074	COLONIA TLALCOLIGIA
	086	COLONIA TEXCALTECO LA MESA
	067	COLONIA TORTILLO GUERRA
	112	COLONIA VALLE DE TEPEPAN
	110	COLONIA VALLE ESCONDIDO
	018	EQUIPAMIENTO BALDIO PERIFERICO PICACHO-AJUSCO
	107	FRACCIONAMIENTO CLUB DE GOLF MEXICO
	109	FRACCIONAMIENTO COLINAS DEL BOSQUE
	043	FRACCIONAMIENTO CONDOMINIO DEL BOSQUE
	012	FRACCIONAMIENTO FUENTES DEL PEDREGAL
	035	FRACCIONAMIENTO INSURGENTES CUICUILCO
	034	FRACCIONAMIENTO JARDINES DE LA MONTANA
	108	FRACCIONAMIENTO LAS TORTOLAS
	015	FRACCIONAMIENTO LOS FRAMROYANES
	037	FRACCIONAMIENTO PARQUES DEL PEDREGAL

DELEGACION TLALPAN

CATALOGO DE COLONIAS POR ZONA

NUMERO	COLONIA
013	FRACCIONAMIENTO RINCON DEL PEDREGAL
036	FRACCIONAMIENTO RINCONADA SANTA TERESA
078	FRACCIONAMIENTO SAN BUENAVENTURA
036	FRACCIONAMIENTO VILLA CHARRA DEL PEDREGAL
081	FRACCIONAMIENTO VILLA TLALPAN
042	R. ECOLOGICO BOSQUE DE TLALPAN
041	R. ECOLOGICO ESPACIO ECOLOGICO CUICUILCO
040	R. ECOLOGICO MANANTIAL PEÑA POBRE
072	R. ECOLOGICO PARQUE NACIONAL FUENTES DROTANTES
014	UNIDAD HABITACIONAL CONJUNTO RESIDENCIAL PEDREGAL
017	UNIDAD HABITACIONAL EMILIO PORTES GIL (PEMEX)
200	UNIDAD HABITACIONAL FAROLES DEL PEDREGAL
082	UNIDAD HABITACIONAL ISSIFAM
039	UNIDAD HABITACIONAL MIGUEL HIDALGO VILLA OLIMPICA
011	UNIDAD HABITACIONAL PEDREGAL DEL LAGO
194	UNIDAD HABITACIONAL POLOTLAN
195	UNIDAD HABITACIONAL RANCHO CANELA INFONAVIT
075	UNIDAD HABITACIONAL RESIDENCIAL INSURG. SUR I Y II
080	UNIDAD HABITACIONAL TORRES TLALPAN
058	UNIDAD HABITACIONAL ZAPOTE I, II Y III
ZONA 02	
165	BALDIO CANAL NACIONAL
129	COLONIA ARSA
122	COLONIA ARBOLEDAS DEL SUR (EX-EJIDOS DE HUIPULCO)
127	COLONIA ARENAL DE GUADALUPE
130	COLONIA BELISARIO DOMINGUEZ
134	COLONIA EX-HACIENDA SAN JUAN DE DIOS
141	COLONIA GRANJAS COAPA
128	COLONIA GUADALUPE
157	COLONIA MAGISTERIAL
139	COLONIA NUEVA ORIENTAL COAPA
133	COLONIA POTRERO DE ACOXPA
135	COLONIA SAN BARTOLO EL CHICO (EX-EJIDOS DE TEPEPAN)
105	COLONIA SAN LORENZO HUIPULCO
119	EQUIPAMIENTO COLEGO MEXICO
120	EQUIPAMIENTO FABRICA FISISA
147	EQUIPAMIENTO PLAZA COAPA
117	EQUIPAMIENTO PREPARATORIA NO. 5
123	FRACCIONAMIENTO CHIMALI
136	FRACCIONAMIENTO FLORESTA COYODACAN
142	FRACCIONAMIENTO GABRIEL RAMOS MILLAN
125	FRACCIONAMIENTO HACIENDA DE SAN JUAN
124	FRACCIONAMIENTO HACIENDA DE SAN JUAN DE TLALPAN
138	FRACCIONAMIENTO PRADOS COAPA 1a. SECCION
131	FRACCIONAMIENTO PRADOS COAPA 2a. SECCION
132	FRACCIONAMIENTO PRADOS COAPA 3a. SECCION
126	FRACCIONAMIENTO RANCHO LOS COLORINES
144	FRACCIONAMIENTO REAL DEL SUR
140	FRACCIONAMIENTO RESIDENCIAL ACOXPA
181	FRACCIONAMIENTO RESIDENCIAL HACIENDA COAPA
151	FRACCIONAMIENTO RESIDENCIAL VILLA COAPA SUP.MZA. 4
152	FRACCIONAMIENTO RESIDENCIAL VILLA COAPA SUP.MZA. 5

DELEGACION TLALPAN

CATALOGO DE COLONIAS POR ZONA

NUMERO	COLONIA
155	FRACCIONAMIENTO RESIDENCIAL VILLA COAPA SUP.MZA. 8
189	FRACCIONAMIENTO RESIDENCIAL VILLA PRADO COAPA
190	FRACCIONAMIENTO RINCON DE SAN JUAN
159	FRACCIONAMIENTO RINCONADA COAPA 1a. SECCION
156	FRACCIONAMIENTO RINCONADA COAPA 2a. SECCION
145	FRACCIONAMIENTO RINCONADA LAS HADAS
137	FRACCIONAMIENTO VERGEL COAPA
118	FRACCIONAMIENTO VERGEL DE COYOACAN
188	FRACCIONAMIENTO VERGEL DEL SUR
187	FRACCIONAMIENTO VERGEL TLALPAN
180	FRACCIONAMIENTO VILLA BINALPADENA
158	FRACCIONAMIENTO VILLA CUERANCO
160	FRACCIONAMIENTO VILLA DEL PUENTE
184	FRACCIONAMIENTO VILLA DEL SUR
121	FRACCIONAMIENTO VILLA LAZARO CARDENAS
146	FRACCIONAMIENTO VILLA ROYALE
162	UNIDAD HABITACIONAL CONJUNTO URBANO CUERANCO
161	UNIDAD HABITACIONAL HUESO PERIFERICO
164	UNIDAD HABITACIONAL HUESO PERIFERICO
163	UNIDAD HABITACIONAL IGNACIO CHAVEZ
186	UNIDAD HABITACIONAL JARDINES VILLA COAPA
143	UNIDAD HABITACIONAL LOTERIA NACIONAL
148	UNIDAD HABITACIONAL NARCISO MENDOZA SUPER MZA. 1
149	UNIDAD HABITACIONAL NARCISO MENDOZA SUPER MZA. 2
150	UNIDAD HABITACIONAL NARCISO MENDOZA SUPER MZA. 3
154	UNIDAD HABITACIONAL NARCISO MENDOZA SUPER MZA. 6
153	UNIDAD HABITACIONAL NARCISO MENDOZA SUPER MZA. 7
185	UNIDAD HABITACIONAL RESIDENCIAL MIRAMONTES
192	UNIDAD HABITACIONAL SAZUALES CEBADALES
142	UNIDAD HABITACIONAL TENORIOS FOVISSSTE
176	UNIDAD HABITACIONAL TENORIOS INFONAVIT I
178	UNIDAD HABITACIONAL TENORIOS INFONAVIT II
183	UNIDAD HABITACIONAL TRES FUENTES

ZONA 03

047	BARRIO CAMISETAS CALVARIO
045	BARRIO CUEVITAS DE CURAMAGUEY
051	BARRIO EL CAPULIN
049	BARRIO EL METRO
048	BARRIO LA FAMA
046	BARRIO LA LONJA
021	COLONIA HEROES DE PADIERNA
019	COLONIA LOMAS DE PADIERNA
044	COLONIA MIGUEL HIDALGO 1a. SECCION
050	COLONIA MIGUEL HIDALGO 2a. SECCION
052	COLONIA MIGUEL HIDALGO 3a. SECCION
053	COLONIA MIGUEL HIDALGO 4a. SECCION
001	COLONIA PEDREGAL SAN NICOLAS 1a. SECCION
002	COLONIA PEDREGAL SAN NICOLAS 2a. SECCION
003	COLONIA PEDREGAL SAN NICOLAS 3a. SECCION
004	COLONIA PEDREGAL SAN NICOLAS 4a. SECCION
014	COLONIA POPULAR SANTA TERESA
022	COLONIA TORRES DE PADIERNA

DELEGACION TLALPÁN

CATALOGO DE COLONIAS POR ZONA

NUMERO	COLONIA
020	COLONIA ZONA EJIDOS DE PADIERNA O JARDINES DEL AJU
071	UNIDAD HABITACIONAL FUENTES BROTANTES FOVISSSTE
ZONA 04	
027	COLONIA AMPLIACION CUCHILLA DE PADIERNA
032	COLONIA DELVEDERE
067	COLONIA BOSQUES DEL PEDREGAL
005	COLONIA CHICHICASPATI
033	COLONIA CHIMILLI
031	COLONIA CRUZ DEL FAROL
025	COLONIA CUCHILLA DE PADIERNA
024	COLONIA CULTURA MAYA (ANTES LOPEZ PORTILLO)
010	COLONIA DOS DE OCTUBRE
055	COLONIA LA PRIMAVERA
028	COLONIA LOMAS ALTAS DE PADIERNA
009	COLONIA LOMAS DE CUILOTEPEC
026	COLONIA LOMAS HIDALGO
023	COLONIA LOS ENCINOS
029	COLONIA MIRADOR I
030	COLONIA MIRADOR II
054	COLONIA PARAJE 3A
006	COLONIA SECTOR XVII
056	COLONIA VERANO
008	COLONIA VISTAS DEL PEDREGAL
ZONA 05	
203	COLONIA AMPLIACION LA MAGDALENA PETLACALCO
199	COLONIA CASERIO LA CIMA
113	COLONIA EJIDOS DE SAN PEDRO MARTIR
204	COLONIA EL CHARCO
205	COLONIA EL CONVENIO
206	COLONIA EL DIVISADERO
114	COLONIA FUENTES DE TEPEPAN
166	COLONIA HEREDAS DE 1910
208	COLONIA JUVENTUD UNIDA
209	COLONIA LA PALMA
115	COLONIA LA TORTUGA XOLALPA
210	COLONIA LAS MARGARITAS
211	COLONIA LOMAS CUARTETLAN
100	COLONIA MARIA ESTHER ZUNO DE ECHEVERRIA
201	COLONIA MIRADOR DEL VALLE
202	COLONIA MOVIMIENTO ORGANIZADO DE TLALPÁN
174	COLONIA NUEVO RENACIMIENTO DE AXALCO
175	COLONIA PARAJE TETENCO
207	COLONIA PEDREGAL DE TOPILEDO
103	COLONIA PLAN DE AYALA
212	COLONIA SAN MIGUEL TEHUIZCO
213	COLONIA SAN MIGUEL TOCIAC
197	COLONIA SAN NICOLAS TOTOLAPAN
214	COLONIA TECOENTITLA
098	COLONIA TECOBRAL
099	COLONIA TLALMILLE
215	COLONIA VALLE VERDE

D E L E G A C I O N T L A L P A N

CATALOGO DE COLONIAS POR ZONA

NUMERO	COLONIA
104	COLONIA VIVEROS DE CUERNAVACA
192	COLONIA ZONA DE TEPEPAN
116	EQUIPAMIENTO H. COLEGIO MILITAR
198	FRACCIONAMIENTO PEDREGAL DE SAN NICOLAS
101	FRACCIONAMIENTO TLALPUENTE
169	PUEBLO MARDALENA PETLACALCO
172	PUEBLO PARRIS O EL GUARDA
102	PUEBLO SAN ANDRES TOTOLTEPEC
170	PUEBLO SAN MIGUEL AJUSCO
167	PUEBLO SAN MIGUEL TOPILEJO
168	PUEBLO SAN MIGUEL XICALCO
097	PUEBLO SAN PEDRO MARTIR
171	PUEBLO SANTO TOMAS AJUSCO
057	R. ECOLOGICO PARQUE NACIONAL AJUSCO
193	UNIDAD HABITACIONAL SAN PEDRO MARTIR FOVISSTE

SECRETARIA DE FOMENTO
FECHA: 17/08/92

SISTEMA DE CATALOGOS
REPORTE POR CLASE DE CATALOGOS

PAGINA:
HORA: 10:13:26

NUMERO DE CATALOGO ANEX TIPO DE BIENES INMUEBLES (EQUIPAMIENTO URBAN
CATALOGO RESERVAION

01	UNIVERSIDAD
02	JARDIN DE NIÑOS (PRE-PRIMARIA)
03	PRIMARIA
04	SECUNDARIA
05	PREPARATORIA
06	UNIVERSIDAD
07	D. C. H.
08	C. BACHILLERES
09	CONVICO
10	C. F. T. I. S.
11	CONSULT. DELEG. TLALPAM
12	CONSULTORIO DE S. S. A.
13	CONSULTORIO PARTICULAR
14	HOSPITAL
15	FINANCIER
16	FINANCIER
17	CENTRO COMERCIAL
18	CENTRO DEAFASIVO
19	PUEBLOS ADJUDIN
20	ESTADISTICA
21	CASA DE LA CULTURA
22	GANCO
23	IGLESIA CATOLICA
24	IGLESIA PROTESTANTE
25	INSTITUCION RELIGIOSA
26	CENTRO DE ASISTENCIA
27	MUSEO
28	TEATRO
29	FINE
30	PLAZA CIVICA
31	EXPOSICION
32	AUDITORIA
33	SALON DE FIESTAS
34	RESTAURANT
35	CARDINERIA
36	LIMPIASIA
37	ALCANTARILLA
38	TIANGUIS

TOTAL DE REGISTROS PARA CATALOGO 0007 ---- 38

RE05101
SCATALIB

DELEGACION DEL D D F EN TLALPAN
CATALOGO DE USO DE SUELO

PAG: 1
24/02/94

CLAVE: H05 000
HABITACIONAL HASTA 50 HAB/HA (LOTE TIPO 1000 M2)

CLAVE: H1 000
HABITACIONAL HASTA 100 HAB/HA (LOTE TIPO 500 M2)

CLAVE: H2B 000
HABITACIONAL HASTA 200 HAB/HA (LOTE TIPO 250 M2) SERVICIOS
BASICOS

CLAVE: H4 000
HABITACIONAL HASTA 400 HAB/HA (LOTE TIPO 125 M2)

CLAVE: H8 000
HABITACIONAL HASTA 800 HAB/HA (LOTE TIPO PLURIFAMILIAR)

CLAVE: H2S 000
HABITACIONAL HASTA 200 HAB/HA /SERVICIOS

CLAVE: H4S 000
HABITACIONAL HASTA 400 HAB/HA /SERVICIOS

CLAVE: H2I 000
HABITACIONAL HASTA 200 HAB/HA /INDUSTRIA MEZCLADA

CLAVE: H4I 000
HABITACIONAL HASTA 400 HAB/HA /INDUSTRIA MEZCLADA

CLAVE: H2IS 000
HABITACIONAL HASTA 200 HAB/HA /INDUSTRIA MEZCLADA/SERVICIOS

CLAVE: H4IS 000
HABITACIONAL HASTA 400 HAB/HA /INDUSTRIA MEZCLADA/SERVICIOS

CLAVE: SU 000
SUBCENTRO URBANO

CLAVE: C 000
CORREDOR URBANO/HABITACIONAL/OFCINAS/INDUSTRIA

CLAVE: CS 000
CORREDOR URBANO/HABITACIONAL/OFCINAS/SERVICIOS

REG05101
SCATALIB

DELEGACION DEL D D F EN TLALPAN
CATALOGO DE USO DE SUELO

PAG: 2
24/02/94

CLAVE: CB	000 CENTRO DE BARRIO
CLAVE: ES	000 EQUIPAMIENTO DE SERVICIOS, ADMINISTRACION, SALUD, EDUCACION Y CULTURA
CLAVE: EA	000 EQUIPAMIENTO DE ABASTO
CLAVE: ED	000 EQUIPAMIENTO DE DEPORTES Y RECREACION
CLAVE: EP	000 EQUIPAMIENTO DE PROTECCION Y SEGURIDAD
CLAVE: EM	000 EQUIPAMIENTO HORTUORIO
CLAVE: EC	000 EQUIPAMIENTO DE COMUNICACIONES Y TRANSPORTES
CLAVE: EI	000 EQUIPAMIENTO DE INFRAESTRUCTURA
CLAVE: AV	000 AREAS VERDES Y ESPACIOS ABIERTOS
CLAVE: IV	000 INDUSTRIA VECINA
CLAVE: IA	000 INDUSTRIA AISLADA
CLAVE: AMR	000 AGRICOLA DE MEJORAMIENTO Y REHABILITACION
CLAVE: AI	000 AGRICOLA INTENSIVO
CLAVE: AC	000 AGRICOLA DE CONSERVACION

RE05101
SCATALIB

DELEGACION DEL D D F EN TLALPAN
CATALOGO DE USO DE SUELO

PAG: 3
24/02/94

CLAVE: APE 000
AGRICOLA DE PROTECCION ESPECIAL

CLAVE: AP 000
AGRICOLA PERENNE

CLAVE: PE 000
PECUARIO ESTABLADO

CLAVE: PS 000
PECUARIO SEMIESTABLADO

CLAVE: PC 000
PASTOREO CONTROLADO

CLAVE: PEFM 000
FORESTAL MULTIPLE

CLAVE: PEFR 000
FORESTAL RESTRINGIDO

CLAVE: PERC 000
AREAS RECREATIVAS Y CULTURALES

CLAVE: PR 000
POBLADO RURAL

CLAVE: ZEDEC 000
ZONAS ESPECIALES DE DESARROLLO CONTROLADO

CLAVE: H2 000
POLIGONOS EXCEPTUADOS POR ACUERDO DEL C. JEFE DEL D.D.F. CON
FECHA 11 DE MAYO DE 1988

CLAVE: RV 000
ESPACIOS ABIERTOS EN LOS POLIGONOS DELIMITADOS EN ESTA FORMA
SE ENCUENTRAN CONDICIONADOS A QUE EL DEPARTAMENTO DFL DIC-
TRITO FEDERAL PUEDA DOTARLOS DE INFRAESTRUCTURA

REDES 102
SCATALIB

DELEGACION DEL D. D. F. EN TLALPAM
CATALOGO DE USO DE SUELO
PROGRESIVO POR CLAVE

PAG: 1
24/02/94

CLAVE: H2B
000

HABITACIONAL HASTA 200 HAB/HA (LOTE TIPO 250 M2) SERVICIOS
BASICOS

001

UNA VIVIENDA

002

DOS VIVIENDAS

003

DE 3 A 50 VIVIENDAS

004

DE 51 A 251 VIVIENDAS (SUJETO A LICENCIA)

005

MAS DE 251 VIVIENDAS (SUJETO A LICENCIA)

006

REPRESENTACIONES OFICIALES Y EMBAJADAS EXTRANJERAS (SUJETO A
LICENCIA)

007

CONSULTORIOS, AGENCIAS DE VIAJE, CASAS DE CAMBIO HASTA 40
METROS CUADRADOS

008

TIENDAS DE PRODUCTOS BASICOS, DE ESPECIALIDADES Y ARTICULOS
EN GENERAL HASTA 40 METROS CUADRADOS

009

VENTA DE REPARACIONES, LLANTAS Y ACCESORIOS DE VEHICULOS SIN
TALLER DE REPARACION HASTA 40 METROS CUADRADOS

010

TALLER DE REPARACION DE MAQUINARIA, LAVADORAS O REFRIGERA-
DORES, BICICLETAS HASTA 40 METROS CUADRADOS

011

TALLERES MENORES DE HERRERIA, CARPINTERIA, EBANISTERIA, TA-
LABARERIA, CALZADO Y PRODUCTOS ARTESANALES HASTA 40 METROS
CUADRADOS

012

BADOS, SANITARIOS PUBLICOS, GIMNASIOS, SAUNA Y MASAJES
(ADIESTRAMIENTO FISICO) (SUJETO A LICENCIA)

013

SALAS DE BELLEZA, LAVANDERIAS, TINORERIAS, SASTRERIAS, ES-
TUDIO Y LABORATORIO FOTOGRAFICO, REPARACION DE ELECTRODOMES-
TICOS Y ARTICULOS EN GENERAL HASTA 40 METROS CUADRADOS

014

TALLER MECANICO DENTAL, LABORATORIO MENOR DE ANALISIS CLINI-
COS Y RADIOGRAFIAS HASTA 40 METROS CUADRADOS

015

CENTROS DE INTEGRACION JUVENIL Y FAMILIAR, ORFANATOS, ASILOS
DE ANCIANOS, CASAS CUNA U OTRAS INSTITUCIONES DE ASISTENCIA

REC05102
SCATALIS

DELEGACION DEL D O F EN TLALPAN
CATALOGO DE USO DE SUELO
PROGRESIVO POR CLAVE

PAG: 2
24/02/94

CLAVE: H2B

- 016 SALONES DE CORTE, CLINICAS O DISPENSARIOS VETERINARIOS, TIENDAS DE ANIMALES Y ACCESORIOS, FARMACIAS VETERINARIAS HASTA 40 METROS CUADRADOS
- 017 GUARDERIAS, JARDINES DE NIÑOS O ESCUELAS PARA NIÑOS ATIPICOS (SUJETO A LICENCIA)
- 018 ESCUELAS PRIMARIAS (SUJETO A LICENCIA)
- 019 GALERIAS DE ARTE, MUSEOS O CENTROS DE EXPOSICIONES TEMPORALES O AL AIRE LIBRE (SUJETO A LICENCIA)
- 020 BIBLIOTECAS O HEMEROTECAS
- 021 INSTALACIONES RELIGIOSAS SEMINARIOS O CONVENTOS (SUJETO A LICENCIA)
- 022 CAFES, FUENTES DE SODAS, FONDAS Y LONCHERIAS SIN VENTA DE BEBIDAS ALCOHOLICAS, JUGOS Y LICUADOS, TAQUERIAS, ESTANQUILLOS, COCINAS ECONOMICAS, TORTERIA HASTA 40 METROS CUADRADOS
- 023 CENTROS COMUNITARIOS, CENTROS CULTURALES Y SALONES PARA FIESTAS INFANTILES (SUJETO A LICENCIA)
- 024 CANCHAS DEPORTIVAS CUBIERTAS HASTA 5000 METROS CUADRADOS (SUJETO A LICENCIA)
- 025 ALBERGAS, CANCHAS Y PISTAS DEPORTIVAS AL AIRE LIBRE (SUJETO A LICENCIA)
- 026 GARITAS O CASETAS DE VIGILANCIA
- 027 ESTACIONAMIENTOS PUBLICOS, PRIVADOS Y SITIOS DE TAXIS
- 028 AGENCIAS DE CORREOS, TELEGRAFOS Y TELEFONOS
- 029 CENTRALES TELEFONICAS SIN SERVICIO AL PUBLICO (SUJETO A LICENCIA)
- 030 ESTACIONES O SUBESTACIONES (SUJETO A LICENCIA)
- 031 ESTACIONES DE BOMBO, PLANTAS DE TRATAMIENTO O CARCAMOS (SUJETO A LICENCIA)

RE05102
SCATALIR

DELEGACION DEL D. F. EN TLALPAN
CATALOGO DE USO DE SUELO
PROGRESIVO POR CLAVE

PAG: 3
24/02/94

CLAVE: H2B
032

ESTACIONES DE TRANSFERENCIA DE BASURA (SUJETO A LICENCIA)

033

PLAZAS, EXPLANADAS, JARDINES O PARQUES

034

HORTALIZAS, HUERTOS, FLORES, PLANTAS, VIVEROS, INVERNADEROS
DE TRASPATIO

COLOMBIA 647 CALLES TORILLO GUERDA

ZONA DE

870 LIMITE AL NORTE	LIMITE AL SUR	LIMITE AL OCCIDENTE	LIMITE AL ORIENTE	USOS DE SUELO
001 AV DEL PESQUERAL	APATILADO	CAJÍ TELURPAN	SAN AUGUSTIN	00
002 PEREZURBEO	FERRUCARREIL	HUQUEZAO	TEPEPAN	00
003 APATILADO	BUVAPALCO	CAJÍ TELURPAN	FERRUCARREIL-PEPERIZADO	00
004 APATILADO	PERIZATECO	CAJÍ TELURPAN	HUQUEZAO	00
005 FERRUCARREIL	SAN JUAN DE DIOS	USOCORATE	FERRUCARREIL	00
006 FERRUCARREIL	SAN JUAN DE DIOS	TEPEPAN	USOCORATE	00
007 PERIZATECO	SAN JUAN DE DIOS	CAJÍ TELURPAN	TEPEPAN	00
008 LAS AMARILLAS	FERRUCARREIL	SAN JUAN DE DIOS	SILLON DE MEXICANA	00
009 SAN JUAN DE DIOS	SILLON DE MEXICANA	USOCORATE	FERRUCARREIL	00
010 SAN JUAN DE DIOS	SILLON DE MEXICANA	TEPEPAN	USOCORATE	00
011 SAN JUAN DE DIOS	DEL RIO	CAJÍ TELURPAN	TEPEPAN	00
012 LAS AMARILLAS	FERRUCARREIL	CARRANCO	NIERAN COMITES	00
013 PUENTE DE PIEDRA	FERRUCARREIL	SILLON DE MEXICANA	ADMITEPEC	00
014 FRANCISCO WELLA	CUTILANAMAC	USOCORATE	LAS FUENTES	00
015 LAS AMARILLAS	CHERAPACOTUIL	ROXICUEJAN	TEPEPAN	00
016 LAS AMARILLAS	CHERAPACOTUIL	CARRANCO	ROXICUEJAN	00
017 LAS AMARILLAS	CHERAPACOTUIL	DEL RIO	CARRANCO	00
018 LAS AMARILLAS	CHERAPACOTUIL	CAJÍ	DEL RIO	00
019 LAS AMARILLAS	CHERAPACOTUIL	ADMITEPEC	CAJÍ	00
020 PUENTE DE PIEDRA	PUENTE DE PIEDRA	ADMITEPEC	CAJÍ	00
021 FERRUCARREIL	FERRUCARREIL	ADMITEPEC	CAJÍ	00
022 SILLON DE MEXICANA	ADMITEPEC	ADMITEPEC	ADMITEPEC	00
023 SILLON DE MEXICANA	ADMITEPEC	ADMITEPEC	ADMITEPEC	00
024 CHERAPACOTUIL	CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	00
025 CHERAPACOTUIL	CUTILANAMAC	ROXICUEJAN	AUSCO	00
026 CHERAPACOTUIL	CUTILANAMAC	CARRANCO	USOCORATE	00
027 CHERAPACOTUIL	CUTILANAMAC	DEL RIO	TEPEPAN	00
028 PUENTE DE PIEDRA	PUENTE DE PIEDRA	ADMITEPEC	ADMITEPEC	00
029 FERRUCARREIL	FERRUCARREIL	ADMITEPEC	ADMITEPEC	00
030 ADMITEPEC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
031 ADMITEPEC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
032 ADMITEPEC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
033 CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
034 CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
035 CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
036 CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
037 CUTILANAMAC	CAJÍ TELURPAN	ADMITEPEC	ADMITEPEC	00
038 PUENTE DE PIEDRA	PUENTE DE PIEDRA	ADMITEPEC	ADMITEPEC	00
039 FERRUCARREIL	FERRUCARREIL	ADMITEPEC	ADMITEPEC	00
040 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
041 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
042 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
043 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
044 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
045 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
046 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
047 CHERAPACOTUIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
048 FERRUCARREIL	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
049 PEÑA POBRE	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
050 PEÑA POBRE	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
051 PEÑA POBRE	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
052 PEÑA POBRE	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
053 COAPA	PEÑA POBRE	ADMITEPEC	ADMITEPEC	00
	NO SAN FERRANDO	ADMITEPEC	ADMITEPEC	00

RESUMEN
SENAZ

DECLARACION DEL OBITO EN TULUMPA
CATALOGO DE BARRIOS POR COLONIA

06/02/74

COLONIA 047 COLONIA TORRELLA GUERRA

ZONA 03

ATA LIMITE AL NORTE	LIMITE AL SUR	LIMITE AL ORIENTE	LIMITE AL PONIENTE	USOS DE SUELO
054 COMPA	83 SAN FERNANDO	TESORITOS	LAS FUENTES	BI
055 COMPA	84 SAN FERNANDO	MOCTEZUMA	TESORITOS	BI
056 COMPA	85 SAN FERNANDO	CACAHUAC	MOCTEZUMA	BI
057 COMPA	86 SAN FERNANDO	IMPULSION	CANQUESO	BI
058 COMPA	87 SAN FERNANDO	FERRUCARTELL	IMPULSION	BI
059 COMPA	88 SAN FERNANDO	JACATEPEC	FERRUCARTELL	BI
060 COMPA	89 SAN FERNANDO	NOBELS	JACATEPEC	BI
061 COMPA	90 SAN FERNANDO	LUIS G. ZUÑIGA	NOBELS	BI
062 COMPA	91 SAN FERNANDO	CAJZ TULUMPA	LUIS G. ZUÑIGA	BI

RESUMEN
SENAEZ

DELEGACION DEL B O F EN TUNJUNPA
CENTRO DE EMPRESAS PDM COLOMBIA

801/21/M

C/CLAFIA SAS COLONIA ESQUER ANTELA

ZONA 01

PKZ	LIMITE AL NORTE	LIMITE AL SUR	LIMITE AL OCCIDENTE	LIMITE AL PONIENTE	USOS DE SUELO
001	PERIFERICO	PIEDRA FERRAZ	46 PTE	ZAPOTE	ES
002	PERIFERICO	26 MTE	30 PTE	46 PTE	NO
003	PERIFERICO	26 MTE	24 PTE	34 PTE	NO
004	PERIFERICO	26 MTE	14 PTE	24 PTE	NO
005	PERIFERICO	26 MTE	JUNTA F. REFORMA	34 PTE	NO
006	PERIFERICO	26 MTE	14 OTE	JUNTA F. REFORMA	NO
007	PERIFERICO	26 MTE	34 OTE	14 OTE	NO
008	PERIFERICO	26 MTE	34 OTE	24 OTE	NO
009	PERIFERICO	26 MTE	44 OTE	34 OTE	NO
010	PERIFERICO	26 MTE	54 OTE	44 OTE	NO
011	PIEDRA JUNTA	PIEDRA FLOSOZAL	PIEDRA CALZADA	ZAPOTE	NO
012	PIEDRA JUNTA	PIEDRA FLOSOZAL	PIEDRA FERRAZ	PIEDRA CALZADA	NO
013	PIEDRA JUNTA	PIEDRA FLOSOZAL	PIEDRA ALVARADO	PIEDRA FERRAZ	NO
014	PIEDRA JUNTA	PIEDRA FLOSOZAL	PIEDRA CALZADA	PIEDRA ALVARADO	NO
015	PIEDRA JUNTA	PIEDRA FLOSOZAL	PIEDRA FERRAZ	PIEDRA CALZADA	NO
016	PIEDRA JUNTA	PIEDRA FLOSOZAL	46 PTE	PIEDRA FERRAZ	NO
017	26 MTE	14 MTE	34 PTE	46 PTE	NO
018	26 MTE	14 MTE	24 PTE	34 PTE	NO
019	26 MTE	14 MTE	14 PTE	24 PTE	NO
020	26 MTE	14 MTE	JUNTA F. REFORMA	14 PTE	NO
021	26 MTE	14 MTE	14 OTE	JUNTA F. REFORMA	NO
022	26 MTE	14 MTE	24 OTE	14 OTE	NO
023	26 MTE	14 MTE	34 OTE	24 OTE	NO
024	26 MTE	14 MTE	44 OTE	34 OTE	NO
025	26 MTE	14 MTE	54 OTE	44 OTE	NO
026	PERIFERICO	14 MTE	44 OTE	54 OTE	NO
027	PERIFERICO	14 MTE	14 OTE	44 OTE	NO
028	PERIFERICO	14 MTE	44 OTE	14 OTE	NO
029	PERIFERICO	14 MTE	44 OTE	44 OTE	NO
030	PERIFERICO	14 MTE	104 OTE	44 OTE	NO
031	PERIFERICO	14 MTE	114 OTE	104 OTE	NO
032	PERIFERICO	14 MTE	124 OTE	114 OTE	NO
033	PERIFERICO	14 MTE	124 OTE	124 OTE	NO
034	PIEDRA FLOSOZAL	14 PTE	46 PTE	PIEDRA ALVARADO	NO
035	14 MTE	CON ZAPOTE	46 PTE	ZAPOTE	NO
036	14 MTE	124 OTE	34 PTE	46 PTE	NO
037	14 MTE	124 OTE	24 PTE	34 PTE	NO
038	14 MTE	124 OTE	14 PTE	24 PTE	NO
039	14 MTE	124 OTE	JUNTA F. REFORMA	14 PTE	NO
040	14 MTE	BENITO JUAREZ	14 OTE	JUNTA F. REFORMA	NO
041	14 MTE	BENITO JUAREZ	24 OTE	14 OTE	NO
042	14 MTE	BENITO JUAREZ	34 OTE	24 OTE	NO
043	14 MTE	BENITO JUAREZ	44 OTE	34 OTE	NO
044	14 MTE	BENITO JUAREZ	54 OTE	44 OTE	NO
045	14 MTE	BENITO JUAREZ	64 OTE	54 OTE	NO
046	14 MTE	BENITO JUAREZ	74 OTE	64 OTE	NO
047	14 MTE	BENITO JUAREZ	84 OTE	74 OTE	NO
048	14 MTE	BENITO JUAREZ	94 OTE	84 OTE	NO
049	14 MTE	BENITO JUAREZ	104 OTE	94 OTE	NO
050	14 MTE	BENITO JUAREZ	114 OTE	104 OTE	NO
051	14 MTE	BENITO JUAREZ	124 OTE	114 OTE	NO
052	14 MTE	BENITO JUAREZ	124 OTE	124 OTE	NO
053	CON ZAPOTE	124 OTE	46 PTE	ZAPOTE	NO

ESTADO:
SECRET

DECLARACION DEL N.º 1 EN TULUMPA
CAMPEANO DE HANNAH'S POR COLONIA

08/03/94

COLONIA 061 COLONIA APLICACION ISSIROS TABLA

FORMA 01

REN	LEJANTE AL MONTE	LEJANTE AL SUR	LEJANTE AL OCCIDENTE	LEJANTE AL PONIENTE	USOS DE SUELO
001	JUANES	VECENTE QUICERO	LAS FUENTES	HANSON 1	R20
002	CANTERA	VECENTE QUICERO	HANSON 4	NA. CERRADO	R20
003	CANTERA	VECENTE QUICERO	COM. 44. 011	HANSON 6	R20
004	CANTERA	VECENTE QUICERO	HANSON 3	COM. 44. 011	R20
005	CANTERA	VECENTE QUICERO	HANSON 4	HANSON 5	R20 ES
006	CANTERA	VECENTE QUICERO	HANSON 1	HANSON 4	R20
007	CANTERA	VECENTE QUICERO	HANSON 2	HANSON 1	R20
008	CANTERA	VECENTE QUICERO	HANSON 1	HANSON 2	R20
009	VECENTE QUICERO	QUEZELARQUE	LAS FUENTES	JONH F. HANSON	R20