



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**DISEÑO DE SISTEMAS DE INFORMACION PARA
PUBLICACIONES DE CIRCULACION CONTROLADA**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

A C T U A R I O

P R E S E N T A:

IAACOV CONTRERAS ARANGO



DIRECTORA DE TESIS: LIC. EN CC. KARLA RAMIREZ PULIDO

2005



**FACULTAD DE CIENCIAS
SECCION ESCOLAR**

0350272



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.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo reespecial.

NOMBRE: IAACOV CONTRERAS ARANGO

FECHA: 22 NOV - 2005

FIRMA: [Handwritten Signature]

ACT. MAURICIO AGUILAR GONZÁLEZ
Jefe de la División de Estudios Profesionales de la
Facultad de Ciencias
Presente

Comunicamos a usted que hemos revisado el trabajo escrito:

"Diseño de sistemas de información para publicaciones de circulación controlada"
realizado por **Iaacov Contreras Arango**

con número de cuenta 09332680-5 , pasante de la carrera de Actuaría

Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis
Propietario

Lic. en C.C. Karla Ramírez Pulido

[Handwritten Signature]

Propietario

M. en C. José de Jesús Galaviz Casas

[Handwritten Signature]

Propietario

Lic. en C.C. Francisco Lorenzo Solsona Cruz

[Handwritten Signature]

Suplente

M. en C. Javier García García

[Handwritten Signature]

Suplente

M. en C. Egar Arturo García Cárdenas

[Handwritten Signature]

Consejo Departamental de Matemáticas



Act. Jaime Vázquez Ramírez

CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

*Eres todo un ejemplo para mí, la guía en éste camino que conocemos como vida. Gracias por llevarme por un buen camino y por todo tu amor, pero sobre todo gracias a Dios por haberte escogido como mi Madre, por permitirnos seguir juntos y permitirme compartir contigo éste momento.
Te amo con todo mi corazón Madre...*

*Inés, tu representas junto con mi Madre y mi hermana lo que más quiero en éste mundo, muchas gracias por todo tu amor y apoyo, gracias por haberme hecho un hombre de bien.
Te amo mamá Inés.*

*Miriam, gracias por creer en mí, todos los días le agradezco a Dios la bendición que me regaló al tenerte como hermana y amiga.
Te amo hermana.*

Andrea, quiero agradecerte todo tu amor y apoyo. Te amo bruja.

Cuñado, agradezco de corazón toda tu ayuda, toda la lata que me diste para acabar con esto me fue de gran ayuda.

Karla, sin tu asesoría, amabilidad y dedicación no hubiera podido terminar éste trabajo. Te agradezco infinitamente todas tus atenciones y el tiempo que me dedicaste.

Paco, muchísimas gracias por toda tu ayuda, asesoría y tiempo.

José, Javier, Egar les agradezco mucho el tiempo que me dedicaron.

Diseño de sistemas de información para publicaciones de circulación gratuita controlada.

Indice

Parte I.- Introducción y situación actual del mercado.

Capítulo 1.- Introducción.	1
Capítulo 2.- Situación actual del mercado.	5

Parte II.- Diseño y justificación del sistema de información.

Capítulo 3.- Diseño y justificación del sistema de información	8
3.1. Hardware.	8
3.2. Software.	9
3.3. Sistemas operativos.	9
3.4. Sistemas de administración de base de datos.	12
3.5. Lenguajes de programación: Java, PHP y Visual Basic.	15
3.6. Redes.	20
3.7. Modelo Cliente-Servidor.	24
3.7.1. Procedimientos.	24
3.7.2. Usuarios.	24
3.8. Desarrollo del sistema de información en conjunto.	24

Parte III.- Análisis, diseño, implementación y futuro del sistema de información de Netmedia.

Capítulo 4.- Análisis, diseño e implementación del sistema de información de Netmedia.

4.1. Análisis.	26
4.2. Diseño.	31
4.3. Implementación.	34

Capítulo 5.- El futuro del sistema.

Conclusiones.

Glosario.

Apéndices.

Bibliografía.

Diseño de sistemas de información para publicaciones de circulación gratuita controlada.

Parte I

Introducción y situación actual del mercado.

Capítulo 1.

Introducción

Hoy en día la labor del personal de mercadotecnia de las empresas es cada vez más compleja, ya que cada vez es más difícil que una empresa gaste en publicidad¹ y aún más si ésta no tiene alguna forma de medir si el dinero invertido ha sido bien aprovechado.

La publicidad es efectiva cuando el mensaje llega a la gente que puede comprar el servicio o producto que estamos ofreciendo y cuando existe una forma de medir el retorno de la inversión (ROI²). Por ejemplo, poco sirve enviar propaganda a un veterinario referente a un servidor³, o a un ingeniero en sistemas sobre medicinas para animales, ya que en teoría ninguno de los dos compraría ese tipo de productos; lo ideal sería que la propaganda que se refiere al servidor se le enviara al ingeniero y la de las medicinas al veterinario, pero ¿cómo puede la gente de mercadotecnia encaminar su publicidad al lugar correcto?, la respuesta es sencilla: disponiendo de *bases de datos* con el perfil adecuado. Partiendo de la misma definición de publicidad, hay que buscar (o crear) un público objetivo; si alguien tuviera una base de datos con información acerca de los ingenieros de sistemas, o bien, de gente que pudiera interesarle comprar en específico un servidor, el desperdicio de la publicidad sería prácticamente nulo, pues de antemano sabemos que esa gente tiene afición por nuestro producto y el mensaje sería recibido por la gente adecuada.

Existen muchas formas de llegar a un posible comprador (televisión, radio, revistas, periódicos, etc.), en particular en el mundo de las revistas, existe un nicho que es conocido como el de las **publicaciones de circulación controlada**; se dice que es controlada porque se escoge a la gente que va a recibir la publicación, en otras palabras, escogemos a la gente que va a formar parte de la base de datos. Aparentemente al regalar una publicación, los ingresos para la empresa se basarían estrictamente en publicidad impresa (anuncios en la publicación), pero al tener conocimiento del tipo de gente que leerá éstas publicaciones, podemos entonces ofrecerle al anunciante el perfil adecuado del producto que desea promocionar. Lo anterior a través de una base de datos en la cual se encuentran los potenciales clientes de cierta publicidad, la cual se encuentra disponible para su explotación y actualización de los mismos por medio de las consultas⁴ que permitan obtener la información necesaria o requerida de algún tipo específico de personas.

Al contar ya con una base de datos en la cual conocemos a detalle a sus integrantes surgen muchas oportunidades de negocio, por ejemplo, campañas de publicidad específicas para cierto perfil de lectores, realización de eventos, estudios de mercado, citas "FACE to FACE"⁵, envíos de algún artículo a algún segmento de la base de datos, etc. Para poder lograrlo se requiere que la información se encuentre reunida en una base de datos y que ésta se encuentre normalizada, es decir, hay que tener perfectamente definido el tipo de información que será almacenada en la base de datos para así obtener las respuestas apropiadas a nuestras necesidades de información.

1. Publicidad: "Es el proceso de comunicación que se enfoca a crear en un público predeterminado, la curiosidad activa por usar (o seguir usando) un producto o servicio, a través de mensajes en forma planeada y sistematizada, para ser emitidos a través de medios e instrumentos de comunicación masiva". [1]

2. ROI "Return of investment" *Ganancia sobre la inversión*. Es la ganancia que produce una inversión en un periodo de tiempo.

3. Servidor: Es una computadora o dispositivo en una red que maneja recursos de red. Dichos recursos pueden ser archivos, impresoras, servicios de correo electrónico, etc.

4. "Query" *Consultas*: Solicitud de información a una base de datos

5. Cita "FACE to FACE" *Cara a cara*: Son aquellas en las que se está en presencia de cierta persona en particular.

Es muy importante conocer la forma en que se construirá la base de datos, la cual permitirá explotar la información disponible para así utilizarla bajo un esquema de revistas de circulación controlada, sea cual sea el tema que manejen dichas publicaciones; éste es el caso real de una casa editorial llamada Netmedia, su modelo de negocio es muy similar al de la editorial estadounidense CMP (Computer Media Publications) United Business Media, en dicha editorial se publican una gran cantidad de revistas de circulación controlada que tratan temas de tecnología aplicada a negocios (entre ellas "InformationWeek" una revista que expone temas relacionados con la aplicación de la tecnología a los negocios y que lleva a la fecha más de 1,000 números). La idea es simple, tener información detallada de cada suscriptor, para saber qué tipo de producto podemos venderle. Para ello es necesario conocer el tipo de sistemas y tecnologías que están instaladas o se piensan instalar en la empresa, el nivel de decisión en la adquisición de dichos productos, el tipo de inversión de la empresa en IT⁶ y algunos datos más que conforman nuestro formato de suscripción.

Netmedia es una casa editorial cuyo compromiso es ofrecer una alternativa confiable, objetiva y del más alto nivel periodístico para comunicar temas relacionados con las tecnologías de información, a través de medios diversos (impresos, eventos y en línea), donde compradores y vendedores puedan entrar en contacto. A lo largo de su existencia Netmedia ha ido evolucionando acorde a las necesidades del mercado, ha diversificado sus productos partiendo inicialmente sólo de medios impresos y en la actualidad llegando al lector por otros medios como lo es en línea y eventos.

La contribución del presente trabajo es describir (entre otras cosas), una manera adecuada en que podemos construir una base de datos y el sistema de información para publicaciones de circulación controlada, y aunque en este caso es para publicaciones de IT, puede aplicarse a cualquier publicación en esta modalidad y a su vez, se puede aplicar a cualquier negocio que maneje formularios con información detallada de los miembros que componen su base de datos.

El problema a resolver es poder controlar toda la información que obtengamos de los suscriptores para contar con una base de datos que pueda ser segmentada de acuerdo a las necesidades de nuestros anunciantes, por lo que al organizarlo, explotarlo y mantenerlo por medio de un sistema de información junto con bases de datos debidamente normalizadas, será mucho más sencillo, eficiente y sin duplicación de información. Asimismo, todos los procesos de captura y actualización de los datos de los suscriptores tendrán un tiempo de respuesta corto y se simplificarán significativamente todas las tareas propias para mantener actualizada un base de datos, gracias a las herramientas que nos brinda un sistema de administración de bases de datos en conjunto con los demás componentes del sistema de información, dicho sistema empezó a operar desde febrero del 2003 y hasta la fecha ha cubierto las necesidades para las que fue diseñado en un principio y ha sido capaz de adaptarse a los cambios que exige la empresa.

En éste trabajo se describirá el sistema de información que ha apoyado durante 5 años a Netmedia, la cual gracias al sistema de información ha obtenido los siguientes resultados a lo largo de dichos años:

- Incremento en la base de datos de lectores de 5,000 iniciales a poco más de 40,000 en la actualidad (un incremento del 800 %).
- El sistema de información ha sido un factor clave para que la empresa crezca, al contar con un sistema de información estable, confiable y que permite filtrar la información de cada registro a detalle, se han podido lanzar productos a la medida de los anunciantes y con ello ha aumentado la participación de mercado progresivamente, comenzó con un 6 % en el año 2,000 y actualmente es del 12 %.

6. "Information Technology", *Tecnologías de información*. Son los aspectos concernientes al manejo y procesamiento de la información.

- En diversas ocasiones se ha usado la información contenida en el sistema para la toma de decisiones críticas en la empresa, por ejemplo, viabilidad de proyectos, lanzamiento de nuevos productos, cambios en los existentes, etc.
- El sistema de información permite llevar un monitoreo sobre los servicios de mensajería, con ello se han reducido los tiempos en las entregas (partimos de 15 días hábiles y en la actualidad dicho proceso se redujo a 10 días hábiles).
- Al momento de empezar el diseño del sistema de información se revisaron los procesos involucrados y en la mayoría de los casos se hicieron cambios para volverlos más eficaces, con lo que la empresa se ha vuelto más productiva gracias a la revisión de procesos para el desarrollo del sistema de información.

Capítulo 2

Situación actual del mercado

En México no existía hasta hace algunos años, una base de datos con un nivel de detalle profundo en cuanto a potenciales clientes para tecnologías de la información, cada empresa tenía su propia base de datos, este panorama presentaba principalmente dos desventajas:

***La actualización de los datos.**- Los datos no se encontraban actualizados, por lo que la información se encontraba errónea o repetida y por lo tanto era poco confiable y actual.

***No se conocía a detalle al cliente.**- Los datos registrados en las bases de datos no contenían toda la información necesaria para explotarlos y así conocer a cada uno de los clientes registrados, ya que en su momento fueron suscritos por diversos motivos, es decir, algunos fueron registrados solo porque compraron algún producto o porque tenían cierto puesto en alguna empresa y otros solo porque alguien los recomendó, pero no se conocía su perfil exacto, la pregunta en ese momento a responder era: ¿en realidad eran clientes potenciales para algún producto específico? En ese momento no lo sabíamos pues probablemente compraron algún tipo de tecnologías de información una única vez (tal vez computadoras o algún dispositivo para las mismas) pero a ciencia cierta no se conocía el tipo de productos que serían de mayor interés a ese cliente.

Bajo estas necesidades surgió en el mercado la empresa llamada Netmedia (como licenciataria⁷) de CMP, cuyos objetivos son: recopilar datos de aspirantes a suscriptores a través de formularios de suscripción (impresos o en nuestros sitios de red), donde se registran desde datos generales para poder hacer un envío físico (dirección, código postal, etc.) hasta datos muy específicos como lo son los hábitos de compra y los niveles de decisión. Éstos formatos se evalúan y de acuerdo a ciertos criterios de evaluación se decide si se regala(n) o no, alguna(s) de las publicaciones. En éste caso nuestro público objetivo es la gente encargada de tomar decisiones en la adquisición de tecnologías de información, tanto el que recomienda e instala (la gente encargada de los sistemas), la persona que decide el presupuesto (gente de finanzas y administración), hasta la persona que da la autorización final (directores generales). Bajo éste perfil definido, el anunciante puede tener la certeza del tipo de gente que va a recibir su anuncio.

Así surgen varias oportunidades de negocio para la editorial (no sólo los anuncios impresos), como:

- Envío de encartes o correos directos, dependiendo de ciertos perfiles.
- Servicios de mercadotecnia vía correo electrónico.
- Realización de eventos patrocinados o exclusivos para los anunciantes.

7.Licenciataria: Empresa que tiene permiso de hacer uso de facultades estipuladas de antemano (en este caso a través de un contrato). [1]

- Con las personas que aparecen en la portada de nuestra revista se forma un club, el cual tiene a la gente más importante de sistemas en México, formado principalmente por directores en sistemas de las empresas. Se organiza junto con el patrocinador una reunión con todo este club, el patrocinador tiene la oportunidad de platicar con aquellos que deciden enormes presupuestos de tecnologías de información, hacer demostraciones de sus productos y finalmente tener la posibilidad de cerrar una venta importante.

Con esto se resuelve el problema de no conocer a detalle a la gente que compone la base de datos, pero aún nos falta responder la pregunta: ¿cómo se mantiene actualizada la información? Al ser una revista quincenal cuyo envío se hace mediante el correo o mensajerías, conocemos de manera precisa el estado de nuestros lectores, esto significa que sabemos cualquier cambio que éste haya o no realizado, (su dirección, teléfono, etc.), pues al tener reportes de devolución se verifican dichos cambios y se hacen las correcciones o actualizaciones necesarias en la base de datos.

Diseño de sistemas de información para publicaciones de circulación gratuita controlada.

Parte II

Diseño y justificación del sistema de información.

Capítulo 3

Diseño y justificación del sistema de información

Para poder hacer todas las tareas de las que hablamos en la Sección I, necesitamos buscar la forma de obtener información y de poder explotarla adecuadamente. De aquí surge la necesidad de contar con un sistema de Información. En este capítulo se describen las tecnologías que sustentan a la última versión del sistema de información, en el siguiente se explica el proceso de implementación.

Un sistema de información es un conjunto de elementos y procedimientos que interactúan entre sí para adquirir y almacenar datos, que posteriormente serán convertidos en información para usarla con algún objetivo en particular (como la toma de decisiones en las empresas).

Hablando de sistemas de información basados en computadoras, se integran por: hardware, software, sistemas de administración de base de datos, redes, procedimientos y usuarios de cada componente. Veamos a continuación cada uno de dichos componentes.

3.1. El Hardware

Entendamos por hardware todos los componentes físicos de una computadora, incluyendo cualquier equipo periférico como impresoras y teclados; en otras palabras, se refiere a todos los componentes que se pueden tocar de dicho sistema [2].

Para implementar un sistema de información es vital contar con una infraestructura confiable y capaz de hacer las tareas requeridas. Dependiendo de la carga de trabajo que esperemos, tenemos que evaluar qué tipo de equipos se acoplan mejor a nuestras necesidades. Asimismo debe existir un balance entre **costo-desempeño**, es decir, tal vez nos gustaría adquirir un servidor con múltiples procesadores, o que se le puedan conectar varios discos duros, sin embargo el sistema que desarrollemos probablemente no demandará tantos recursos del servidor y por ende, incurriríamos en gastos innecesarios. Lo mismo pasa para los demás componentes que integren el sistema, por ejemplo, si las estaciones de trabajo fueran los modelos más recientes, nos preguntaríamos ¿en verdad necesitamos que las estaciones de trabajo tengan un procesador tan rápido o un monitor de plasma? Veamos ahora la otra cara de la moneda, si llega algún proveedor y nos ofrece estaciones de trabajo muy baratas, lo más probable es que no cubran los requerimientos mínimos del sistema (y demás aplicaciones que en ellas se ejecuten), por lo que en poco tiempo se volverán obsoletas y se tendrá que hacer un nuevo gasto para conseguir equipo que sí cumpla con los requerimientos del sistema. En resumen, hay que hacer un balance entre el desempeño requerido y el precio, por supuesto dicha tarea no es sencilla y el desempeño óptimo del sistema dependerá en gran parte de las elecciones que se hagan sobre el hardware.

3.2. El Software

Son datos o instrucciones de computadora, en general se refiere a todo lo que pueda ser almacenado electrónicamente [2]. Haciendo una analogía con un libro, la cubierta, las páginas y la tinta serían el hardware y las palabras, imágenes y en general todo el contenido impreso del libro serían el software. Podemos dividir el software en 2 categorías principales:

1.- El software del sistema: Consiste en programas de bajo nivel que interactúan con la computadora en forma muy básica. Incluye al sistema operativo¹, compiladores² y demás utilerías (programas) que permiten manejar los recursos de la computadora.

2.- El software de aplicaciones: Que son los programas con los que trabajará el usuario final³, un ejemplo de éstos son los procesadores de texto o las hojas de cálculo. Este tipo de software se ubica en una capa superior del sistema, ya que no puede operar sin el sistema operativo y las utilerías del mismo. Al igual que con el hardware (y en general con todos los componentes del sistema de información) hay que escoger bien el software que vamos a usar de acuerdo a nuestras necesidades. Necesitamos responder a preguntas como: ¿el sistema operativo que tenemos en mente es compatible con las aplicaciones que queremos usar?, ¿cuánto dinero tenemos disponible para invertir en software?, ¿nos conviene usar un software libre como GNU/Linux?, ¿los demás componentes del sistema de información son compatibles con éste sistema operativo?, entre otras.

3.3. Sistemas operativos

Entendamos por sistema operativo un programa que actúa como intermediario entre el usuario final y el hardware de la computadora. Su propósito es proveer un medio en el cual el usuario pueda ejecutar programas de manera conveniente y eficiente. El sistema operativo debe asegurarse de la correcta operación del sistema, para prevenir que los programas de los usuarios interfieran con el funcionamiento propio del mismo, el hardware debe proveer mecanismos adecuados.

Los sistemas operativos proveen algunos servicios a los programas y a sus usuarios para hacer sus tareas más sencillas, difieren de un sistema operativo a otro pero tienen varios servicios en común. Es más fácil definir un sistema operativo por lo que hace, que por lo que es. El principal objetivo de algunos sistemas operativos es que sean convenientes para el usuario, existen porque están diseñados para hacer más fácil el uso de la computadora con ellos, que sin ellos. Este punto de vista es claro cuando pensamos en sistemas operativos para computadoras pequeñas o caseras. Por otro lado hay sistemas operativos cuya principal prioridad es la eficiencia, este es el caso de sistemas grandes, compartidos para múltiples usuarios. Estos dos objetivos **–eficiencia y conveniencia–** a menudo son contradictorios.

1. "SO" "Operating system" *Sistema operativo*: Es un software que controla la ejecución de programas de computadora. Una colección organizada de rutinas y procedimientos para operar la computadora. Tiene 4 tareas principales: (1) calendarizar, cargar, iniciar y supervisar la ejecución de programas (2) asignación de almacenamiento (3) iniciar y controlar las operaciones de entrada y salida y (4) manejo de errores. [2]

2. "Compiler" *Compilador*: Es un programa que traduce código fuente en instrucciones que puede ejecutar la computadora [5]

3. "End user" *Usuario final*: Es la persona que usa computadoras y aplicaciones de computadora

En el pasado la eficiencia era más importante que la conveniencia, sin embargo los sistemas operativos han ido evolucionando a lo largo del tiempo, por ejemplo UNIX⁴ empezó sólo con un teclado y una impresora en su interfaz, poniendo una gran limitante en cuanto a conveniencia para el usuario, al paso del tiempo el hardware cambió y fue posible usar interfaces más amigables que usaban ya un ratón (mouse) e incluso se agregaron interfaces de usuario gráficas (GUIs)⁵ haciendo a UNIX más manejable para los usuarios finales sin dejar de estar concentrado en la eficiencia. [12]

Para nuestro caso en particular decidimos usar como sistemas operativos Windows 2003 para la parte de la base de datos y Linux para nuestro servidor web, veamos a continuación características de cada uno de ellos.

LINUX

LINUX parece y se siente como cualquier otro sistema basado en UNIX, aún más, la compatibilidad con UNIX ha sido uno de sus principales objetivos en su diseño. Sin embargo, LINUX es mucho más joven que la mayoría de los sistemas UNIX. Su desarrollo empezó en 1991, cuando Linus Torvalds escribió y bautizó a LINUX, con un pequeño pero auto-contenido núcleo (kernel)⁶ para procesadores 80386, el primer procesador verdadero para procesamiento a 32 bits.

Al comienzo de su desarrollo, el código fuente de LINUX estaba disponible gratuitamente en Internet, de ahí que la historia de LINUX ha sido una colaboración de muchos usuarios alrededor del mundo. En sus inicios, el desarrollo de LINUX giraba alrededor del núcleo central del sistema operativo. Sin embargo se necesitaba mucho más que este núcleo para producir un sistema operativo completo.

Un sistema básico de LINUX es un ambiente estándar para aplicaciones y ambientes de programación para los usuarios, pero no fuerza a ningún estándar lo que significa que su funcionalidad se conserva. LINUX ha ido madurando, han surgido necesidades para agregar otra capa de funcionalidad en la parte más alta del sistema. Una distribución de LINUX incluye todos los componentes estándar del sistema, además de un conjunto de herramientas administrativas que simplifican la instalación inicial, actualizaciones posteriores e instalaciones y actualizaciones de otros paquetes del sistema. Una distribución moderna también contiene herramientas para administración de sistemas de archivos, creación y administración de cuentas de usuarios, administración de redes, entre otras. En teoría, cualquiera puede instalar LINUX descargándolo de los sitios ftp⁷ y compilándolo. Sin embargo esta operación era una tarea compleja que requería conocimientos específicos de LINUX y que no cualquiera podía llevar a cabo. LINUX maduró y varios individuos y grupos han intentado hacer que esta tarea sea sencilla y han desarrollado paquetes precompilados que prometen una instalación sencilla.

4. UNIX: Es un sistema operativo desarrollado por AT&T y Bell multiusuario y multitareas.

5. GUI "Graphical user interface" *Interfaz gráfica de usuario*: Es una interfaz que representa programas, archivos y opciones a través de imágenes. Estas imágenes pueden ser íconos, menús y cuadros de diálogo, entre otros.

6. Kernel: Es el módulo central de un sistema operativo. Es la parte del sistema operativo que se carga primero y que pertenece en la memoria principal. Como debe permanecer en memoria, es importante que el kernel sea pequeño pero capaz de proveer los servicios esenciales requeridos por otros componentes del sistema operativo y aplicaciones. Típicamente, el kernel es responsable del manejo de la memoria, de procesos, de tareas administrativas y del manejo del disco.

7. FTP "File transfer protocol" *Protocolo de transferencia de archivos*: Es un protocolo de aplicación que forma parte de la pila de protocolos TCP/IP, que se utiliza para la transferencia de archivos entre los nodos de la red. FTP está definido en el RFC 959

Estas colecciones o distribuciones, incluyen mucho más que sólo el sistema básico LINUX. Típicamente incluyen utilerías extras al sistema, así como paquetes precompilados y listos para instalar muchas herramientas comunes en UNIX, como servidores web, de correo, editores de texto, etc. Un gran número de distribuciones comerciales y no comerciales están disponibles hoy día. Red Hat (la que elegimos aquí) es una distribución con soporte comercial y Debian es gratuito para la comunidad de LINUX.

Componentes de un sistema LINUX

Un sistema LINUX se compone de tres módulos principales de código, como la mayoría de las implementaciones tradicionales de UNIX:

- 1.- Núcleo: El núcleo es el responsable de mantener todas las abstracciones importantes del sistema operativo, incluyendo aspectos como el manejo de memoria virtual y de procesos.
- 2.- Bibliotecas del sistema: Las bibliotecas del sistema definen un conjunto de instrucciones estándar por medio de las cuales las aplicaciones pueden interactuar con el núcleo, e implementan muchas funcionalidades del sistema operativo que no necesitan los privilegios completos del código del kernel.
- 3.- Utilerías de sistema: Son programas que llevan a cabo tareas de administración individuales especializadas. Algunas utilerías de sistema son utilizadas sólo una vez para inicializar y configurar algún aspecto del sistema; otras —conocidas como *demonios* en terminología de UNIX— pueden ejecutarse permanentemente, manejando tareas como respuestas a conexiones de red, actualizando archivos de registro, etc.

LINUX hereda de UNIX una gran estabilidad y un excelente uso de los recursos del sistema, eso lo hace especialmente bueno para aplicaciones como bases de datos y servidores web.

Windows 2003

A mediados de la década de los ochentas Microsoft e IBM cooperaron conjuntamente para desarrollar el sistema operativo llamado OS/2, el cual fue escrito en lenguaje ensamblador para sistemas con procesadores individuales Intel 80286. En 1988 Microsoft decidió empezar de nuevo y desarrollar una "Nueva Tecnología" (NT). Originalmente el equipo planeó para el desarrollo de NT usar OS/2 como su medio nativo, pero durante su desarrollo Windows NT fue cambiado para usar el API⁸ de 32 bits de Windows (WIN32 API)⁹

Los objetivos de diseño que Microsoft estableció para Windows 2000 (y que se conservan para su versión 2003), incluyen:

Extensibilidad: Se refiere a la capacidad del sistema operativo de mantenerse al día con los avances en la tecnología.

Portabilidad: Se dice que un sistema operativo es portable si puede ser movido de una arquitectura de hardware a otra con relativamente pocos cambios, Windows 2000 está diseñado para ser portable.

8. API "application programming interface" Interfaz de programación de aplicaciones: Un API es cualquier interfaz que permite a un programa usar las facilidades provistas por otro.

9. WIN32 API: Es la interfaz fundamental de programación para Windows, este API interactúa con los objetos del kernel, comparte objetos entre procesos, administra procesos, intercomunica procesos y asigna memoria.

Confiabilidad: Es la habilidad de manejar condiciones de error, incluyendo la habilidad del sistema operativo de protegerse asimismo y a sus usuarios de software con posibles errores o corrupciones. Windows 2000 resiste defectos y ataques usando protección de hardware para la memoria física y mecanismos de protección de software para los recursos del sistema operativo. Además Windows 2000 viene con un sistema de archivos nativo (NTFS)¹⁰ que se recupera automáticamente de una gran variedad de errores de archivo cuando el sistema falla.

Compatibilidad: Windows 2000 provee compatibilidad a nivel de código con aplicaciones que siguen el estándar IEEE 1003.1 (POSIX, un estándar que define portabilidad de código en aplicaciones)¹¹. Así, pueden ser compiladas para correr en Windows 2000 sin necesidad de cambios en su código fuente.

Rendimiento: Está diseñado para ofrecer un buen rendimiento. Ya que los subsistemas que lo constituyen pueden comunicarse entre sí eficientemente por medio de LPCs (Local Procedure Calls) lo que provee un alto rendimiento en el intercambio de mensajes.

Soporte internacional: Está diseñado para uso internacional. Provee soporte para diferentes configuraciones locales, cuenta con rutinas para dar formato a fechas, tiempo y unidades monetarias de acuerdo a los requerimientos de cada país.

Componentes de un sistema Windows 2000

La arquitectura de Windows 2000 es un sistema en capas de módulos interconectados. Los principales componentes son:

El HAL (Hardware-Abstraction Layer): Es la capa de software que esconde las diferencias del hardware a niveles más altos del sistema operativo para ayudar a Windows 2000 a ser portable.

El núcleo: El núcleo de Windows 2000 provee los fundamentos para el ejecutivo (una colección de componentes del núcleo de sistemas basados en arquitecturas NT) y los subsistemas.

El Ejecutivo: Provee un conjunto de servicios que todos los subsistemas del medio pueden usar, dichos servicios se agrupan en: administrador de objetos, administrador de memoria virtual, administrador de procesos, facilidades de llamadas a procedimientos locales (RPC), administrador de entrada/salida (I/O) y el monitor de referencias de seguridad.

Hablando propiamente de Windows 2003, las tareas de administración de usuarios y de edición de los permisos sobre los recursos de la red, se hacen más sencillas en esta versión que en su antecesora (Windows 2000), al instalar éste sistema prácticamente no carga características que pudieran no ser necesarias y que podrían afectar el desempeño del equipo, asignando recursos a servicios no utilizados.

3.4. Sistemas de administración de base de datos "Database Management System" (DBMS)

Es un conjunto de programas de computadora que permiten guardar, modificar, asegurar y extraer información de una base de datos [2]. El objetivo principal de un DBMS es proveer una manera de almacenar y consultar información en la base de datos tanto de manera conveniente como eficiente.

10. NTFS "New Technology File System" *Nueva tecnología de sistema de archivos*. es un sistema de archivos diseñado específicamente para Windows NT, con el objetivo de crear un sistema de archivos eficiente, robusto y con seguridad incorporada desde su base. También soporta compresión nativa de ficheros y encriptación (esto último sólo a partir de Windows 2000).

11. IEEE 1003.1 (POSIX) "Portable operating system interface" *Interface de sistema operativo portable* la definición de estándar se encuentra en: <http://www.ieee.org>

Los sistemas de bases de datos están diseñados para manejar grandes cantidades de información. El manejo de los datos involucra definir estructuras para guardar la información y proveer mecanismos para poder manipularla.

Además, el sistema de bases de datos debe garantizar que la información almacenada este a salvo a pesar de que el sistema pudiera fallar e impedir accesos no autorizados. Si los datos fueran compartidos entre varios usuarios, el sistema debe estar preparado para evitar resultados incorrectos.

Por otro lado, existen varios modelos de datos, una colección de herramientas conceptuales para describir la información, sus relaciones, semántica y condiciones de consistencia en otras palabras, un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas; son abstracciones que permiten la implementación de un sistema eficiente de base de datos, por lo general se refieren a algoritmos y conceptos matemáticos.

Algunos modelos utilizados con frecuencia en las bases de datos son:

Bases de datos de red: Este permite que un mismo nodo tenga varios padres ofrece una solución eficiente al problema de redundancia de datos, pero aún así, la dificultad que significa administrar la información en una base de datos de red, ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

Bases de datos relacionales: Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM, no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de tablas compuestas de registros y campos.

En éste modelo, el lugar y la forma en que se almacenan los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la ventaja considerable de que es más fácil de entender y de utilizar para un usuario casual de la base de datos. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

El modelo relacional es hoy en día el principal modelo de datos para aplicaciones comerciales de procesamiento de datos. Se ha ganado esta posición principal gracias a su simpleza, ya que facilita el trabajo de los programadores comparada con los modelos anteriores como el jerárquico o el de red. Como decíamos anteriormente, un modelo de bases de datos relacional consiste de un conjunto de tablas cada una con un nombre único. Una fila de una tabla representa una relación entre un conjunto de valores. Dado que una tabla es una colección de dichas relaciones, hay una estrecha correspondencia entre el concepto de tabla y el concepto matemático de relación, y es precisamente de ahí de donde éste modelo toma su nombre.[14]

El lenguaje más común para construir las consultas a bases de datos relacionales es SQL [G2], un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. [8]

Bases de datos orientadas a objetos: En éste modelo los datos se almacenan como objetos completos y deben satisfacer dos criterios, primero: debe ser un sistema de administración de bases de datos con persistencia, administración de almacenamiento, coincidencia, recuperación y herramientas adecuadas de consulta. Segundo, debe ser un sistema orientado a objetos con identidad de objeto, encapsulación, tipos y clases, herencia, que sobre escriba combinaciones y sea extensible. [9]

Bases de datos objetos relacionales: En este modelo la información sigue almacenándose en tablas, pero algunas de las entradas en las tablas pueden tener estructuras complejas de datos, a dichas estructuras se les conoce como *tipo de datos abstractos* (ADT). Un ADT es un tipo de dato que es construido combinando tipos de datos alfanuméricos básicos. El soporte para ADT es atractivo por que las operaciones y funciones asociadas con este nuevo tipo de datos pueden ser usadas para Indexar, guardar y solicitar registros en el nuevo contenido ADT (por ejemplo multimedia). Al usar una base de datos de objetos relacionales se dispone de más estructuras para modelar datos más complejos. Este modelo ofrece soporte limitado para relaciones, identidad, herencia, polimorfismo, composición, extensibilidad o creación de clases definidas por el usuario e integración con lenguajes orientados a objetos como Java [24].

Sistemas de bases de datos de diferentes modelos están disponibles en el mercado, y es tarea del diseñador de la base de datos conocer cuál es el que se acopla mejor a sus necesidades.

Almacenar datos en una base de datos sólo es útil cuando somos capaces de extraerlos en el momento que los necesitamos y convertirlos en información. Algunas veces (cuando se hace un catálogo de productos, por ejemplo), todo lo que se necesita es una lista con los datos necesarios. Sin embargo, usualmente necesitaremos un conjunto de datos que respondan alguna pregunta en particular, por ejemplo ¿cuántos registros tenemos en el estado de Durango? o ¿cuántas mujeres entre 30 y 35 años hay en la base de datos? Existen herramientas para realizar consultas y reportes que ayuden a presentar información de la base de datos, ya sea en la pantalla o en papel.

Al elegir un sistema de administración de base de datos tenemos que decidimos por uno que sea capaz de soportar el número de usuarios que esperamos, que no tenga problemas para manejar la cantidad de información esperada y que sea flexible de acuerdo al incremento de información que se presente y de las modificaciones que se pudieran presentar.

También debemos decidir la manera en que se manejará el acceso a la información, ya que es muy peligroso tener un sistema en el que todos los usuarios puedan ver, modificar, borrar o incluso hacer cambios sobre el sistema de administración de base de datos, tenemos que hacer un análisis de necesidades para conocer y decidir la forma en que se otorgarán los permisos mínimos necesarios para que cada usuario pueda ejecutar sus tareas sin poner en riesgo la información, el mal uso que pudiera hacerse de ella o un mal funcionamiento del sistema. Dicho esquema de permisos puede otorgarse desde un directorio en un servidor que haga la validación de nombres de usuario, contraseñas y administre los permisos de recursos o con permisos otorgados dentro del propio manejador de base de datos.

Uno de los sistemas de administración de bases de datos más populares en la actualidad es SQL Server. Es un sistema de administración de bases de datos relacional originalmente diseñado en los 80's por SYBASE para sistemas UNIX y posteriormente adoptado para sistemas NT por Microsoft. Desde 1994 Microsoft ha entregado versiones de SQL Server independientemente de Sybase. La última versión disponible es la 2003 disponible en varias versiones y lenguajes, aunque para efectos de esta tesis trabajaremos con SQL 2000.

El servidor de SQL provee servicios de replicación sobre múltiples copias del mismo y también sobre otros sistemas de bases de datos. Sus servicios de análisis, una parte integral del sistema, incluyen procesamiento analítico en línea (OLAP) y facilidades para minería de datos. Cuenta con una variada colección de herramientas gráficas y de asistentes que guían a los administradores de bases de datos en ciertas tareas como configurar respaldos, replicaciones entre servidores, parámetros para fijar rendimiento, etc.

Provee un conjunto de herramientas para administrar todos los aspectos de desarrollo, consultas, depuración, pruebas y administración. La mayoría de estas herramientas se encuentran en el administrador empresarial de SQL Server (Enterprise Manager), un accesorio complementario de la consola de administración la cual brinda una interfaz común para trabajar con varias aplicaciones de servidor en una red. [14]

En resumen, SQL Server es un paquete completo para administración de bases de datos con opciones para importar y exportar datos en XML, un servidor de análisis, clientes para minería de datos, replicación entre almacenes de datos heterogéneos.

3.5. Lenguaje de programación

Los lenguajes de programación ayudan al usuario a comunicarse con la computadora. La computadora puede entender directamente algunos de ellos, pero otros requieren de pasos de traducción intermedios. Hoy día se utilizan cientos de lenguajes de computadora, los cuales pueden dividirse en tres tipos generales:

- Lenguaje de máquina.
- Lenguaje de ensamblador.
- Lenguajes de alto nivel.

Una computadora sólo puede entender directamente su propio lenguaje de máquina. El lenguaje de máquina es el "lenguaje natural" de una computadora en particular, y está definido por el diseño de hardware de esa computadora. Los lenguajes de máquina generalmente consisten en cadenas de números (que en última instancia se reducen a unos y ceros) que ordenan a la computadora realizar sus operaciones más elementales una por una. Los lenguajes de máquina son dependientes de la máquina; es decir, un lenguaje de máquina en particular sólo puede usarse en un tipo de computadora.

Los lenguajes de máquina son difíciles de usar para las personas, conforme las computadoras se volvieron más populares, se hizo evidente que la programación en lenguaje de máquina era sencillamente demasiado lenta y tediosa para la mayoría de los programadores. En lugar de usar las cadenas de números que las computadoras podían entender directamente, los programadores comenzaron a utilizar abreviaturas de palabras en inglés para representar las operaciones elementales de la computadora. Estas abreviaturas constituyeron el fundamento del lenguaje **ensamblador**.

El uso de computadoras aumentó rápidamente con la aparición de los lenguajes de ensamblador, pero éstos todavía requerían muchas instrucciones para llevar a cabo incluso las tareas más sencillas. A fin de acelerar el proceso de programación, se desarrollaron lenguajes de alto nivel en los que se podía escribir un solo enunciado para realizar tareas sustanciales. Los programas traductores que convierten los programas escritos del lenguaje de alto nivel a lenguaje de máquina se denominan compiladores. Los lenguajes de alto nivel permiten a los programadores escribir instrucciones que semejan el inglés cotidiano y contienen notaciones matemáticas de uso común [17].

JAVA y JAVASCRIPT

En 1991 Sun Microsystems financió un proyecto de investigación cuyo resultado fue la creación de un lenguaje basado en C y C++ que su creador, James Gosling, llamó Oak. Más tarde vieron que ya existía un lenguaje de programación llamado así y decidieron cambiarle el nombre a Java [17]. Java está diseñado para satisfacer necesidades de desarrollo en redes de ambiente heterogéneo. A su vez busca que sus aplicaciones consuman el mínimo de recursos del sistema, puedan ejecutarse en cualquier plataforma de hardware y software [18].

Java es un lenguaje de programación de alto nivel que tiene como principales características el ser simple, orientado a objetos, distribuido, interpretado, robusto, seguro, con arquitectura neutral, portable, ofrece un gran rendimiento y es dinámico (explicaremos a continuación brevemente dichas características. [19]

Una de las principales características de Java es que es un lenguaje simple que puede ser programado sin un entrenamiento extensivo. Los conceptos fundamentales de la tecnología Java son aprendidos rápidamente, los programadores pueden ser productivos desde el principio.

Java está diseñado para ser orientado a objetos, la tecnología Java provee una plataforma sencilla y eficiente para desarrollo basado en objetos. Java se ve como C ++ lo que da como resultado un lenguaje *familiar* con la ventaja de que elimina las complicaciones innecesarias de C ++ haciendo que Java retenga muchas de las características orientadas a objetos y "se vea y se sienta" como C ++ con lo que éstos programadores pueden migrar fácilmente a la plataforma Java y ser productivos rápidamente.

Java está diseñado para crear aplicaciones muy confiables, el manejo de la memoria es extremadamente simple: los objetos son creados como un nuevo operador. No hay tipos de datos de apuntadores definidos explícitamente por los programadores, no hay aritmética de apuntadores y la recolección de basura es automática. Está diseñado para operar en ambientes distribuidos, lo que implica que la seguridad es de vital importancia, permite construir aplicaciones que no puedan ser invadidas desde afuera. En un ambiente de red, las aplicaciones escritas en Java están a salvo de que algún código no autorizado intente contaminar o invadir los archivos del sistema.

Está diseñado para soportar aplicaciones heterogéneas que serán usadas en ambientes de red. En dichos ambientes, las aplicaciones deben ser capaces de ejecutarse en una gran variedad de arquitecturas de hardware. Para acomodarse a la gran diversidad de ambientes operativos el compilador de Java genera código en un formato intermedio de *arquitectura neutral* diseñada para transportar eficientemente a diversas plataformas de hardware y software. La plataforma de arquitectura neutral y portable de la tecnología Java se conoce como máquina virtual de Java.

JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador en el web para decidir mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Toda esta potencia se pone a disposición del programador, quien se convierte en el verdadero dueño y controlador de cada cosa que ocurre en la página.

JAVASCRIPT y HTML

En el web hay una cantidad desmesurada de textos (hipertexto), que contienen enlaces que relacionan cada una de las unidades básicas donde podemos encontrar información, las páginas web. En un principio, para diseñar este sistema de páginas con enlaces se pensó en un lenguaje que permitiera presentar cada una de éstas junto con unos pequeños estilos, este lenguaje fue HTML.

Cabe mencionar que HTML no es suficiente para realizar todas las acciones que se pueden llegar a necesitar en una página web. Esto es debido a que conforme fue creciendo el web y sus distintos usos se fueron complicando las páginas y las acciones que se querían realizar a través de ellas. El HTML se había quedado corto para definir todas estas nuevas funcionalidades, ya que sólo sirve para presentar el texto en un página, definir su estilo estático y poco más.

Uno de los primeros lenguajes que cubría las necesidades que estaban surgiendo fue Java, a través de la tecnología de los *Applets*¹², que son pequeños programas incrustados en las páginas web y que pueden realizar las acciones asociadas a los programas de propósito general. La programación de applets fue un gran avance y por aquel entonces el navegador más popular, había roto la primera barrera del HTML al hacer posible la programación dentro de las páginas web. No cabe duda que la aparición de los applets supuso un gran avance en la historia del web, pero no ha sido una tecnología definitiva y muchas otras han seguido implementando el camino que comenzó con ellos.

Netscape, después de hacer sus navegadores compatibles con los applets, comenzó a desarrollar un lenguaje de programación al que llamó LiveScript el cual permite crear pequeños programas en las páginas y además era mucho más sencillo de utilizar que Java. De modo que el primer Javascript se llamó LiveScript, pero no duró mucho ese nombre, pues antes de lanzar la primera versión del producto se forjó una alianza con Sun Microsystems, creador de Java, para desarrollar en conjunto ese nuevo lenguaje.

La alianza hizo que Javascript se diseñara como un hermano pequeño de Java, solamente útil dentro de las páginas web y mucho más fácil de utilizar, de modo que cualquier persona, sin conocimientos de programación pudiese adentrarse en el lenguaje y utilizarlo como mejor le conviniere. Además, para programar Javascript no es necesario un conjunto de herramientas de desarrollo, ni compilar los scripts, ni realizarlos en archivos externos al código HTML, como ocurría con los applets.

Netscape 2.0 fue el primer navegador que entendía Javascript y su estela fue seguida por los navegadores de la compañía Microsoft a partir de la versión 3.0.

12. Applet: Es un pequeño programa en Java que puede ser añadido al código de una página web. Los applets difieren de una aplicación completa en Java en que no tienen permiso de acceder a algunos recursos en la computadora, como archivos o dispositivos conectados por el puerto serial (módems, impresoras, etc.). La regla común de los applets es que sólo pueden hacer una conexión a Internet con la computadora a la cual fue enviado el applet.

Actualmente son productos totalmente distintos y no guardan entre sí más relación que la sintaxis idéntica y poco más. Algunas diferencias entre estos dos lenguajes son las siguientes:

- **Compilador.** Para programar en Java necesitamos un conjunto de herramientas de desarrollo y un compilador. Sin embargo, Javascript no es un lenguaje que necesite que sus programas se compilen, sino que éstos se interpretan por parte del navegador cuando éste lee la página. Recordemos que un programa que ha sido compilado puede correr por sí solo, pues en el proceso de compilación se transformó a lenguaje máquina. En cambio el intérprete traduce el programa cuando lo lee, convirtiendo el código del programa directamente en acciones. La ventaja del intérprete es que el programa puede interpretarse en cualquier plataforma, en cambio el archivo generado por el compilador sólo funciona en la plataforma en la que se ha creado. Por otro lado un archivo compilado puede ser distribuido fácilmente si se conoce la plataforma, mientras que un archivo interpretado no funciona si no se tiene el intérprete.
- **Orientado a objetos.** Java es un lenguaje de programación orientado a objetos. Javascript no hace un uso tan estricto a la orientación a objetos, esto quiere decir que podremos programar sin necesidad de crear clases, tal como se realiza en los lenguajes de programación estructurada como C o Pascal.
- **Propósito.** Java es mucho más potente que Javascript, esto es debido a que Java es un lenguaje de propósito general, con el que se pueden hacer aplicaciones de lo más variado, sin embargo, con Javascript sólo podemos escribir programas para que se ejecuten en páginas web.
- **Estructuras fuertes.** Java es un lenguaje de programación tipado. Por su parte Javascript no tiene esta característica y podemos asignar en una variable la información que deseemos, independientemente del tipo de ésta. Además, podemos cambiar el tipo de información de una variable cuando queramos.
- **Otras características.** Como vemos Java es mucho más complejo, aunque también más potente, robusto y seguro. Tiene más funcionalidades que Javascript y las diferencias que los separan son lo suficientemente importantes como para distinguirlos fácilmente.

Javascript es ideal para hacer validaciones en formularios que se encuentren en páginas web, y precisamente para eso lo usaremos más adelante.

PHP

PHP es uno de los lenguajes de lado del servidor más extendidos en la red. Nació en 1994, se trata de un lenguaje de creación relativamente reciente que ha tenido una gran aceptación en la comunidad de administradores de servidores debido a la potencia y simplicidad que lo caracterizan.

PHP nos permite añadir pequeños fragmentos de código dentro de una página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas desarrollados íntegramente en un lenguaje distinto a HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a los lenguajes pensados para realizar CGI⁹, PHP ofrece un sin fin de funciones para la explotación de bases de datos de una manera llana, sin complicaciones, las cuales se expondrán más adelante.

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page

Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiera utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico, se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez -gracias a que primero se compila y luego se ejecuta, mientras que antes se interpretaba el código-, su mayor independencia del servidor web -creando versiones de PHP nativas para más plataformas- y un API más elaborado.

En el último año, el número de servidores que utilizan PHP se ha disparado, lo que le ha convertido a PHP en una tecnología popular. Esto es debido, entre otras razones, a que PHP es el complemento ideal para que la dupla GNU/Linux-Apache sea compatible con la programación del lado del servidor de sitios web. Gracias a la aceptación que ha logrado, y los grandes esfuerzos realizados por una creciente comunidad de colaboradores para implementarlo de la manera más óptima, podemos asegurar que el lenguaje se convertirá en un estándar que compartirá los éxitos augurados al conjunto de sistemas desarrollados en código abierto.

Poco a poco PHP se ha ido convirtiendo en un lenguaje que nos permite hacer de todo. En un principio diseñado para realizar poco más que un contador y un libro de visitas, PHP ha experimentado en poco tiempo una verdadera revolución y, a partir de sus funciones, en estos momentos se pueden realizar una variedad de tareas útiles para el desarrollo del web como son:

Funciones de correo electrónico

Podemos con una facilidad asombrosa enviar un correo electrónico a una persona o lista parametrizando toda una serie de aspectos tales como el correo electrónico de procedencia, asunto, persona a responder, etc.

En nuestro caso en particular se usaron este tipo de funciones para mandar el formulario capturado en el sitio web al capturista correspondiente.

Gestión de bases de datos

Resulta difícil concebir un sitio actual, potente y rico en contenido que no es gestionado por una base de datos. El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas, a partir de las cuales podremos editar el contenido de nuestro sitio con absoluta sencillez.

Gestión de archivos

Crear, borrar, mover, modificar, etc. Cualquier tipo de operación medianamente razonable que se nos pueda ocurrir puede ser realizada a partir de una amplia biblioteca de funciones para la gestión de archivos por PHP. También podemos transferir archivos por FTP a partir de sentencias en nuestro código.

Tratamiento de imágenes

Evidentemente resulta mucho más sencillo utilizar algún editor de imágenes para el tratamiento de imágenes, el problema se complica si tenemos que tratar miles de imágenes enviadas por nuestros visitantes, ¿cuál sería la mejor opción? También puede resultar muy tedioso uniformar el tamaño y formato miles de imágenes recibidas día tras día. Todo esto puede ser también automatizado

eficazmente mediante PHP.

También es útil el crear botones dinámicos, es decir, botones en los que utilizamos el mismo diseño y solo cambiamos el texto. Podremos por ejemplo crear un botón haciendo una única llamada a una función en la que introducimos el estilo del botón y el texto a introducir, obteniendo automáticamente el botón deseado. A partir de la biblioteca de funciones gráficas podemos hacer esto y mucho más.

Muchas otras funciones pensadas para Internet (tratamiento de cookies, accesos restringidos, comercio electrónico, entre otras) o de propósito general (funciones matemáticas, explotación de cadenas, de fechas, corrección ortográfica, compresión de archivos, etc) son realizadas por este lenguaje. A esta inmensa biblioteca cabe ahora añadir todas las funciones personales que el programador va creando por necesidades propias y que luego son reutilizadas en otros sitios. Como puede verse, las posibilidades que se nos presentan son sorprendentemente vastas. Lo único que se necesita es un poco de ganas de aprender y algo de paciencia en nuestros primeros pasos. En nuestro caso utilizaremos PHP para enviar la información capturada en los formularios de nuestros sitios web a los capturistas correspondientes.

Visual Basic

En 1987 Microsoft lanzó Visual Basic. Fue su primera herramienta de desarrollo visual, y fue desarrollada para competir con C, C++, Pascal y otros lenguajes de programación. Al principio Visual Basic no fue bien recibido, no fue sino hasta su versión 2 liberada en 1991 que la gente realmente descubrió el potencial del lenguaje y ya para su versión 3 se volvió muy popular. Anteriormente el desarrollo de aplicaciones para ambientes de ventanas era más complicado que para aplicaciones en ambientes de texto, los programadores tenían que preocuparse por lo que estaba haciendo el ratón, dónde estaba el usuario dentro de un menú, si oprimía una vez o dos un botón, etcétera, el desarrollar aplicaciones demandaba mucho tiempo de los programadores. Visual Basic cambió esta situación, se pueden desarrollar aplicaciones para ambientes de ventanas mucho más rápido, y se reducen los errores relacionados con la programación de la interfaz gráfica, incluye dos conceptos importantes:

1. Un método visual de creación de aplicaciones, incluyendo formularios, controles y componentes de dicho formulario.
2. La habilidad de asociar código directamente a cada evento de cada elemento del diseño visual.

Visual Basic es ideal para desarrollar aplicaciones que se ejecuten en ambientes Windows y/o en aplicaciones de Microsoft Office. En nuestro caso, utilizaremos Visual Basic para construir una función que convierta cadenas de texto a formato de oración (mayúsculas y minúsculas), sobre una hoja de cálculo.

3.6. Redes (Network)

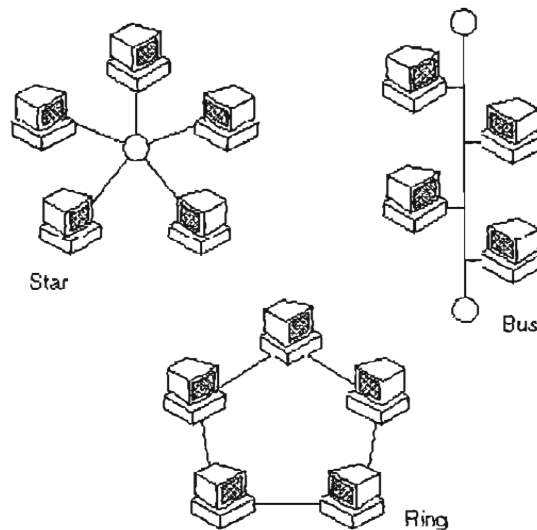
Hoy en día es básico poder intercambiar mensajes, archivos y tener acceso a recursos compartidos (como un servidor de archivos o de impresión), de aquí surge la necesidad de poner dichos recursos en una red. Una red está compuesta por un conjunto de computadoras, impresoras, ruteadores¹³, switches¹⁴ y otros dispositivos que pueden comunicarse entre sí a través de un medio en particular. Aunque se manejan varias clasificaciones para las redes, hay 4 factores a tomar en cuenta [2]:

13. "Router", *Ruteador*: Es un dispositivo de red que utiliza medidas para determinar la trayectoria óptima a lo largo de la cual deba direccionarse el tráfico de la red. Los ruteadores dirigen paquetes de una red a otra con base en la información de la capa de red. [3]

14. "Switch" *Commutador*: Es un equipo en la red que filtra, direcciona y difunde tramas con base a la dirección de destino de cada trama. [3]

1.- Por su **ubicación** las redes pueden clasificarse en LAN¹⁵ y WAN¹⁶. En el caso de las LAN las computadoras (o equipos en general) están geográficamente cerca unas de otras (por ejemplo en el mismo edificio) a diferencia de las WAN donde los equipos no están cerca unos de otros y normalmente se conectan por enlaces como líneas telefónicas o por ondas de radio.

2.- Por su **topología**. Se refiere al arreglo físico de los nodos y el medio de transmisión dentro de una estructura de red, en otras palabras, es la forma geométrica en la que se disponen los equipos en la red. Las más populares son en línea (bus), en estrella (star) o en anillo (ring). A continuación un diagrama donde se muestran éstas 3 topologías:



3.- Por su **protocolo**: Un protocolo define un conjunto común de reglas y señales que los equipos en la red usan para comunicarse. Hablando de protocolos de red [G5], uno de los protocolos más populares para redes LAN es Ethernet, otro de ellos es token ring por nombrar algunos.

La red de anillo (Token Ring) fue desarrollada originalmente por IBM en los 70's. Es un protocolo de red que corre a 4 y 16 Mbps sobre una topología tipo anillo. Este tipo de redes mueven un pequeño paquete de información llamado lexema (token) a través de la red. La posesión de dicho token da derecho a transmitirlo. Si un nodo recibe el token y no tiene información que transmitir, pasa el token a la siguiente estación. Cada estación puede retener por un periodo máximo de tiempo determinado.

El término Ethernet se refiere al protocolo de red para la implementación de LAN que incluye 4 categorías principales:

* Ethernet e IEEE 802.3 – Son las especificaciones LAN que operan a 10 Mbps a través de cable coaxial.

15. LAN "Local Area Network", Red de área local.

16. WAN "Wide Area Network", Red de área extendida.

* Ethernet a 100 Mbps – Es una sola especificación LAN, también conocida como Fast Ethernet, que opera a 100 Mbps a través de un cable de par trenzado.

*Ethernet a 1000 Mbps – Es una sola especificación LAN, también conocida como Gigabit Ethernet, que opera a 1000 Mbps a través de cables de fibra óptica y de par trenzado.

*Ethernet a 10000 Mbps - Es una sola especificación LAN, también conocida como 10 Gigabit Ethernet, que opera a 10000 Mbps a través de cables de fibra óptica y de par trenzado.

La red Ethernet fue creada por Xerox en la década de los 70's, y ha prevalecido como una tecnología de transmisión fundamental, gracias principalmente a 4 aspectos:

1 Es fácil de entender, implementar, administrar y mantener.

2 Permite implementaciones de red de bajo costo.

3 Provee una gran flexibilidad para instalaciones de red.

4 Garantiza la interconexión y operación de los productos que cumplen con el estándar sin importar el fabricante [15].

Además de los protocolos de red, es importante determinar el protocolo de transporte [G5], TCP/IP (Transmission Control Protocol/Internet Protocol *Protocolo de control de transmisión / Protocolo de Internet*) es un conjunto de reglas que siguen los equipos para poder transferir información entre ellos, se encuentra en la capa de transporte del modelo OSI [G5]. TCP monitorea y garantiza la transmisión correcta de datos. IP recibe los datos de TCP, los descompone en paquetes, y los entrega sobre una red (como Internet). TCP/IP en conjunto, es un conjunto de protocolos que incorpora las funcionalidades de ambos.

HISTORIA

El Protocolo de Internet (IP) y el Protocolo de Transmisión (TCP), fueron desarrollados inicialmente en 1973 por el informático estadounidense Vinton Cerf como parte de un proyecto dirigido por el ingeniero norteamericano Robert Kahn y patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA, siglas en inglés) del Departamento Estadounidense de Defensa.

Internet comenzó siendo una red informática de ARPA (llamada ARPAnet) que conectaba redes de computadoras de varias universidades y laboratorios en investigación en Estados Unidos de Norteamérica. World Wide Web se desarrolló en 1989 por el informático británico Timothy Berners-Lee para el Consejo Europeo de Investigación Nuclear (CERN, siglas en francés).

QUÉ ES Y ARQUITECTURA DE TCP/IP

TCP/IP es el protocolo común utilizado por todas las computadoras conectadas a Internet, de manera que éstos puedan comunicarse entre sí. Hay que tener en cuenta que en Internet se encuentran conectadas computadoras de clases muy diferentes con *hardware* y *software* incompatibles en muchos casos, además de todos los medios y formas posibles de conexión. Aquí se encuentra una de las grandes ventajas del TCP/IP, pues este protocolo se encargará de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de *hardware*. *TCP/IP no es un único protocolo, sino lo que se conoce en realidad con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI*. Los dos protocolos más importantes son el TCP (*Transmission Control Protocol*) y el IP (*Internet Protocol*), que son los que dan nombre al conjunto.

La arquitectura del TCP/IP consta de cinco niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI [G6].

El TCP/IP necesita funcionar sobre algún tipo de red o de medio físico que proporcione sus propios protocolos para el nivel de enlace de Internet. Por este motivo hay que tener en cuenta que los protocolos utilizados en este nivel pueden ser muy diversos y no forman parte del conjunto TCP/IP. Sin embargo, esto no debe ser problemático puesto que una de las funciones y ventajas principales del TCP/IP es proporcionar una abstracción del medio de forma que sea posible el intercambio de información entre diferentes medios y tecnologías que inicialmente son incompatibles.

CARACTERÍSTICAS DE TCP/IP

Ya que dentro de un sistema TCP/IP los datos transmitidos se dividen en pequeños paquetes, éstos resaltan una serie de características.

La tarea de IP es llevar los datos a granel (los paquetes) de un sitio a otro. Las computadoras que encuentran las vías para llevar los datos de una red a otra (denominadas enrutadores) utilizan IP para trasladar los datos. En resumen IP mueve los paquetes de datos a granel, mientras TCP se encarga del flujo y asegura que los datos estén correctos.

Los datos no tienen que enviarse directamente entre dos computadoras. Cada paquete pasa de computadora en computadora hasta llegar a su destino. Éste, claro está, es el secreto de cómo se pueden enviar datos y mensajes entre dos computadoras aunque no estén conectadas directamente entre sí. Lo que realmente sorprende es que sólo se necesitan algunos segundos para enviar un archivo de gran tamaño de una máquina a otra, aunque estén separadas por miles de kilómetros y pese a que los datos tienen que pasar por múltiples computadoras. Una de las razones de la rapidez es que, cuando algo anda mal, sólo es necesario volver a transmitir un paquete, no todo el mensaje.

Los paquetes no necesitan seguir la misma trayectoria. La red puede llevar cada paquete de un lugar a otro y usar la conexión idónea que esté disponible en ese instante. No todos los paquetes de los mensajes tienen que viajar, necesariamente, por la misma ruta, ni necesariamente tienen que llegar todos al mismo tiempo.

La flexibilidad del sistema lo hace muy confiable. Si un enlace se pierde, el sistema usa otro. Cuando usted envía un mensaje, el TCP divide los datos en paquetes, ordena éstos en secuencia, agrega cierta información para el control de errores, después los lanza hacia fuera, y los distribuye. En el otro extremo, el TCP recibe los paquetes, verifica si hay errores y los vuelve a combinar para convertirlos en los datos originales. De haber error en algún punto, el programa TCP destino envía un mensaje solicitando que se vuelvan a enviar determinados paquetes.

4.- Finalmente, podemos clasificar a las redes por su **arquitectura**, es una descripción de la manera en que las comunicaciones se llevan a cabo en una red. En este punto clasificaremos las redes por esquemas de punto a punto o por modelos cliente-servidor

Bajo los esquemas punto a punto, cada estación de trabajo tiene capacidades y responsabilidades equivalentes, por otro lado, en esquemas tipo cliente-servidor, cada equipo es un cliente o un servidor, es decir, hay algunos equipos en la red que dan servicios a los demás. Cada vez las computadoras se han vuelto más rápidas, más poderosas y más baratas, los diseñadores se han alejado cada vez más de arquitecturas de sistemas centralizados.

3.7. El modelo cliente-servidor

La llegada de las redes permitió la comunicación entre computadoras distantes. La red permite compartir recursos distribuidos entre diferentes equipos, los sistemas de archivos remotos permiten a una computadora montar uno o más sistemas de otras máquinas remotas. En éste caso, la máquina que contiene los archivos es el servidor, y la máquina que requiere acceder a los archivos es el cliente. La relación cliente-servidor es común en máquinas conectadas a alguna red. Generalmente, el servidor declara que existe un recurso disponible para los clientes y especifica exactamente de qué recurso se trata y especifica para qué clientes está disponible. Un servidor puede manejar múltiples clientes, y un cliente puede usar recursos de múltiples servidores [12].

En resumen, en cuanto a la red, es muy importante verificar que las aplicaciones y los equipos sean compatibles con el tipo de red que seleccionemos y que el nivel de rendimiento de la red (velocidad, capacidad de transmisión y disponibilidad) cumplan con lo requerido. Otro aspecto muy importante a considerar es si tendremos oficinas remotas, así como la manera en que intercambiarán datos dichas oficinas.

3.7.1. Procedimientos (Procedures)

Es una serie de actos u operaciones con que se logra hacer alguna cosa en particular. De poco servirá tener un excelente hardware, haber escogido correctamente el software, el sistema de administración de bases de datos y tener una red eficiente si no existen procesos bien definidos que expliquen la forma en que interactúan cada uno de los componentes del sistema. Los procesos que se usarán deben estar perfectamente documentados para que el sistema de información realice las tareas que requiere, hacer diagramas de los procesos que sigue la información desde su recolección hasta su uso en la base de datos facilita ver el flujo de los mismos. Ya con los procesos en un diagrama podemos buscar la manera de mejorar dichos procesos (hacer re-ingeniería de procesos) y ver más fácilmente donde existan fallas.

3.7.2. Usuarios (Users)

Se refiere a cualquier persona en general que use una computadora, en nuestro caso serán las personas que usen el sistema de información. Se deben asignar los permisos necesarios y suficientes para que cada uno de ellos pueda realizar sus tareas sin poner el riesgo al sistema.

3.8. Desarrollo del sistema de información en conjunto

La planeación de un sistema de información se divide en 3 etapas principales (*análisis, diseño e implementación* [23]):

Análisis

Etapas 1: Definir el alcance del sistema de información. En esta etapa hay que hacer entrevistas con toda la gente involucrada y hacer un documento de alcances donde se especifique todo lo que debe hacer el sistema.

Etapas 2: Definir los procedimientos para usar el sistema. Definir los procedimientos de cada proceso involucrados en el sistema. Como se mencionó en el punto 3.5, es recomendable tener diagramas que ejemplifiquen como interactúan los procesos con los usuarios y la infraestructura.

Diseño

Etapa 3: Escoger los componentes que sean capaces de realizar las tareas especificadas en la Etapa 1. El hardware, software, sistemas de administración de base de datos y redes, deben ser adecuadas. Debemos verificar que puedan soportar la carga de trabajo requerida y que están listos para los cambios que pudieran presentarse. Tenemos que tomar en cuenta varios factores para hacer una elección adecuada de los componentes, partiendo de un presupuesto y en base a éste, buscar posibles soluciones que cumplan con lo requerido (sin pasar del presupuesto). Para la parte del hardware se deben ocupar equipos que darán un buen rendimiento en un tiempo aceptable de vida. Para el software, (hablando del sistema operativo), se debe escoger uno estable que sea compatible con el sistema de administración de base de datos que piense usar y con la infraestructura instalada (si es que la hubiese), el sistema de administración de base de datos debe ser estable, confiable, capaz de soportar la carga de trabajo proyectada además de proveer la Información requerida. En cuanto a la red debemos verificar que las aplicaciones y los equipos sean compatibles con el tipo de red que se haya seleccionado y que cubra las expectativas de desempeño.

Además hay que tener presente que no por usar tecnologías o metodologías al último grito de la moda, vamos a tener un mejor sistema. Antes de optar por algún componente en particular tenemos que explorar a detalle sus características. El simple hecho de conocer a fondo cada opción permitirá idear formas óptimas de Integración entre dichos componentes.

Etapa 4: Hacer un diagrama del sistema en su totalidad, con la finalidad de tener una visión completa del sistema de información, esto ayudará a ver posibles fallas tanto en los procedimientos como en la infraestructura.

Implementación

Etapa 5: Implementar el sistema. Desarrollar el sistema, preparar los equipos, dar de alta la base de datos, poner los datos de inicio de dicha base (si es que los hubiere), en otras palabras, dejar todo listo para entrar a la etapa de pruebas.

Etapa 6: Probar el sistema ANTES de entrar a producción. Lo ideal es que los usuarios hagan simulaciones de uso del sistema. También simular fallas y probar si el sistema es capaz de ser restaurado. Crear una línea de base¹⁷ y probar el sistema en situaciones de cargas de trabajo excesivas para ver como se comportará respecto a la línea de base establecida una vez que se libere dicho sistema.

Etapa 7: Puesta en operación del sistema. Después de haber hecho todas las pruebas y correcciones tanto en diseño, procedimientos y equipos, se liberará el sistema en producción, hay que seguir muy de cerca su operación durante los primeros días, es muy probable que se presenten más detalles a pesar de haber hecho pruebas exhaustivas. Documentar el sistema y elaborar manuales para cada tipo de usuario de acuerdo a su rol en el mismo forma parte del ciclo de la creación del software.

¹⁷ "Baseline" *Línea de base:* Es una medida usada como referencia para hacer comparaciones. En este caso en particular consideraremos que la línea base medirá hasta que punto se considera aceptable el desempeño de algún componente del sistema.

Diseño de sistemas de información para publicaciones de circulación gratuita controlada.

Parte III

Análisis, diseño, implementación y futuro del sistema de información de Netmedia.

Capítulo 4.- Implementación del sistema de información de Netmedia.

En este capítulo se describe el proceso de implementación del sistema de netmedia de acuerdo a las etapas vistas en el capítulo 1.7 de la sección anterior. En general se siguió a pie de la letra la teoría expuesta, la implementación del sistema de información se hizo de la siguiente manera:

4.1. Análisis

Etapas 1: Definir lo que va a hacer el sistema exactamente

El objetivo del sistema de información es manejar la base de datos de los suscriptores en nuestras publicaciones y poder segmentar la información para ofrecer campañas de publicidad a los anunciantes. Entendamos por "manejar", todos aquellos procesos relacionados con la contratación de nuevos lectores (sea cual sea su origen), mantener actualizados los que tenemos, capturar todos los formularios y establecer niveles de seguridad para los diferentes tipos de usuarios. Asimismo el sistema debe ser lo suficientemente flexible para poder adecuarse a futuros cambios y permitir agregar o quitar publicaciones de acuerdo a las necesidades de la empresa.

Etapas 2: Definir los procedimientos

Durante la ejecución del sistema los principales procesos simultáneos son los siguientes:

- 1.- Captura de nuevas solicitudes de suscripción que lleguen por alguno de nuestros sitios web¹.
- 2.- Captura de nuevos registros que no lleguen por medio de nuestros sitios web (puede ser que obtengamos otras bases de datos, ya sea en forma electrónica o en papel).
- 3.- Verificación de los datos registrados de alguna publicación que fuese reportada como devolución por los servicios de mensajería.
- 4.- Verificación general de registros aunque no se hayan reportado como devolución, esto con el objetivo de asegurarnos de que realmente este recibiendo la información que nosotros creemos y de tener la información lo más actualizada que sea posible.
- 5.- Escoger el segmento de nuestra base de datos que consideremos más importante (basándonos en los requerimientos más frecuentes de nuestros anunciantes) y tener lo más actualizado posible estos registros.

¹ Sitio Web: Es un lugar en Internet que contiene documentos que pueden ser consultados por los visitantes a dicho sitio

A continuación se detallarán los 5 procesos:

Proceso 1.- Captura de nuevas solicitudes de suscripción que lleguen de alguno de nuestros sitios web.

Paso 1: El aspirante a cliente-lector entra desde su oficina/casa (o de algún lugar con acceso a Internet) para llenar la forma de suscripción.

Paso 2: Toda la información capturada por el usuario debe cumplir con ciertos requisitos, por ejemplo si nos interesa recibir una solicitud con nombre, correo electrónico² o con algunos datos que consideremos importantes. En este caso particular y de acuerdo a las exigencias más comunes de los clientes consideramos como obligatorios los siguientes campos³ del formulario: Nombre, correo electrónico, giro de la empresa en la que trabaja, número de empleados en la empresa y año de nacimiento. De aquí surge otra necesidad, la validación, es decir que el formulario certifique los campos que mencionamos.

Paso 3: Ya que se encuentra validada la información capturada, debe llegar a nuestras manos. Básicamente esto puede ocurrir de 3 formas, ya sea que la información sea depositada directamente en la base de datos, ó se genere un archivo con la información capturada y se almacene en algún lugar en el servidor o por último, que se genere un correo electrónico el cual recibirá la persona encargada de la captura a la base de datos.

Cada una de ellas tiene ventajas y desventajas, analicemos cada una de éstas. Si escogemos la opción de que los datos sean depositados directamente en la base de datos hay tres problemas principales a resolver. El primero es la seguridad, pues al tener un sitio en Internet es necesario asegurarnos de que la información y los sistemas relacionados con esta, se encuentren bien resguardados pues cualquier falta de la misma podría ocasionar la pérdida o corrupción de los datos, entre otras cuestiones. El segundo es el prevenir las posibles fallas ocasionadas por desastres naturales o algún dispositivo físico en mal estado de la computadora, por ese motivo la base de datos se encuentran dispuesta fuera de las oficinas de la empresa, particularmente se contrató el hospedaje externo⁴ de la misma. El tercer problema es la validación de los datos que ingresan a la base de datos, esto es que cumplan con los estándares establecidos para su capturada. Por ejemplo, si alguien trabajara en IBM en el campo de empresa pudiera escribir una gran variedad de combinaciones: IBM, I.B.M. IBM de México, IBM corp, etc. Provocando un problema grave, ya que muchos requerimientos de información dependen de que todo se capture de manera consistente, y así poder cazar la información y extraerla de manera consistente y fidedigna. Un ejemplo de esto sería al hacer una consulta a la base de datos para conocer el número de registros de la empresa IBM, si la información no se encontraría estandarizada, sería muy complicado responderla, ya que probablemente nos falten los datos capturados de manera distinta. La ventaja indiscutible es que se elimina el proceso de captura, si se tuviera un formulario donde todas las preguntas fueran cerradas, es decir, que todas las preguntas tuvieran opciones preestablecidas por nosotros, entonces sin duda éste sería el método más conveniente, pero al vernos forzados a campos con preguntas abiertas en la práctica no resulta ser el método más adecuado.

²"E-mail", *Correo electrónico*: Es una forma de mandar mensajes electrónicamente de una computadora a otra sobre una red o una conexión de Internet.

³ *Field Campa* Es un espacio reservado para guardar algún tipo de información, en bases de datos un campo es la unidad más pequeña de información que se puede acceder. Por ejemplo la edad de una persona, su tipo de sangre, etc. [2]

⁴ "Web hosting" *Hospedaje externo*: Es un servicio que brinda una empresa que permite guardar páginas y demás contenido web en un servidor para que pueda ser vista por el público en general desde Internet. [6]

Por otro lado, si optáramos por la creación de un archivo en el servidor después de que el aspirante llene el formulario, si existiría un proceso extra entre la captura del aspirante y la entrada de datos a la base de datos, una persona se encargaría de estandarizar la información y de capturarla en la base, sin embargo dichos archivos siguen en el servidor y si llegara a haber problemas con el enlace de telecomunicaciones no tendríamos acceso a ellos.

Es por eso que decidimos adoptar un tercer modelo, en el cual el formulario genera un correo electrónico que llega a un moderador encargado de depositar la información a la base de datos, en éste proceso intermedio se hace la estandarización independientemente del enlace al servidor web, ya que la información enviada a través de los correos electrónicos se encuentra físicamente en la computadora del capturista y la razón más importante de todas es que al tener la base de datos en nuestra oficina en un ambiente de red LAN, se puede seguir aprovechando el tiempo para continuar trabajando. El siguiente paso es desarrollar el formulario el cual generará un correo electrónico con la información.

Paso 4: Una vez que es recibido el correo electrónico con la información capturada, la persona a cargo debe entrar a la base de datos para ingresar al nuevo aspirante y/o modificar sus datos si es que ya tenía un registro en la base de datos. Si bien es cierto que las primeras preguntas del formulario son abiertas (nombre, dirección, empresa, etc.) las siguientes son cerradas, es decir con opciones preestablecidas, las respuestas de éstas se pueden almacenar tal cual en la base de datos ya que no necesitan ser estandarizadas; pero las primeras no, (sería un tanto complicado guardar directamente los datos de la aplicación de correo en la base de datos). De este modo tenemos un paso intermedio, a través de una hoja de cálculo en donde se almacenará la información recibida, se le harán los cambios necesarios para mantener la estandarización y por último se almacenará en la base de datos.

Paso 5: Conforme se van estandarizando los registros son enviados uno por uno a la base de datos. Como cada suscriptor tendrá un número único de identificación y no queremos que haya huecos en la numeración de dicha clave, cuando un registro ya no es útil se le pone una marca que indica que puede ser reutilizado, en caso de haber una cantidad suficiente de ellos disponibles, se capturan ahí los nuevos registros, en caso contrario se agregan nuevos al final de la base de datos, ya sea reutilizándolos o agregándolos los cuales deben ser calificados de acuerdo a los criterios que establecimos para saber si se autoriza la suscripción gratuita o no. Dichos criterios se basan en el perfil que estamos buscando, recordemos que la gente que nos interesa son personas que toman las decisiones en sistemas de las empresas más importantes del país. Si califican o no, el registro queda capturado en la base de datos, en caso de no calificar para el envío de publicaciones en papel, se le envían versiones electrónicas de las revistas por correo electrónico.

Paso 6: Notificar al aspirante si calificó o no. Ya que tenemos calificados los registros, usamos un programa que permite enviar correos electrónicos personalizados, que incluyen el nombre del aspirante y el resultado de su calificación.

Proceso 2.- Captura de registros que no llegan de nuestro sitio web.

Este proceso varía mucho dependiendo del origen de los datos a capturar. Si se trata de una base de datos capturada ya electrónicamente, el administrador de la base de datos deberá hacer los cambios necesarios en el formato de la base de datos para que sea compatible con nuestro sistema, es decir, hay que asegurarnos que la información se encuentre estandarizada de manera que no tengamos problemas futuros, en nuestra base de datos.

Por ejemplo, si en nuestra base de datos hay un campo como "nombre" y en él se captura el nombre completo empezando por el nombre de pila y luego los apellidos, hay que asegurarnos de que la base que pensamos capturar tenga el mismo formato en cuanto al nombre de la persona y si es necesario, hacer los cambios pertinentes, respetando la estandarización. Si pensamos capturar una base de datos que no se encuentre en formato electrónico, el capturista abrirá un formato de captura en nuestra base de datos e ingresará de uno en uno los nuevos registros. En ambos casos, tenemos que decidir si los registros capturados deben ser marcados para su calificación posterior o si entran ya autorizados a recibir alguna(s) publicación(es).

Proceso 3.- Verificación de la devolución.

Debido a los costos y tiempos de entrega, hemos decidido que las publicaciones que abarquen al D.F. y al área Metropolitana sean manejados por una empresa de mensajería y el resto del país se hace por medio del servicio postal mexicano (SEPOMEX). Con cada una de ellas se tiene una manera diferente de trabajar.

Para la mensajería, acordamos un tiempo máximo de entrega de 9 días, durante los cuales la mensajería nos envía por correo electrónico un listado con los identificadores que tengan problemas, este correo electrónico le llega al capturista encargado de la verificación, los busca en nuestra base de datos y hace los cambios correspondientes o marca el registro para que pueda ser reutilizado en caso de no poder ser verificado.

Para el caso de SEPOMEX el tiempo en que conocemos de una devolución es muy variable, y está en función de que tan lejos fue la publicación, puede tardar hasta 3 meses en regresar alguna publicación para los estados más alejados. El proceso se hace conforme llega la devolución física, se toma de la etiqueta el número de suscriptor, se intenta localizar al suscriptor y se hace el cambio pertinente.

Proceso 4.- Verificación de otros registros.

Existen publicaciones que no son reportadas como devoluciones, solo algunos datos del suscriptor ya no son correctos. También es muy probable que la información de un registro capturado hace 5 años ya no sea válida. En el mundo ideal, si los servicios de mensajería no reportasen una devolución significaría que los datos del registro son correctos, pero en la vida real no funciona así, muchas veces se llega preguntando por alguien y como no se paga un servicio de mensajería con acuse de recibo debido a su costo elevado, puede que lo reciba otra persona y nunca nos enteremos de ese error operacional. Todos los registros tienen 2 fechas, una es su fecha de ingreso al sistema y la otra es la fecha de su última verificación, a los capturistas se les asignan registros con fechas de verificación cuya diferencia a la fecha actual es de un año o más.

Proceso 5.- Escoger el segmento más importante de nuestra base de datos y mantenerlo lo más actualizado posible.

Tendremos en la base de datos registros de gente que nos interesen mucho más que los demás, esa pequeña porción de nuestra base de datos debe de ser ubicada con facilidad y debe tener la información lo más reciente posible. De acuerdo al sector del mercado, nos referimos a los directores de sistemas de las 1,000 empresas más importantes del país, nuestra experiencia nos ha hecho ver que si verificamos esos registros 2 veces al año tenemos un buen control sobre los cambios que pudieran presentarse.

Entonces los capturistas toman el listado de las 1,000 empresas más importantes del país (actualmente nos basamos en un listado que publica la revista "Expansión"), buscamos a los directores de sistemas, y en caso de no tenerlos registrados se dan de alta, y en caso contrario solo se verifica el registro. Con esto logramos tener al día los registros que son más importantes de acuerdo a los requerimientos de nuestros anunciantes.

4.2. Diseño

Etapa 3: Escoger los componentes

Teniendo en mente lo que debe hacer exactamente el sistema y los procesos que se usan concurrentemente, desarrollamos una lista inicial de requerimientos.

En la parte relacionada con el hardware y software para el servidor corporativo en nuestras oficinas optamos por un servidor con procesador Xeon con discos duros SCSI⁵. Se escogió éste equipo debido a que el procesador Xeon ofrece una gran estabilidad y poder de procesamiento. Se optó por discos SCSI sobre uno IDE/ATA⁶ normal, debido a que ofrecen un mejor rendimiento en aplicaciones complejas. Dichos discos se encuentran en un arreglo RAID5⁷ [G1] para tener un ambiente tolerante a fallas con un excelente desempeño. En cuanto al sistema operativo para el servidor hemos decidido usar una infraestructura basada en productos Microsoft, que además de ser lo más comercial cumplen con las exigencias del sistema y era lo que nos ofrecía una buena relación costo-desempeño.

Dividimos en 2 partes los componentes a escoger, primero los que están físicamente en nuestra oficina y luego los que se encuentran con la empresa que hace el hospedaje de nuestro sitio web.

El sistema operativo del servidor será Windows 2003 Server, por las siguientes razones [22]:

- Toda nuestra infraestructura estará basada en productos de Microsoft, para varios componentes es requisito contar con un sistema operativo de servidor Microsoft, por ejemplo para operar el servidor de correo (Exchange 2000) o para el manejador de bases de datos (SQL Server 2000) es necesario un sistema operativo de servidor Microsoft.
- Al usar el servicio de directorio que almacena la información sobre objetos en la red, facilita a los administradores la búsqueda, el uso y la administración de dichos objetos. Este servicio utiliza un tipo de almacenamiento de datos estructurado como punto de partida para la organización jerárquica y lógica de la información del directorio. Centraliza la administración y se tiene control sobre los permisos de los diversos recursos en la red.
- Es fácil de implementar, administrar y usar. Los asistentes y las interfaces de administración facilitan mucho la configuración del servidor ya que ofrece una mayor estabilidad y seguridad sobre su versión anterior.

5 SCSI "Small computer system interface" *Pequeña interfaz de sistema informático*: Es una interfaz paralela estándar para conectar dispositivos a la computadora como discos o impresoras. [2]

6 IDE "Integrated drive electronics" *Unidad de disco electrónica integrada*: Interfaz que se usa para conectar dispositivos a la computadora, mucha gente lo usa para referirse a la especificación ATA. [2]

7 RAID5 "Redundant array of independent disks" *Arreglo redundante de discos independientes*: Es una categoría de unidades de disco que emplean una o más unidades combinadas para ofrecer tolerancia a fallos y mejor rendimiento. Por lo normal son usados en servidores. [7]

- Sus herramientas gráficas de administración de discos, permiten realizar fácilmente tareas que antes eran complicadas.
- Permite habilitar la opción de copias de seguridad sobre archivos abiertos.
- Ofrece una consola muy completa para administrar accesos remotos (tipo VPN⁸) y manejar conexiones inalámbricas.
- El servicio de DHCP se puede integrar con DNS y con el servicio de directorio con lo que las tareas de asignación y mantenimiento de las direcciones IP se hace de manera automática.
- El servidor puede reconocer y adaptarse de manera automática (en la mayoría de los casos) a cambios de configuración del hardware, sin necesidad de una intervención o bien de reiniciar el equipo.
- Puede hacer las veces de un ruteador para unir dos segmentos diferentes de red dentro de una LAN.
- En el área de diseño de la editorial se utilizan equipos Macintosh, Windows Server 2003 cuenta con un servidor de archivos que es accesible mediante redes TCP/IP y redes de Apple Talk. Sus servicios de impresión permiten a los clientes de Macintosh imprimir con los recursos de impresión compartidos con otros sistemas Windows mediante el protocolo de Apple Talk.

Al principio se usaba una base de datos desarrollada en Microsoft Access, el problema es que al empezar a trabajar con varios usuarios concurrentes la aplicación daba muchos problemas, la base pedía ser compactada constantemente para reparar sus índices y como iba creciendo el número de usuarios recurrentes tuvimos que evaluar otras opciones. La que nos pareció mejor para manejar la información contenida en la base de datos fue SQL Server de Microsoft, debido a que SQL Server es un manejador de bases de datos fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Ofrece un modelo de programación rápido y sencillo para desarrolladores, elimina la administración de la base de datos para operaciones estándar. Disminuye el costo total de propiedad a través de características como administración multi-servidor con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y tareas administrativas, libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Algunas de las características por las que se decidió utilizar dicho manejador de bases de datos son:

Estabilidad: Mantiene la seguridad y confiabilidad que exigen los sistemas de negocios de misión crítica.

Escalabilidad: Se adapta a las necesidades de la empresa, soportando desde pocos usuarios hasta varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.

Potencia: Es un manejador de bases de datos que se integra adecuadamente en ambientes Windows.

Gestión: Una completa interfaz gráfica que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas que son compatibles con el manejador de base de datos.

⁸ VPN "Virtual private network" *Red privada virtual*: Es una red de datos privada que usa la infraestructura pública de telecomunicaciones, manteniendo privacidad mediante el uso de protocolos de túnel y procedimientos de seguridad.

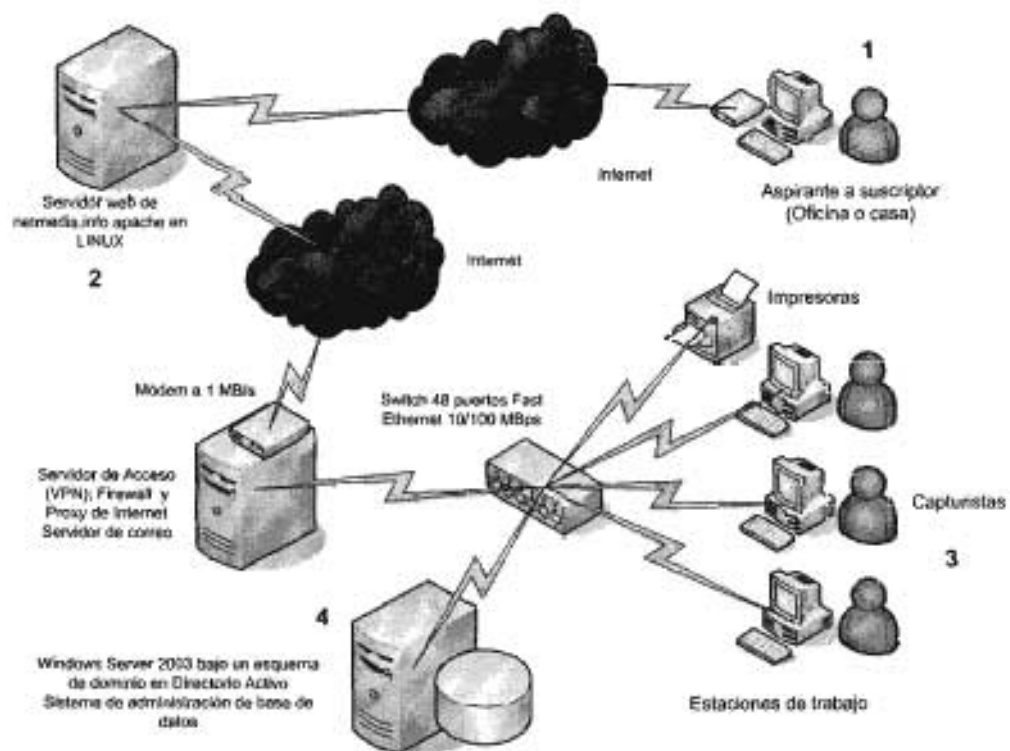
En cuanto a la red es pertinente pensar en una red Ethernet en una topología de tipo estrella en una arquitectura tipo cliente-servidor para el sistema que desarrollaremos. Ethernet es compatible con una gran cantidad de plataformas y su velocidad de transmisión de datos es aceptable para los requerimientos de nuestro sistema de información.

Se optó por un modelo cliente – servidor debido a que facilita las tareas de administración del sistema al tener la base de datos en un servidor. Otra razón es que al usar este esquema el servidor es el encargado del procesamiento (el trabajo pesado), y podemos usar clientes que no sean equipos tan poderosos con lo que podemos reducir costos.

En lo que concierne al equipo que está en la empresa que realiza la tarea referente al hospedaje, usamos un servidor con características similares al que tenemos en la oficina, sólo que éste equipo usa Linux RedHat versión 9, y el servidor de Internet Apache. Decidimos usar Linux y Apache por su gran estabilidad, disponibilidad y seguridad para aplicaciones web.

Etapa 4: Hacer un diagrama del sistema en su totalidad

El diagrama completo de la red tiene la siguiente forma:



El proceso es el siguiente:

- 1.- El aspirante a suscriptor entra a nuestro sitio web y llena el formulario de captura.
- 2.- La información llega al servidor web, el cual genera un correo electrónico con la información y que es enviado a la persona responsable de la captura en la base de datos.
- 3.- El capturista normaliza la información y posteriormente la deposita en la base de datos, se decide si el aspirante califica para recibir una suscripción y se le manda un correo electrónico avisándole si fue aprobada su solicitud.
- 4.- La información se encuentra almacenada en la base de datos para disponer de ella según se necesite.

Disponemos de 2 servidores con Windows Server 2003. Ambos están dados de alta como controladores de dominio para tener un ambiente tolerante a fallas. En uno de ellos se encuentra el manejador de base de datos SQL Server 2000, en el otro se encuentra un servidor de acceso remoto (por medio de VPN), un Proxy⁹ para compartir el acceso a Internet con los demás clientes de la oficina, un Firewall¹⁰ y un servidor de Exchange¹¹ en su versión 2000 para dar servicios de correo electrónico. Las estaciones de trabajo validan sus credenciales con el directorio activo de alguno de los controladores de dominio cada que necesiten hacer uso de algún recurso. Todo está interconectado en una red Ethernet 10/100 por medio de un switch. Nuestro acceso a la parte que no está en nuestras oficinas (y a Internet en general) se lleva a cabo mediante un enlace a 1 MBs. Este enlace ha sido más que suficiente para las necesidades de los clientes de la oficina.

En las instalaciones del proveedor de hospedaje se encuentra el servidor que tiene como sistema operativo LINUX y ahí se encuentran nuestros sitios web y el formulario que genera los correos electrónicos que se capturan en la base de datos.

4.3. Implementación.

Etapa 5: Desarrollar el sistema

Lo primero que se debe hacer es preparar los equipos, empezamos instalando el sistema operativo al primer servidor, es una tarea sencilla ya que Windows 2003 Server, incluyen un asistente muy amigable para llevar a cabo su instalación. Posteriormente hay que instalar el manejador de base de datos en el servidor, la instalación del servidor SQL versión 2000 tiene un asistente muy fácil de seguir y no representa mayor problema. Ya con SQL instalado viene la parte de programar la base de datos.

El desarrollo de una base de datos por parte de una organización puede descomponerse en tres grandes fases, cada una se subdivide en un conjunto de acciones concretas [1]:

- *Construcción de una base de datos, para lo cual habrá que definir los objetivos para los que se crea, identificar los datos necesarios y establecer cómo se van a usar dichos datos.*
- *Mantenimiento de la base de datos, que supondrá la verificación de los datos y sus actualizaciones posteriores.*
- *Uso de la base de datos es decir, la transformación de los datos en información útil.*

9 Proxy: Es un programa intermediario que actúa tanto como cliente como servidor con el propósito de hacer peticiones a nombre de otros clientes.

10 Firewall *Muro de fuego*: Es un sistema diseñado para prevenir accesos no autorizados a o desde una red privada. Los firewalls pueden ser implementados tanto en hardware como en software o una combinación de ambos. Frecuentemente son usados para prevenir accesos no autorizados de usuarios de internet a redes privadas, especialmente intranets.

11 Servidor de Exchange: Es un software de colaboración para servidor de Microsoft. Entre otras cosas da servicios de correo electrónico y calendarios a los usuarios de una red.

Siguiendo lo anterior, una vez que se tienen los objetivos de la base de datos bien definidos en la etapa 1, definimos los datos que realmente son necesarios. Los datos que contendrá la base de datos se dividen en 4 tipos, los datos generales del suscriptor, los del formulario de captura, los de las bases de correos electrónicos y los que controlan los registros. La información es adquirida de diversos lugares y bases de datos; el requisito para ingresar a nuestra base de datos es que fuese gente que tomara decisiones para la adquisición de tecnologías de información. Visto de otra forma en la base de datos existen en la actualidad 3 tipos de registros:

1.- Registros de inicio (conocemos los datos suficientes para poder enviar algo físicamente), aunque no tengamos su dirección de correo electrónico, ni el formulario contestado.

2.- Registros cuya única información disponible es su correo electrónico, dentro de éste rubro se encuentra la gente que se suscribe a los servicios de revistas electrónicas (e-Newsletters)¹².

3.- La gente que se suscribe directamente (ya sea por medio de los formatos en papel o por la forma que está en nuestro sitio web), llena el formulario y de esa forma conocemos muchos más datos, sin embargo este grupo debe tener los datos generales (por ejemplo, no puede haber un formato lleno "sin dueño").

Si bien es cierto que la base de datos tiene que partir de la información que obtenemos del formulario con las preguntas que se le hacen a los aspirantes a suscriptores, partiendo de estos tres tipos de registros, en el diseño no se pueden tratar igual, ya que seña un desperdicio de espacio tratar cada uno de los registros como si todos tuvieran su formulario lleno. Sin embargo todos los registros deben tener datos que sirvan para su administración (quién capturó el registro, cuándo, etc.).

En base a lo anterior, lo más conveniente es usar un modelo relacional, el cual se compone de múltiples tablas que pueden relacionarse entre sí mediante un identificador común, éstas relaciones enlazan diferentes tablas con distintos tipos de datos. Basándonos en un número único para cada registro (sin importar en que tipo de registro se ubique). A éste número único de registro le llamamos identificador (ID), que además de relacionar las tablas, será usado como índice [G3] para optimizar y facilitar búsquedas. Las 4 tablas con que se diseñó la base de datos fueron:

Tabla 1.- Tabla con los datos relacionados a la captura en el sistema, llamada T_Sistema, en esta tabla se encuentran los datos propios al control del registro, la fecha en que fue creado, cuándo fue la última vez que se modificó, el ID para poder relacionarlo a las demás tablas (todas las tablas deben llevarlo) entre otros datos que ayudan a tener un control administrativo sobre el registro.

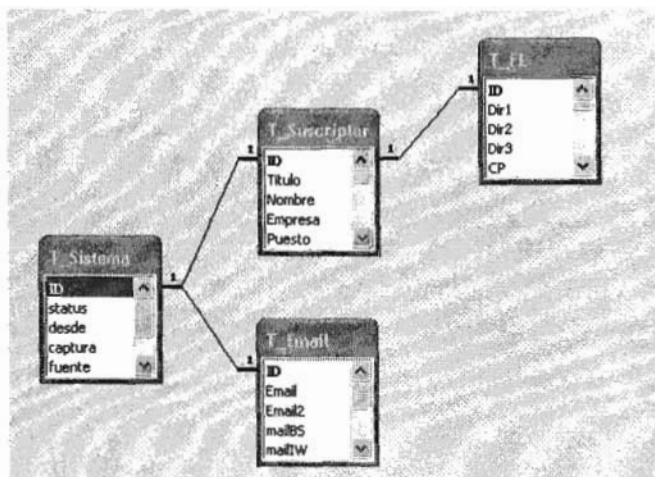
Tabla 2.- Tabla con los datos referentes al correo electrónico, conocida como T_Email, en ella se capturan los datos relacionados con todo el proceso de envíos electrónicos, empezando por la dirección de correo electrónico e indicadores de listas a las que esta suscrito.

Tabla 3.- Tabla referente a la persona suscrita, llamada T_Suscriptor, en ella se encuentran los datos propios del suscriptor como son: nombre, dirección, teléfono, código postal, etc.

Tabla 4.- Tabla con los datos obtenidos del formulario, conocida como T_FL, contiene las respuestas a las preguntas del formulario para conocer a detalle al suscriptor.

Partiendo del identificador único para cada suscriptor (ID), el esquema de relaciones de la base de datos quedaría de la siguiente manera:

¹² e-Newsletter *Boletín de noticias en formato electrónico*



La razón por la que se decidió diseñar así la base de datos es por que su estructura separa los diferentes tipos de registros y facilita su administración.

De está manera se había construido también la base original en Access, simplemente llevamos el mismo diseño a SQL Server para tener un sistema más estable y confiable.

En cuanto a los suscriptores a los que se les enviará "algo", la base de datos contendrá datos propios de un servicio de mensajería (el nombre de la persona, dirección, código postal, etc.) Dichos datos contendrían el nombre completo, la calle en la que se encuentra la empresa, número exterior e interior, la ciudad, el estado y el código postal, éstos son el mínimo número de datos necesarios para tener una etiqueta de correo, la cual se ve de la siguiente forma:

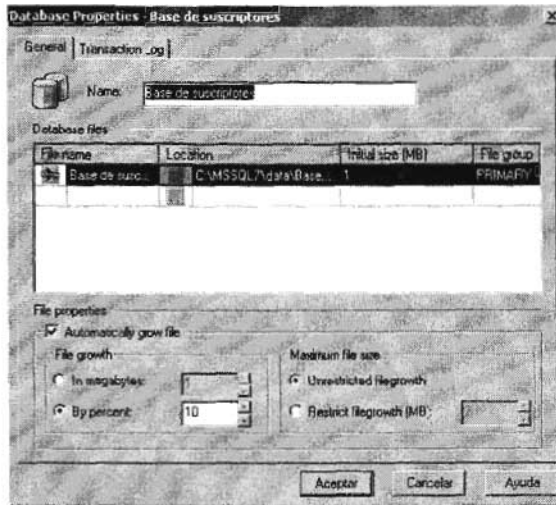
```

<<Nombre>>
<<Calle>><<Número Exterior>><<Número Interior>>
<<Código Postal>><<Ciudad>><<Estado>>
  
```

Usando los asistentes de SQL Server empezamos a programar la base de datos, escogemos iniciar con una nueva base de datos y aceptamos las opciones por omisión. Es importante seleccionar que el archivo pueda crecer automáticamente (automatically grow file), para no tener problemas de espacio conforme va creciendo la base de datos. La forma de manejar el crecimiento del archivo nos conviene que sea por porcentaje, sobre todo al principio por que no tenemos certeza de la forma en que se va a comportar el crecimiento de la base de datos. Asimismo, a pesar de que a ciencia cierta no sabemos hasta donde pudiera llegar el tamaño de la base de datos, tenemos un estimado de acuerdo a la experiencia con otras bases de datos que cada 10,000 registros ocupan aproximadamente 7.5 Mega bites de espacio en disco. De acuerdo a algunos cálculos de planeación estimamos que en unos 5 años de vida, la base de datos albergará aproximadamente entre cuarenta mil y cincuenta mil registros, que traduciéndolo a espacio en disco, los datos debieran ocupar entre 30 y 40 megas, que para las capacidades de los discos duros actuales no representan problema.

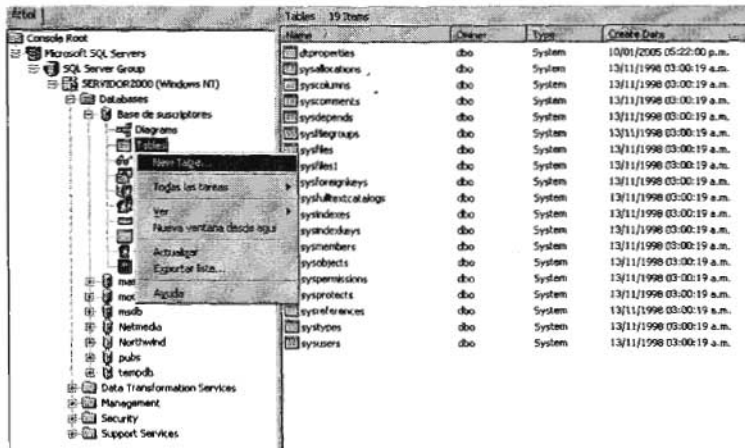
En la segunda pestaña del primer paso del asistente se encuentran las opciones para los archivos de transacción, los cuales contienen registros consecutivos de todas las transacciones [G4] que se han realizado en la base de datos desde la última vez que dicho archivo fue respaldado.

Manejando respaldos con registros de transacción es posible recuperar la base de datos desde un punto específico en el tiempo, por ejemplo, antes de que se presente una falla. [10]



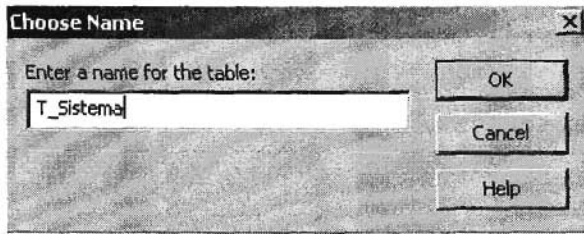
Vista del primer paso del asistente para crear la base de datos

Una vez que la base de datos fue dada de alta en SQL Server podemos empezar a trabajar en ella, lo primero que vamos a hacer es construir las tablas, partiendo del esquema de relaciones planteado de las tablas anteriores y un identificador (ID) el cual es un valor requerido sin posibilidad de duplicados, que nos servirá para relacionarlas, iniciamos el asistente para crear tablas y creamos la tabla llamada T_Sistema.



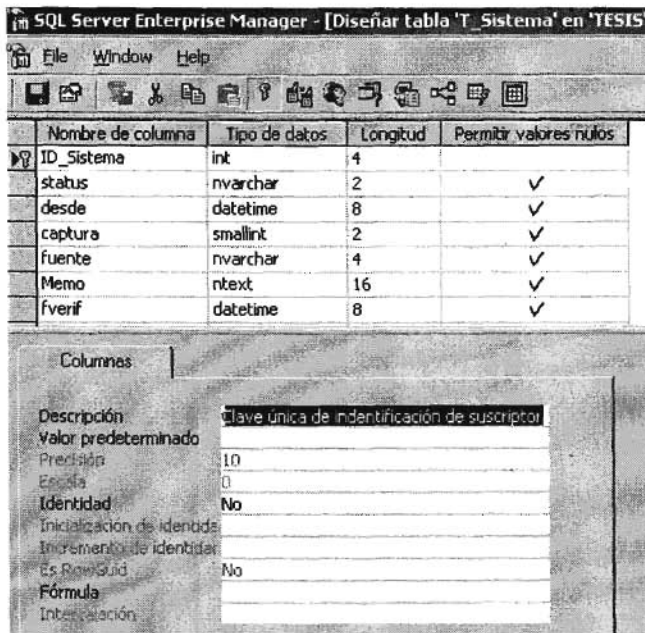
Vista del asistente para crear tablas en SQL Server

Primero ponemos el nombre de la tabla (el mismo proceso se repetirá para cada tabla)



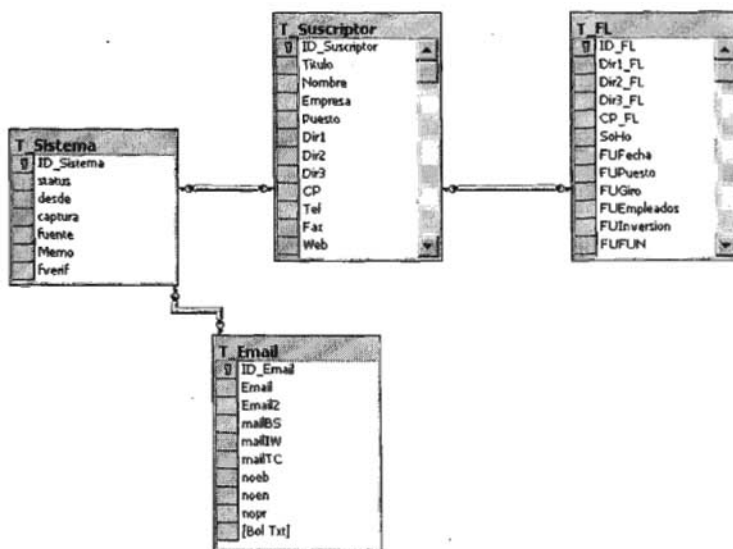
En esta parte del asistente definimos los nombres de las tablas

Posteriormente definimos los campos, la tabla finalizada se ve de la siguiente forma:



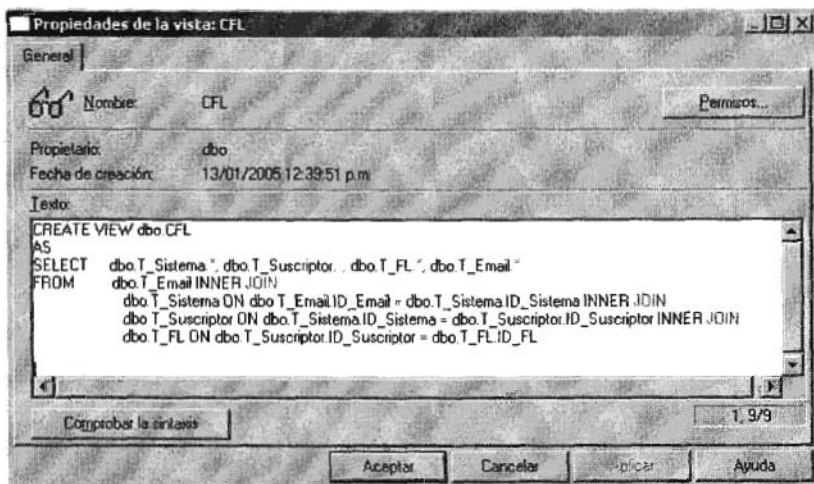
Ya con la tabla definida creamos los campos necesarios

Una vez que tenemos las cuatro tablas hacemos un diagrama para verificar que las relaciones sean correctas, con el asistente seleccionamos las 4 tablas, el diagrama queda de la siguiente manera:



Esquema de relaciones en SQL

Enseguida creamos una vista que presente las 4 tablas juntas, esta vista nos servirá cuando necesitemos toda la información junta de algún registro, la vista la podemos construir utilizando el asistente o directamente con sentencias SQL, en éste último caso, el código queda de la siguiente manera:



Vista CFL, muestra la información de las cuatro tablas

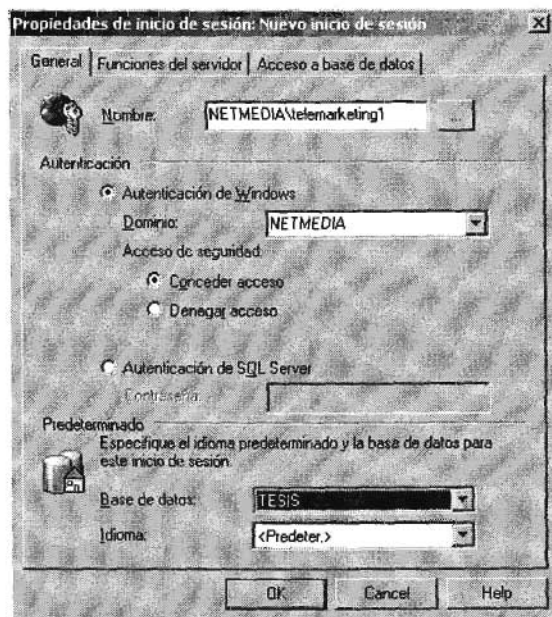
Lo que dice éste código es que se debe crear una vista llamada CFL, que seleccione las cuatro tablas y haga una operación de JOIN¹³ para formar la vista.

De manera adicional a estas cuatro tablas vamos a necesitar algunos catálogos que se desplegarán en los formularios de la base de datos, y que harán más sencilla la labor de los capturistas. Por ejemplo, necesitaremos varios catálogos, uno con los nombres de los usuarios que entran a la base de datos, el segundo con los giros de las empresas, otro con los posibles estados de un registro y en el último las claves que nos permiten ubicar el origen del registro.

Ahora creamos a los usuarios que vamos a necesitar, esto depende de las funciones que se van a realizar en la base de datos, SQL permite otorgar permisos específicos por usuario o por grupos sobre todas las bases de datos o sobre cada objeto en particular de cada una de las bases de datos que tengamos. Supongamos que creamos un usuario que se llame "telemarketing1", el cual puede ver todo lo que está en la base de datos TESIS, pero que no puede modificar nada, entonces utilizamos el asistente de **usuarios-crear nuevo usuario**, si estamos trabajando en un dominio de Windows, podemos usar la base de datos del directorio activo con los usuarios que ya tenemos creados, de esta forma la administración de la base de datos de usuarios sigue centralizada en el directorio activo, incluso podríamos usar grupos de usuarios ya creados y asignar los permisos desde ahí.

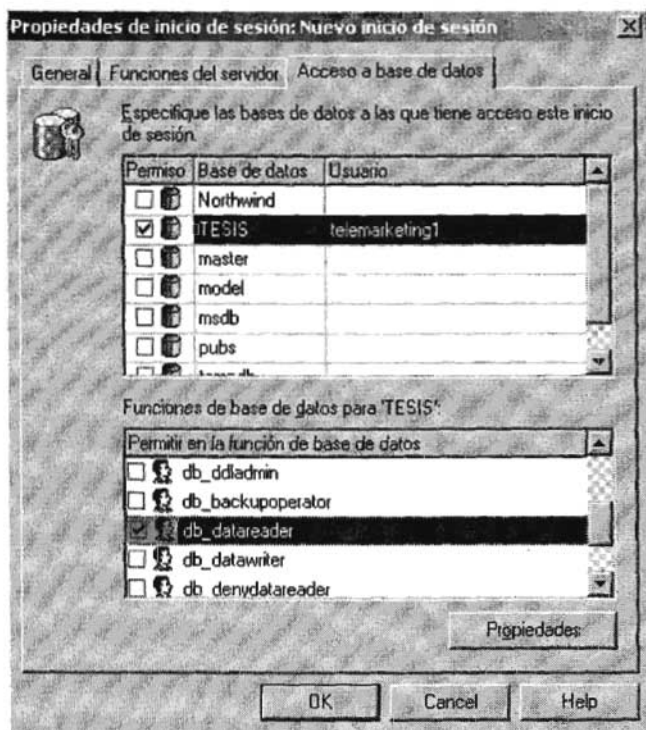
En la primer pantalla del asistente ponemos dichos datos, especificamos sobre el dominio NETMEDIA (así se llama nuestro dominio), al usuario *telemarketing1* con acceso a la base de datos TESIS.

La pantalla ya con los datos se ve de la siguiente forma:



13 Join Unión: Es una operación que combina registros en dos o más tablas de una base de datos relacional.

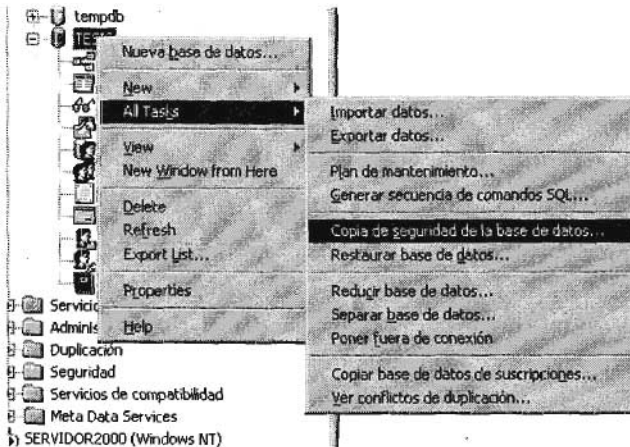
Ahora, en la tercer pestaña (Acceso a base de datos), especificamos que dicho usuario tendrá el rol de "db_datareader", el cual otorga permisos de sólo lectura sobre los objetos de la base de datos seleccionada. Esta parte del asistente se ve de la siguiente manera:



Asistente para asignar permisos.

Seguimos el mismo procedimiento para los demás usuarios que necesitemos. Por ejemplo, los capturistas tendrán el rol de "db_datawriter", el cual permite escribir información directamente en la base de datos.

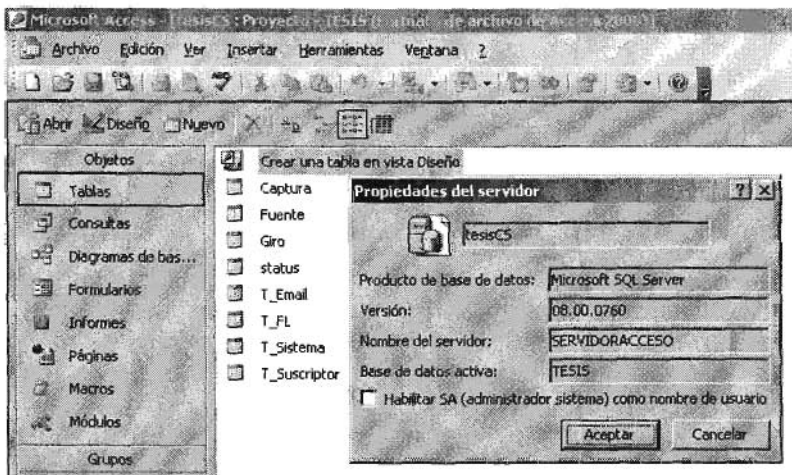
Ya que tenemos la base de datos tenemos que planear una forma de respaldarla para estar preparados en caso de alguna eventualidad. Cada base de datos debe ser analizada para ver cuál es la manera óptima de planear los respaldos, dos factores importantes a considerar para tomar esta decisión son: el tamaño de la base de datos y que tanto cambia en cierta unidad de tiempo. En nuestro caso el tamaño no es problema, lo que sigue es ver que tanto trabajo podría perderse de acuerdo a los cambios realizados en digamos, un día. Después de algunas simulaciones llegamos a la conclusión de que es buena idea respaldarla dos veces al día, suponiendo que las capturistas comenzaran su labor a las 9 de la mañana y terminarían a las 6 de la tarde, decidimos programar respaldos a las 2 pm y a las 7 pm. Programar los respaldos es muy sencillo usando el asistente de SQL Server, seleccionamos la base de datos, después "todas las tareas" y "copia de seguridad de la base de datos" como se ve en la siguiente figura:



Asistente para copias de seguridad.

Hablar de las diferentes estrategias de respaldos es un tema muy extenso y no es propiamente el objetivo de ésta tesis, en nuestro caso vamos a tomar una copia de la base completa (debido a que es relativamente pequeña en cuanto a tamaño). El "código" completo para generar la base de datos desde SQL Server se anexa en el Apéndice 5.

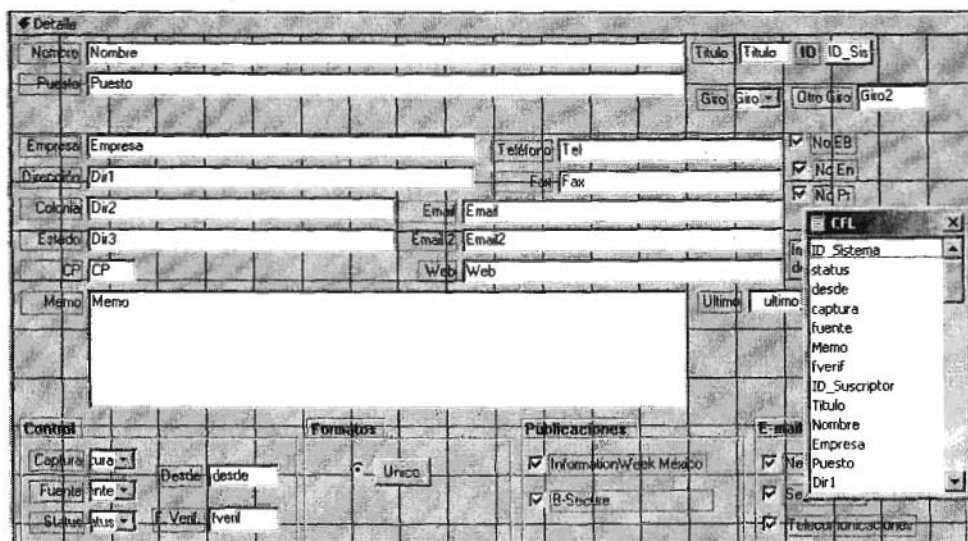
Lo siguiente es ver la forma en que los usuarios van a poder ver la información. La idea era crear una aplicación con la que los clientes tuvieran una interfaz amigable, flexible y funcional que les permitiera realizar fácilmente las tareas de captura, actualización y consulta de información. Usando los asistentes de Access, nos conectamos al servidor de SQL y especificamos que la base de datos a usar es TESIS y cargamos todo lo que hemos creado, las tablas creadas por nosotros (que están marcadas como tipo "usuario" en SQL Server, la vista CFL que también está marcada como tipo "usuario" y el diagrama de la base de datos. Ya en Access las tablas se ven así:



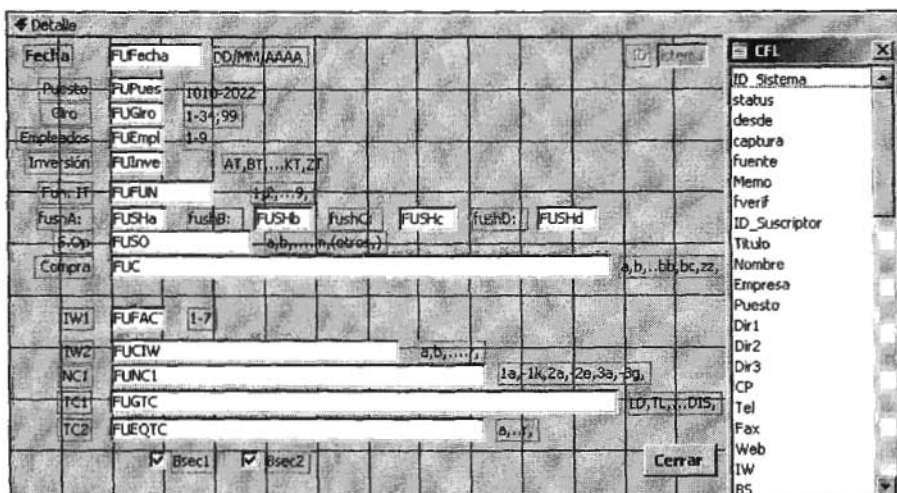
En este caso el servidor donde se está ejecutando SQL Server se llama SERVIDORACCESO.

Lo que sigue es crear los formularios que utilizarán los clientes, Access cuenta con un asistente muy amigable para ésta tarea, en la parte de "formularios" seleccionamos "crear un formulario en vista diseño", seleccionamos que el origen de los datos sea la vista CFL, se pensó en la necesidad de dos formularios, en uno de ellos se capturarán los datos generales del suscriptor y del registro en sí, contendrá datos de las tablas T_Sistema, T_Email y T_Suscriptor; a éste formulario le llamamos "Suscriptores". Por otro lado se creó otro formulario en el que caen los datos de la forma de suscripción, dicho formulario le llamamos "_funcio" y contiene datos de la tabla T_FL.

El diseño del formulario "Suscriptores" quedó de la siguiente forma:



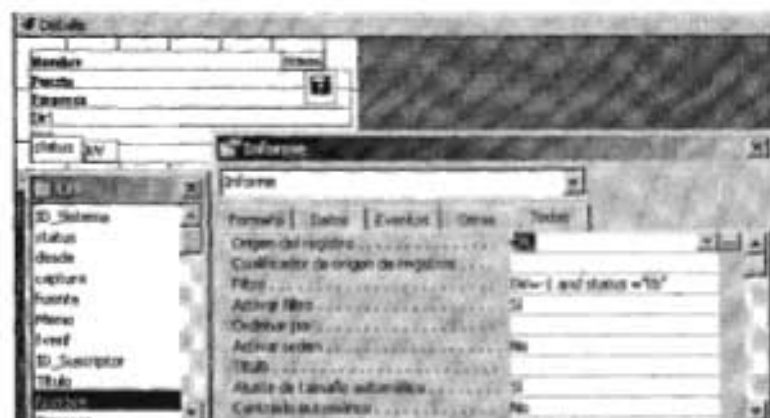
Y el del formulario "_funcio" de ésta forma:



Un aspecto muy importante en este sistema es la forma en que le debemos decir a la base de datos, que cada que se inserte un registro nuevo incremente el número de identificación (el número único de suscriptor), de la tabla T_Sistema (que es la que lleva el control de los registros) y que, al estar relacionada con las demás por un índice, va a incrementar automáticamente dichos números de identificación para cada una de las cuatro tablas.

Ya en éste punto los usuarios pueden entrar al sistema, sin embargo vamos a programar un par de informes, que serán de utilidad por ejemplo, cada que salga una publicación va a ser necesario imprimir las etiquetas para poder distribuir las revistas. Otra pregunta muy frecuente es saber cuántos suscriptores tenemos en cada empresa y cuántas publicaciones reciben cada uno.

Para el caso de las etiquetas, habrá un informe para cada publicación. Para las de InformationWeek, usamos el asistente para informes, su diseño es parecido al de los formularios, escogemos el origen de los datos (la consulta CFL) e insertamos los datos que nos interesan para poder armar una etiqueta típica de correo, después ponemos las condiciones necesarias para que el registro salga en el listado de etiquetas que básicamente son 2, que los datos del registro sean correctos (esto lo marcamos en el campo "status" y el valor "tb"), y que esté habilitada la entrega de esa publicación en particular (en éste caso el campo "IW" se marca con un "-1" para indicarlo). Visto en el asistente de diseño el informe con el correspondiente criterio de filtración (antes mencionado) el cual se puede visualizar de la siguiente manera:



De manera similar se construye el informe para los suscriptores por empresas, el diseño queda de la siguiente forma:

Evaluación del sistema									
Suscriptores X Empresa									Normal
Tratamiento de página									
Empres									
Detalle									
1	1	1	1	1	1	1	1	1	1
Pie de página									
Empres									
Pie de página									
Empres									
Pie de página									
Empres									

Lo que hace éste informe es que agrupa a los suscriptores de una misma empresa y regresa el número de suscriptores que hay para cada publicación.

De forma paralela cargamos en nuestros sitios web un formato de inscripción, el cual genera un correo electrónico el cual es capturado en nuestra base de datos. El diseño del formulario se divide en 2 partes, (1) el formulario en sí, que es el encargado de solicitar los datos y de enviarlos a un programa (2) que genera el correo electrónico y despliega un mensaje el cual indica que los datos fueron enviados con éxito. En los Apéndices 2 y 3 se anexan ambos archivos (formulario.php y correo.php).

Veamos primero las partes más importantes del primer archivo (formulario.php) Este archivo es el encargado de solicitar y validar los datos y si la información cumple con los criterios de validación, entonces será enviada al archivo que generará el correo electrónico.

Una manera sencilla para manipular la información es poniendo pequeñas claves a cada apartado. Por ejemplo, para la cuestión de número de empleados en la empresa creamos el siguiente catálogo:

Clave	Número de empleados
1	Más de 20,000
2	10,000-19,999
3	5,000-9,999
4	1,000-4,999
5	500-999
6	100-499
7	50-99
8	25-49
9	Menos de 25

La idea es plantear situaciones que nos puedan solicitar en un futuro, por ejemplo, tal vez necesitemos empresas de más de cien empleados, suponiendo que el campo se llamara "empleados", bastaría con hacer una consulta como 'empleados < 4'. Para responder a otras preguntas como ¿en qué funciones está usted involucrado? Usamos claves de este estilo:

Clave	Descripción
1,	Intranet/Internet
2,	Redes/Telecomunicaciones
3,	Desarrollo de aplicaciones

En general, todo lo que podemos catalogar lo hacemos, de ésta forma las tareas de segmentación de la base de datos se simplifican. Habrá otros campos en los que no será posible hacer un catálogo (por ejemplo en los nombres de las personas), pero no es relevante por el momento. Una vez que el formulario se ha completado, seguimos con la parte de las validaciones. Al momento de presionar el botón de "Enviar" del formulario, se hace la validación de la información capturada de acuerdo a nuestros requerimientos, la cual (posteriormente en caso de cumplir con la validación) será enviada al archivo correo.php, dicha acción se puede ver en el siguiente fragmento de código:

```
<form onsubmit="return ValidarForma()"
method="POST" action="http://www.netmedia.info/forma/correo.php" name="correo">
```

Seguimos con la parte de validación, lo que quiere decir éste fragmento de código es que al presionar el botón de enviar, se llama a la función ValidarForma y después (si se cumple la validación) se hace el envío de la información. De acuerdo a nuestros requerimientos más comunes, decidimos que los campos a validar son: nombre, año de nacimiento, correo electrónico, giro de la empresa y número de empleados de la empresa donde trabaja. Para hacer las validaciones creamos una función en Javascript la cual valida cada uno de los campos sujetos a validación (que llamamos "validarForma"), dicha función se compone de condiciones de éste estilo:

```
if (document.correo.n.value == ""){  
    alert ("Debe escribir su nombre");  
    return false}
```

Lo que hace ésta condicional es que en el formulario hay un campo que definimos como "n" y que es el correspondiente al nombre del suscriptor, la función establece que si el valor de "n" es nulo, el navegador envía una ventana de alerta con el mensaje "Debe escribir su nombre", con los que nos aseguramos de que el campo no éste vacío.

Funciones similares se usan para el año de nacimiento, el giro de la empresa y el número de empleados. Además, incluye una verificación extra que exige que el campo de correo electrónico tenga una "@" y al menos un punto. El código de la función es:

```
if((document.correo.m.value.indexOf("@")==-1)||document.correo.m.value.indexOf(".")==-1){  
    alert ("Debe escribir correctamente su email");  
    return false}
```

A continuación se presenta la versión final del formulario de suscripción (formulario.php) como se presenta en un navegador, también se incluye en los apéndices la versión impresa de dicho formulario (Apéndice 4)

Netmedia

INFORMATIONWEEK **b:Secure** Califique para una suscripción **GRATUITA!!!**

- Incluye además todos los beneficios
- En primer lugar, gratuita, el acceso a profesionales expertos en el sector que ayudan a mejorar la productividad de sus negocios, tanto en el sector público como en el privado.
- Así, además de a la oferta de un curso online de gestión de recursos humanos, el acceso a los cursos online de gestión de recursos humanos.

Noticias económicas • Inversión • Comercio electrónico • Gestión de recursos humanos

Datos generales para enviar las inscripciones:

Titulo:

Nombre:
 Apellidos:
 Apellido:
 Dirección:
 P.º:
 Código:
 Ciudad / País:

Teléfono:

Fax:

E-mail:
 Web:
 Fecha de nacimiento:

Enviar • Publicar

Enviar • Publicar

Ya que se han cumplido con todas las validaciones, el formulario es enviado al archivo correo.php que es el que generará el correo electrónico, éste archivo contiene instrucciones en PHP como lo es la función "mail"¹⁴, ésta función tiene la siguiente estructura:

14 La especificación de la función mail para php puede consultarse en: <http://mx2.php.net/manual/es/ref.mail.php>

55 53372734 Ext.
55 53372915 Ext.
daniel.ortega@bms.com
<http://www.bms.com.mx>
28/10/1970
H
6

Estos datos son los que se llevan a la hoja de cálculo auxiliar para ser estandarizados y posteriormente insertados en las tablas correspondientes en SQL Server.

En la base de datos capturamos todas las palabras en formato de título (la primer letra de cada palabra de una oración sea con mayúscula y el resto minúsculas, por ejemplo "Pedro Torres"), sin embargo una situación frecuente es que las solicitudes de inscripción lleguen en puras mayúsculas o en puras minúsculas, en la hoja auxiliar de Excel se programó una macro¹⁵ para ayudarnos con esa tarea, programamos un macro que queda de la siguiente manera:

```
Dim c As String  
Sub convierte()  
c = ActiveCell  
ActiveCell = StrConv(c, 3)  
End Sub
```

Primero se define una variable "c" de tipo string (texto), luego creamos una función que llamamos "convierte", la cual le asigna a la variable "c" el contenido de la celda actual en la hoja de cálculo, a continuación se usa una función almacenada de Visual Basic llamada *StrConv* que convierte el contenido de la variable "c" a tipo oración (mayúsculas y minúsculas), finalmente a la celda activa en la hoja de cálculo se le envía el nuevo valor de "c" y termina la función "convierte" creada por nosotros.

Veamos a detalle la función *StrConv*¹⁶:

StrConv devuelve un tipo texto que se convierte según se especifique. La sintaxis es:

```
StrConv(string, conversion)
```

String: Requerido. Es la expresión de cadena de texto que se va a convertir.

Conversion: Requerido. Un tipo entero, la suma de los valores que especifica el tipo de conversión a realizar. En nuestro caso el valor es 3, que convierte la primera letra en mayúscula de cada palabra de la cadena.

¹⁵ Macro: Una macro es un conjunto de instrucciones y/o combinaciones de secuencias de teclas que son grabadas y salvadas, a las cuales se les asigna un código de acceso corto. Cuando dicho código de acceso es llamado (por ejemplo, mediante una combinación de teclas como "control-p" o presionando algún icono asignado), las instrucciones almacenadas son ejecutadas. Los macros pueden simplificar tareas frecuentes que podrían resultar muy tediosas.

¹⁶ La definición completa de la función *StrConv* puede consultarse en la ayuda de Visual Basic.

El sistema prácticamente no ha requerido de modificaciones complejas desde que fue creado, los únicos cambios que ha sufrido corresponden a las necesidades que se van presentando en la empresa, por ejemplo agregar o remover algunos campos de acuerdo a cambios en productos que surgen o desaparecen, lo cual es muy sencillo de hacer en SQL Server. Asimismo los formularios de nuestros sitios web han servido perfectamente a nuestros intereses y no han requerido grandes cambios. Lo único que sí cambió fue el servidor en el que se encuentra la base de datos, originalmente teníamos un equipo con Windows 2000 Server para manejar la base de Access, pero los requerimientos de las nuevas tecnologías (Windows 2003 Server y SQL Server) necesitaban de un equipo más poderoso optamos por instalar en un nuevo servidor el software.

Capítulo 5.- El futuro del sistema

Hoy en día todas las empresas (sea cual sea su giro) deben mejorar constantemente para ser más competitivas y mantener o mejorar su posición en el mercado; uno de los retos principales del sistema de información es que debe ser capaz de apegarse a dichos cambios. Es indispensable contar con un sistema flexible que permita satisfacer las necesidades de información para la toma de decisiones.

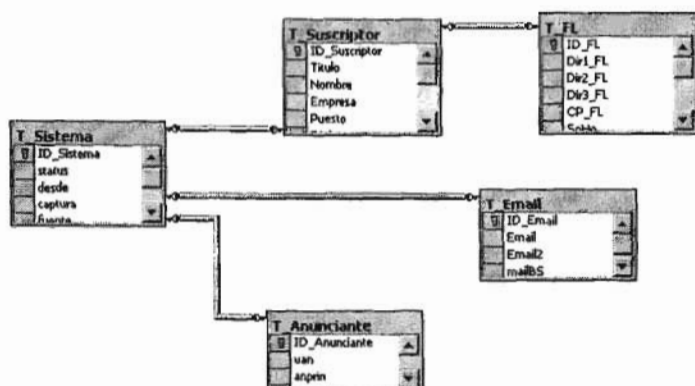
Una necesidad que hemos observado es el tener control sobre los vendedores, el supervisor de ventas necesita saber a qué clientes atiende cada uno de los vendedores y el estado en el que se encuentran sus cuentas (para no descuidar a ningún cliente) y con ello no perder posibles ventas por falta de atención. Por su parte, para los vendedores sería muy útil llevar un expediente de cada cliente, donde se pueda conocer los productos de la empresa en los que ésta interesado, cuándo fue la última vez que estuvo en contacto con el y cuándo le corresponde volver a contactarlo, en otras palabras, contar con un histórico de la atención que ha recibido cada cliente, y saber cuándo tiene que volver a contactarlo.

Ya que los clientes de la editorial son a su vez lectores de nuestras publicaciones, una opción es que dicha información se almacene en la misma base de datos que ya hemos construido, lo siguiente es adaptar la base de datos para que pueda manejar esta información, básicamente es una tarea parecida a la realizada para definir la estructura de la base de datos. Primero, hay que definir el tipo de información que requerirá el departamento de ventas y en base a ello especificar los datos que vamos a necesitar para llevar el control de los vendedores. Después de analizar con el departamento de ventas sus requerimientos, llegamos a la conclusión que los datos requeridos serán: el nombre del vendedor que está atendiendo al cliente, la última llamada realizada, la siguiente llamada programada, el tipo de productos de nuestra empresa en que están interesados, un campo para hacer anotaciones relacionadas con el producto y el cliente, otro campo en el que podamos asignar una probabilidad de venta y obviamente el campo del identificador para poder relacionarlo al resto de las tablas. La idea es que cada que el vendedor entre a la base de datos se active un procedimiento que busque si el vendedor que esta accediendo a la base de datos tenga agendada alguna llamada para ese día, y el sistema le avise mediante un mensaje en la pantalla que tiene pendiente dicha llamada. Cada que un vendedor se ponga en contacto con un cliente, las fechas de última llamada y de la siguiente deberán ser actualizadas, así como la información contenida en el campo de anotaciones para ver en que estado se encuentra la negociación con dicho cliente.

Por otro lado, ya con esta información se pueden preparar varios reportes para el supervisor de ventas, por ejemplo, podemos saber la cantidad de clientes que atiende un vendedor en una unidad de tiempo determinada, cuáles son los clientes que están descuidados, cuáles clientes son los más probables a cerrar una venta para brindarles atención especial, manejar promociones especiales para cada tipo de producto de la empresa, etc.

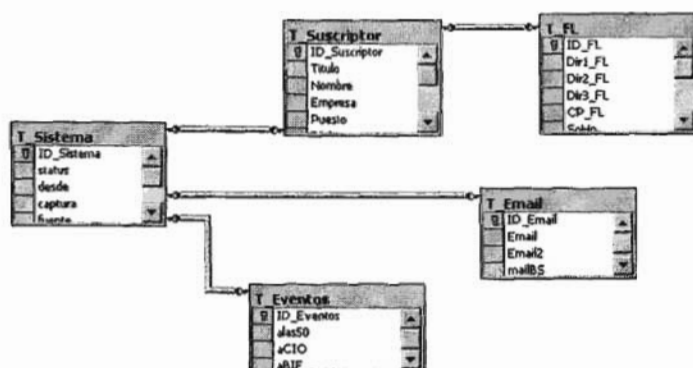
Ubicando esto en la base de datos lo que tendríamos que hacer es construir una tabla (siguiendo con la nomenclatura usada llamaremos T_Anunciante), con los campos solicitados la cual se construye de la

misma forma que las demás, mediante los asistentes de SQL Server, al incluirla en el esquema de relaciones queda de la siguiente forma:



Ya con esto, podemos modificar el formulario de captura (o crear otro) en la aplicación correspondiente, para almacenar los datos.

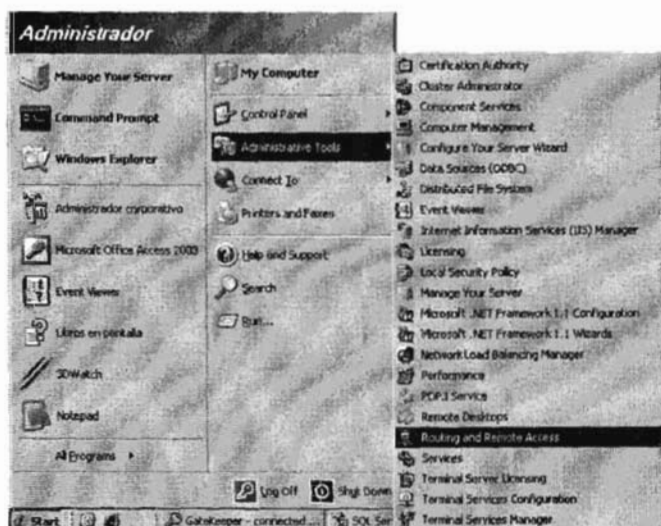
Una de las áreas de negocio que han crecido más en la empresa es la de los eventos. Una necesidad que ha surgido es la de llevar un histórico de los asistentes a nuestros diferentes eventos, algo muy similar a lo que se está haciendo para el control de anunciantes. Se analizaron los requerimientos y se llegó a la conclusión de hacer otra tabla, que debe llevar campos para cada uno de los eventos que hacemos (5 diferentes en el año), en los que se captura si el lector asistió al evento y si lo hizo por invitación o pagando su entrada, con ésta información disponible podemos planear mejor la forma en que convocamos gente a nuestros eventos, por ejemplo, podemos armar promociones especiales para los que han asistido con anterioridad, no volver a invitar a los que ya han sido invitados para cubrir segmentos diferentes de la base de datos, etc. La tabla en la base de datos se verá de la siguiente forma:



Cada que se organiza un evento, se hace una campaña de ventas por teléfono (que llamaremos **telemarketing**) para vender entradas o para confirmar a la gente que fue invitada. Observamos que se necesita una especie de "archivo" donde esté disponible la información sobre las llamadas realizadas al suscriptor. Una forma de hacerlo es crear una nueva tabla en la base de datos, la diferencia entre ésta tabla y las anteriores es que a un lector pueden hacerse más de una llamada (de hecho esa es la situación más común), a diferencia de las demás donde a un número de suscriptor le corresponde uno y sólo un, elemento de otra tabla, por ejemplo, a cada suscriptor le corresponde un único correo electrónico, un único formato de suscripción, etc. en otras palabras, estamos hablando de relaciones uno a uno, pero para el caso del telemarketing se le harán varias llamadas a un mismo suscriptor. Entonces debemos diseñar la tabla con el asistente de SQL Server especificando que el campo ID_Telemarketing de la tabla T_Telemarketing (así los llamaremos para continuar con la nomenclatura) permita duplicados, y lo ligamos a la tabla T_Sistema para hacer la relación correspondiente. También podemos programar alertas que le indiquen al agente de telemarketing las llamadas pendientes que tiene para el día. El supervisor de telemarketing puede llevar varios controles, por ejemplo, puede comparar productividad entre los diferentes vendedores, sacar promedios de duración de llamadas, llamadas realizadas al día, número de registros que deben ser tocados para lograr una venta, etc. con toda esa información se pueden planear las campañas de telemarketing para lograr los resultados esperados.

Otro aspecto que se está implementado son los accesos remotos. Varias áreas de la empresa necesitan consultar la base de datos desde fuera de la oficina (sobre todo los vendedores), la solución que estamos manejando es la implementación de un esquema de VPN que funciona de la siguiente manera:

Tenemos un servidor que tiene conectado un módem ADSL a 1 megabyte de velocidad, a éste equipo (con Windows 2003 como sistema operativo), se le habilitó la opción de "ruteo y acceso remoto" y así los usuarios se validan con su cuenta de Active Directory y usan los permisos de acceso desde fuera de la oficina para tener acceso a la base de datos y en general a los recursos de la red local. Para hacer esto, desde el menú de "herramientas administrativas" seleccionamos "Routing and Remote Access"

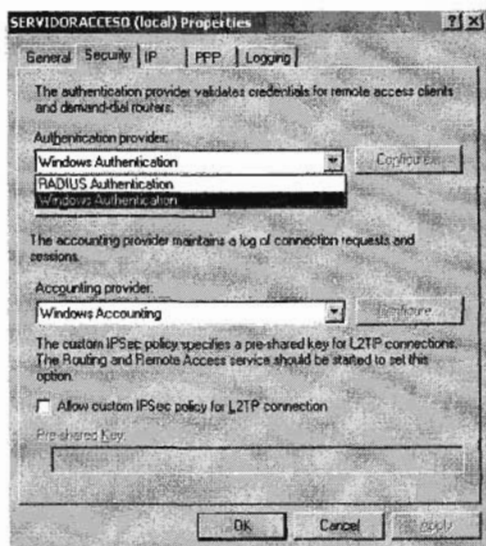


Desde aquí podemos configurar y administrar el acceso remoto.

Ya dentro de la consola de administración de ruteo y acceso remoto, configuramos el servicio, el asistente es muy amigable e intuitivo, los aspectos principales a tomar en cuenta son:

* Determinar si el servidor va a servir como servidor de acceso, ruteador y/o proxy de Internet, en nuestro caso hará funciones acceso para VPN y de ruteo.

* Determinar la manera en que validarán los clientes, puede ser con autenticación de Windows o mediante un servidor RADIUS¹³, en nuestro caso utilizaremos autenticación Windows para poder usar los permisos definidos en Active Directory. Para que el servidor de acceso remoto pueda leer los permisos de Active Directory es necesario dar de alta a dicho servidor como un miembro de servidores de RAS en Active Directory.



Escogemos autenticación Windows para validar a los usuarios.

* Seleccionar la manera en que se van a otorgar direcciones IP a los usuarios, puede ser por medio de un servidor de DHCP o seleccionando un rango estático de direcciones.

Ya con el servicio habilitado es posible entrar a la red local desde fuera de la oficina, la ventaja que esto trae consigo es que la información contenida en los servidores está disponible desde cualquier lugar, con ello se pueden hacer una gran cantidad de cambios administrativos en la editorial, la mayoría de las áreas de la empresa podrían trabajar sin necesidad de estar físicamente en la oficina y se tendrían ahorros para las dos partes, para la oficina en la parte de consumo de energía, reducción de espacio en la oficina, llamadas telefónicas, etc. Y para los trabajadores en traslado y otros factores.

13 Servidor RADIUS: Es un protocolo de cliente-servidor que permite a los servidores de acceso remoto comunicarse con un servidor central y autenticar a los usuarios que intentan acceder a un determinado sistema o servicio. RADIUS permite mantener perfiles de usuario en una base de datos central que los servidores de acceso remoto pueden consultar.

Conclusiones:

Es de vital importancia tener disponible la información para toma de decisiones en las empresas, gracias a los avances tecnológicos es posible manipular grandes cantidades de información en periodos de tiempo muy cortos, lo que nos lleva a tener un mejor control sobre las decisiones que tomaremos. La información debe estar concentrada en un sistema confiable y poderoso, al que podamos confiarle lo más valioso que tiene la empresa: **la información.**

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para lograr ventajas competitivas, tales como:

1. Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
2. Eliminación de información duplicada.
3. Permite compartir información. Varios sistemas o usuarios pueden utilizar una misma entidad.
4. Permite mantener la integridad en la información, es decir, sólo se almacena la información correcta.
5. Independencia de datos. La independencia de datos implica una división entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos o tener acceso a la base de datos de diferente manera, sin hacer cambios en las aplicaciones o en los programas mismos.

Hablando en particular del sistema de Netmedia, ha funcionado perfectamente a lo largo de sus cinco años de vida, ha logrado adaptarse de manera satisfactoria a los constantes cambios en las necesidades de la empresa. En un principio se utilizó una base de datos en Access, pero empezó a crecer mucho en registros por lo que comenzó a dar ciertos problemas, se buscó una solución más eficiente y es precisamente la que se planteó en esta tesis. El servidor de SQL es capaz de manejar volúmenes de información muy por encima de los que manejamos en Netmedia, el servidor web con Linux y Apache han probado ser muy estables y han ofrecido un excelente rendimiento para las operaciones que se realizan en nuestros sitios web.

Los procedimientos que se utilizan han ayudado a crear una base de datos estandarizada, lo que permite realizar consultas muy específicas de manera sencilla, los esquemas de seguridad implementados permiten que las diferentes áreas de la empresa tengan acceso directo a la información requerida sin depender de algún tercero y sin poner en peligro la integridad de los datos o el funcionamiento del sistema. La estrategia de respaldos garantiza que en caso de alguna eventualidad el sistema pueda ser restaurado rápidamente para tener continuidad en las tareas de la empresa. El sistema permite manipular grandes cantidades de registros simultáneamente, lo que facilita en gran medida la labor de actualización y verificación de la base de datos.

En general un sistema de información realiza cuatro actividades básicas, que son:

1. Entrada de información: Es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información.

2. Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes de una computadora, a través de esta propiedad el sistema puede recordar la información almacenada en el proceso anterior.

3. Procesamiento de información: Es la capacidad del sistema de información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Esta característica de los sistemas de información permite la transformación de los datos de origen en información útil para la toma de decisiones.

4. Salida de información: Es la capacidad del sistema para sacar la información procesada o bien datos de entrada al exterior (por ejemplo en nuestro caso, el listado de las etiquetas con los datos de los suscriptores).

El sistema de información implementado en Netmedia cumple a la perfección con éstas cuatro actividades, el proceso de entrada de información que se explica detalladamente en la presente tesis es muy eficiente, la manera en que se almacena la información garantiza un sistema estable, el procesamiento de la información permite sacar una gran cantidad de reportes que apoyan a la toma de decisiones y cuando necesitamos listados específicos la salida de información se lleva a cabo sin problema.

Si bien es cierto el sistema de información de Netmedia se usa para llevar control de los suscriptores a nuestras revistas, es lo suficientemente flexible como para ser adoptado fácilmente a otras empresas, en general cualquier empresa que recopile información de sus clientes podría implementar éste sistema con relativamente pocos cambios siempre y cuando disponga de las tecnologías pertinentes.

En base a lo expuesto en esta tesis, en el contexto global en que se mueven las organizaciones actualmente se necesita cada vez más de sistemas que apoyen la toma de decisiones y que mejoren la operación de los procesos, es por ello que los sistemas de información son indispensables, ya que ayudan a las organizaciones a almacenar información cierta, oportuna, estructurada y confiable; esto permite realizar análisis de información exactos y reducir costos, mejorar los procesos de las organizaciones, entre otros, por lo que permite obtener una ventaja competitiva en el mercado.

GLOSARIO

[G1] RAID5: Acceso independiente con paridad distribuida: Es un tipo de arreglo que ofrece tolerancia a fallos y que además optimiza la capacidad del sistema permitiendo una utilización de hasta el 80% de la capacidad del conjunto de discos. Esto lo consigue mediante el cálculo de información de paridad y su almacenamiento alternativo por bloques en todos los discos del conjunto. La información del usuario se guarda por bloques y de forma alternativa en todos ellos. De ésta manera, si cualquiera de las unidades de disco falla, se puede recuperar la información en tiempo real, sobre la marcha mediante una simple operación lógica de O exclusivo sin que el servidor deje de funcionar. A diferencia del RAID 4 que plantea el uso de un disco de comprobación, el RAID 5 no asigna un disco en particular, se distribuye la función de comprobación entre todos los discos y se disminuyen posibles cuellos de botella. RAID5 es un nivel muy eficaz y comparado con otras opciones en niveles RAID para tolerancia a fallos, ofrece una muy buena relación rendimiento-costo. [7]

[G2] IBM desarrolló la versión de SQL en sus laboratorios de investigación en San José. Implementó el lenguaje, originalmente llamado "Sequel", como parte de un proyecto llamado sistema R al principio de los 70's. El lenguaje Sequel ha evolucionado desde entonces y su nombre cambió a SQL (Structured Query language *Lenguaje de consultas estructurado*). Muchos productos soportan hoy el lenguaje SQL. SQL se ha establecido como el lenguaje estándar para bases de datos relacionales.

En 1986, el Instituto Nacional Americano de Estándares (ANSI) y la Organización Internacional para la Estandarización (ISO) publicaron el estándar de SQL llamado SQL-86.

El lenguaje SQL tiene varias partes:

*Data-definition language (DDL) *lenguaje de definición de datos*: provee comandos para definir esquemas de relación, borrar relaciones y modificar esquemas de relación.

*Interactive data-manipulation language (DML) *lenguaje interactivo de manipulación de datos*: incluye un lenguaje de consultas basado en álgebra y cálculo relacional.

*View definition *definición de vistas*: El DDL de SQL incluye comandos para definir vistas.

*Transaction control *control de transacciones*: SQL incluye controles para especificar el comienzo y el final de las transacciones.

*Embedded SQL y dynamic SQL *SQL integrado y SQL dinámico*: SQL integrado y dinámico definen la forma en que las instrucciones SQL puedan ser integradas con lenguajes de programación de propósito general, tales como C, C++, Java, etc.

*Integrity *Integridad*: El DDL de SQL incluye comandos para reglas específicas de integridad que la base de datos almacenada debe de satisfacer. Actualizaciones que violasen la integridad definida en las reglas son descartadas.

*Authorization *Autorización*: El DDL de SQL incluye comandos para especificar derechos de acceso a relaciones y vistas.

[G3] Índices: Asignar índices apropiados es una parte necesaria de una aplicación de base de datos eficiente. En realidad se pueden observar altos incrementos en el desempeño agregando índices útiles en las tablas de la base de datos. Este artículo introduce las bases de los índices agrupados o no

agrupados, y da algunas ideas y recomendaciones para seleccionar una apropiada estrategia de indexación.

Un índice es una estructura física que contiene apuntadores hacia los datos. SQL Server 2000 soporta hasta 250 índices por tabla, aunque generalmente se tendrán muchos menos por tabla. No está permitido construir índices en columnas con ciertos tipos de datos como TEXT, NTEXT e IMAGE.

Hay dos tipos de índices, los agrupados y los no agrupados. La estructura de índices de SQL Server a menudo se compara con la estructura de un árbol. Ambos índices, agrupados y no agrupados, tienen dos tipos de nodos: nodos de hoja y nodos de raíz. El nivel de hoja de un índice agrupado es la información en sí misma. El nivel de hoja de un índice no agrupado es un apuntador al nivel raíz del índice agrupado.

La clave del índice agrupado —es decir, la(s) columna(s) en la(s) cual(es) el índice es construido— determina la forma en que la información se ordena físicamente. Poniendo un ejemplo, si se construye un índice agrupado en la columna "estado" de la tabla "autores" en la base de datos "publicaciones", los datos serán ordenados basándose en los valores de "estado" —Ya sea de forma ascendente o descendente—. La instrucción CREATE INDEX asigna un orden ascendente por default, sin embargo se puede cambiar a descendente si es necesario.

Después de leer el párrafo anterior, no debe sorprendernos que sólo podamos tener un único índice agrupado por tabla —sólo se pueden ordenar los datos de una manera—. Dado que el nivel de hoja de un índice agrupado es la información en sí misma, leer los datos usando el índice agrupado es mucho más rápido que usando un índice no agrupado. Para tener un desempeño óptimo debemos de escoger la columna que será el índice agrupado con mucho cuidado.

Los índices agrupados deben ser únicos; dado que la información se ordena de acuerdo a los valores en la clave del índice agrupado. SQL Server debe tener una forma única de identificar la posición de cada registro en el índice. Si se presentan valores duplicados en la clave agrupada, SQL Server añadirá una columna oculta (clave) al índice para hacerlo único.

Los índices no agrupados no son tan eficientes como los agrupados; sin embargo, buscar sobre un índice no agrupado puede ser mucho más eficiente que buscar sobre toda la tabla. Dado que sólo podemos tener un índice agrupado por tabla, todos los demás deben ser no agrupados.

Para seleccionar las columnas que serán indexadas, se debe considerar el concepto de selectividad del índice, esto se refiere al número de valores distintos en la clave del índice. Mientras más selectivo sea el índice, mejor desempeño tendrá en relación a alguno que tenga pocos valores distintos en la clave de índice (por ejemplo, en una que contenga sólo 1s y 0s).

Un aspecto importante a considerar es que si la tabla tiene un número pequeño de registros (10,000 o menos), los índices no harán una gran diferencia. De hecho, si SQL Server puede buscar sobre toda la tabla más rápido de lo que haría sobre un índice, ignorará la búsqueda sobre dicho índice.

Ocasionalmente SQL Server pudiera tomar una decisión equivocada respecto a un índice, si estamos seguros de querer usar algún índice podemos forzarlo, simplemente se debe usar ésta instrucción:

```
SELECT column1  
FROM table_name WITH (INDEX = N)
```

Donde N es el índice que queremos usar.

En resumen, el uso adecuado de índices ayudan a incrementar el desempeño de las búsquedas en la base de datos. [11]

[G4] Transacción: El término transacción se refiere a una colección de operaciones que forman una única unidad lógica de trabajo. Por ejemplo, transferir dinero de una cuenta a otra es una transacción que consiste en dos actualizaciones, una a cada cuenta.

Es importante que tanto, todas las acciones de una transacción puedan ser ejecutadas, o en caso de falla, los efectos parciales de una transacción puedan ser deshechos. A esta propiedad se le conoce como atomicidad. Una vez que la transacción se ejecutó con éxito, sus efectos deben perdurar en la base de datos, a esta propiedad se le conoce como durabilidad.

En un sistema de bases de datos donde ocurren múltiples transacciones simultáneamente, si los cambios en la información compartida no son controlados existe el riesgo de que quede en estado inconsistente debido a actualizaciones de otras transacciones, los sistemas de bases de datos deben tener mecanismos para aislar las transacciones de los efectos de otras transacciones concurrentes. A esta propiedad se le llama aislamiento.

[G5] El modelo OSI (Open systems interconnection *Interconexión de sistemas abiertos*) es el programa de estandarización internacional creado por la ISO y la ITU-T para desarrollar estándares para las redes de datos que faciliten la interoperabilidad de equipos fabricados por diferentes proveedores.

El modelo OSI se compone de siete capas.

Capa física (capa 1): Esta capa define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas de redes de comunicaciones. Las especificaciones de la capa física definen características como niveles de voltaje, temporización de cambios de voltaje, velocidades de transferencia de información, etc.

Capa de enlace de datos (capa 2): Proporciona el tránsito confiable de datos a través del enlace de red. Diferentes especificaciones de la capa de enlace de datos definen diferentes características de red y protocolo, incluyendo el direccionamiento físico, la topología de red, la notificación de error, la secuencia de tramas y el control de flujo.

Capa de red (capa 3): Esta capa proporciona el ruteo y funciones relacionadas que permiten a múltiples enlaces de datos combinarse en una red. Esto se logra a través del direccionamiento lógico (como opuesto al direccionamiento físico) de los dispositivos. Los protocolos de capa de red son de hecho protocolos de ruteo. Algunos ejemplos son RIP o OSPF.

Capa de transporte (capa 4): Implementa servicios confiables de datos entre redes, transparentes a las capas superiores. Entre las funciones habituales de la capa de transporte se encuentran el control de flujo, la administración de circuitos virtuales y la verificación y recuperación de errores. Algunas implementaciones de la capa de transporte incluyen el protocolo de control de transmisión, el protocolo de enlace de nombres y protocolos de transporte del estándar OSI. TCP (protocolo de control de transmisión) es el protocolo en el conjunto TCP/IP que proporciona una transmisión confiable de datos.

Capa de sesión (capa 5): Establece, administra y finaliza las sesiones de comunicación entre las entidades de la capa de presentación. Las sesiones de comunicación constan de solicitudes y respuestas de servicio que se presentan entre aplicaciones ubicadas en diferentes dispositivos de red. Estas solicitudes y respuestas están coordinadas por protocolos implementados en la capa de sesión. Un

ejemplo de implementación en la capa de sesión es AppleTalk que coordina el proceso de enlace de nombres.

Capa de presentación (capa 6): Brinda una gama de funciones de codificación y conversión que se aplican a los datos de la capa de aplicación. Estas funciones aseguran que la información enviada desde la capa de aplicación de un sistema sea legible por la capa de aplicación de otro sistema.

Capa de aplicación (capa 7): Esta es la capa más cercana al usuario final, lo cual significa que tanto la capa de aplicación como el usuario interactúan de manera directa con la aplicación de software.

Esta capa interactúa con las aplicaciones de software que implementan un componente de comunicación. Dichos programas de aplicación están fuera del alcance del modelo OSI. Las funciones de la capa de aplicación incluyen la identificación de socios de comunicación, la determinación de la disponibilidad de recursos y la sincronización de la comunicación.

Hay dos tipos clave de implementaciones de la capa de aplicación: las aplicaciones TCP/IP y las aplicaciones OSI. Las primeras son protocolos, como Telnet, FTP y SMTP; éstos forman parte del grupo de protocolos de Internet. Las aplicaciones OSI son protocolos como FTAM, VTP que pertenecen al conjunto OSI.

La comunicación entre dos equipos (A y B), sucedería de la siguiente forma de acuerdo al modelo OSI. Una aplicación en software en el sistema A tiene información para transmitir a una aplicación en software en el sistema B, el programa de aplicación del sistema A transferirá su información a la capa de aplicación (capa 7) del sistema A. Esta, entonces, transferirá la información a la capa de presentación (capa 6), la cual transferirá la información a la capa de sesión (capa 5), y así sucesivamente hasta la capa física (capa 1). En esta última, la información se coloca en el medio de transmisión de la red física y se envía al sistema B. La capa física del sistema B quita la información del medio físico y, posteriormente, su capa física transfiere la información hasta la capa de enlace de datos (capa 2), que la transfiere hacia la capa de red (capa 3), y así sucesivamente hasta que la información llega a la capa de aplicación (capa 7) del sistema B. Finalmente, esta última capa transfiere la información al programa de aplicación receptor para completar el proceso de comunicación.

[G6] La arquitectura de TCP/IP consta de cinco niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

Aplicación: Se corresponde con los niveles OSI de aplicación, presentación y sesión. Aquí se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de ficheros (FTP), conexión remota (TELNET) y otros más recientes como el protocolo HTTP (*Hypertext Transfer Protocol*).

Transporte: Coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.

Internet: Es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.

Físico : Análogo al nivel físico del OSI.

Red : Es la interfaz de la red real. TCP/IP no especifica ningún protocolo concreto, así es que corre por las interfaces conocidas, como por ejemplo: 802.2, CSMA/CD, X.25, etc.

Apéndice

- 1.- Características de los equipos seleccionados.**
- 2.- Formato de inscripción en PHP (Ver archivo formulario.php)**
- 3.- Programa generador del correo electrónico (ver archivo correo.php)**
- 4.- Formato de inscripción (Ver archivo forma.pdf)**
- 5.- Script para generar la base de datos desde SQL Server.**

Apéndice 1.- Características de los equipos seleccionados.

Los equipos seleccionados para la infraestructura de Netmedia tienen actualmente las siguientes características:

Servidores

Los 3 servidores cuentan con: procesador Intel Xeon a 2.4 Ghz. 1 Gb. de memoria RAM, 2 discos SCSI de 70 GB en arreglo de RAID 5.

Enlace a Internet

Módem Alcatel ASDL a 1 Mb.

Red LAN

Switch 3COM 48 puertos fase ethernet 10/100 Mb.

Estaciones de trabajo

Pentium 4 a 2 Ghz. 256 Mb. de memoria RAM, disco duro de 30 Gb.

Apéndice 2.- Formato de inscripción en PHP (Ver archivo formulario.php)

```
<html>
<head>
<title>Forma de suscripción; n. Netmedia Publishing.</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
function ValidarForma() {

//Verifica campos vacíos

if (document.correo.n.value == ""){
    alert ("Debe escribir su nombre");
    return false}

if (document.correo.m.value == ""){
    alert ("Debe escribir su email");
    return false}

if (document.correo.g2.value == "V"){
    alert ("Debe escribir el giro de su empresa");
    return false}

if (document.correo.e3.value == "V"){
    alert ("Debe escribir el número de empleados en su empresa");
    return false}

if (document.correo.dga.value == ""){
    alert ("Por favor escriba su año de nacimiento");
    return false}

//Inversiones

if (document.correo.i4.value == ""){
    alert ("Por favor escriba el total de su inversión en equipo (pregunta 4)");
    return false}

//Verifica e-mail

if((document.correo.m.value.indexOf("@")!=-1)&&(document.correo.m.value.indexOf(".")!=-1)){
    alert ("Debe escribir correctamente su email");
    return false}

} //Fin de función

</script>
</head>

<body bgcolor="#FFFFFF">
<p>
</p>
<p> 
<br>
</p>
<form onsubmit="return ValidarForma()" method="POST" action="http://www.netmedia.info/forma/s.php" name="correo">
<table border="0" width="610" cellspacing="0" cellpadding="0">
<tr>
<td height="2" align="CENTER" valign="BOTTOM"></td>
</tr>
<tr>
<td bgcolor="#336699" height="13"><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=2>
<ul>
<li>Por favor conteste todas las preguntas. </li>
<li><font color=#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="2">Las
suscripciones gratuitas están limitadas a profesionales residentes
en México que cumplen los criterios de calificación de cada publicación;
```



```

<div align="right"><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">T&iacut;ulo:&nbsp;</font></b></font>
<font color="#FFFFFF"></font></b></font></div>
</td>
<td width="184" height="34" bgcolor="#336699"> <font color="#000000" face="Verdana, Arial, Helvetica, sans-serif"
size="2"><b>
<div align="center">
<select name="t">
<option selected>---Seleccione---
<option value="Lic.">Licenciado
<option value="Ing.">Ingeniero
<option value="Doc.">Doctor
<option value="Prof.">Profesor
<option value="Act.">Actuario
<option value="C.P.">Contador
<option value="Mtro.">Maestro
<option value="Arq.">Arquitecto
<option value="Quim.">Qu&iacut;mico
<option value="M.en C.">Maestro en Ciencias
<option>Ninguno
</select>
</div>
</b></font></td>
<td width="1" height="34" bgcolor="#336699">&nbsp;</td>
<td height="34" colspan="3" bgcolor="#336699">
<p> <font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font color="#FFFFFF">
Otro<font size="1"> (Especifique):</font></font></b></font>
<input type="text" name="ot" size="30">
</p>
</td>
</tr>
<tr>
<td colspan="2">
<div align="center"><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Nombre:</font></b></font></div>
</td>&nbsp;</td>
<td colspan="3">
<input type="TEXT" name="n" size="50">
</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Empresa:</font></b></font></div>
</td>
<td colspan="3">
<input type="TEXT" name="e" size="50">
</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> <font size=2><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">Puesto:
</font><font
color=#000000 face="Verdana, Arial, Helvetica, sans-serif"> </font></b></font>
</div>
</td>
<td colspan="3">
<input type="TEXT" name="p" size="50">
</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> <font color=#000000

```

```

        face="Verdana, Arial, Helvetica, sans-serif" size=2><b><font color="#FFFFFF">Área:
</font></b></font> </div>
</td>
<td colspan="3">
    <input type="TEXT" name="a" size="50">
</td>
</tr>
<tr>
<td colspan="3">
    <div align="CENTER"> <font size=2><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif">Dirección</font></b></font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">(oficina)</font><font
color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"><b>:</b></font><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"><b></b></font></font>
    </div>
</td>
<td colspan="3">
    <input type="TEXT" name="d1" size="50">
</td>
</tr>
<tr>
<td colspan="3">
    <div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">C.P.:&nbsp;</font></b></font> </div>
</td>
<td colspan="3">
    <input type="TEXT" name="CP" size="50">
</td>
</tr>
<tr>
<td colspan="3">
    <div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Colonia:&nbsp;</font></b></font> </div>
</td>
<td colspan="3">
    <input type="TEXT" name="c" size="50">
</td>
</tr>
<tr>
<td colspan="3">
    <div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Ciudad
/ Población:</font></b></font> </div>
</td>
<td colspan="3">
    <input type="TEXT" name="cd" size="50">
</td>
</tr>
<tr>
<td colspan="3">
    <div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Estado:&nbsp;</font></b></font> </div>
</td>
<td colspan="3">
<select name="ed">
<option selected>-----Seleccione un Estado-----</option>
<option value="Ags.">Aguascalientes</option>
<option value="B.C.N.">Baja California Norte</option>
<option value="B.C.S.">Baja California Sur</option>
<option value="Camp.">Campeche</option>
<option value="Chis.">Chiapas</option>
<option value="Chih.">Chihuahua</option>
<option value="Coah.">Coahuila</option>
<option value="Col.">Colima</option>
<option value="D.F.">Distrito Federal</option>
<option value="Dgo.">Durango</option>
<option value="Gto.">Guanajuato</option>

```

```

<option value="Gro.">Guerrero</option>
<option value="Hgo.">Hidalgo</option>
<option value="Jal.">Jalisco</option>
<option value="M&eacute;x.">M&eacute;xico</option>
<option value="Mich.">Michoac&aacute;n</option>
<option value="Mor.">Morelos</option>
<option value="Nay.">Nayarit</option>
<option value="N.L.">Nuevo Le&oacute;n</option>
<option value="Oax.">Oaxaca</option>
<option value="Pue.">Puebla</option>
<option value="Qro.">Quer&eacute;taro</option>
<option value="Q.Roo.">Quintana Roo</option>
<option value="S.L.P.">San Luis Potos&iacute; </option>
<option value="Sin.">Sinaloa</option>
<option value="Son.">Sonora</option>
<option value="Tab.">Tabasco</option>
<option value="Tam.">Tamaulipas</option>
<option value="Tlax.">Tlaxcala</option>
<option value="Ver.">Veracruz</option>
<option value="Yuc.">Yucat&aacute;n</option>
<option value="Zac.">Zacatecas</option>
</select>
</td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER">
<p><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Teléfono:&nbsp;</font></b></font> </p>
</div>
</td>
<td width="112"><div align="center"><font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif"><strong>&nbsp;</strong></font>
<input name="te1" type="text" id="te15" size="3" maxlength="3">
<strong><font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif">&nbsp;</font></strong><br>
<font color="#FFFFFF" size="1" face="Arial, Helvetica, sans-serif"><strong>LADA
(2 &oacute; 3 d&iacute;gitos)</strong></font></div>
</td>
<td width="127"> <div align="center">
<font color="#FFFFFF" size="1">
<strong>
<input name="te2" type="text" id="te22" size="10" maxlength="20">
<font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif">+</font><br>
<font face="Arial, Helvetica, sans-serif">N&uacute;mero (7 &uacute; 8
d&iacute;gitos)</font></strong></div>
</td>
<td width="104"><div align="center">
<input name="te3" type="text" id="te3" size="7" maxlength="15">
<font color="#FFFFFF" size="2" face="Arial, Helvetica, sans-serif"><strong><br>
<font size="1"> Extensi&oacute;n</font></strong></font><font size="1"> </font></div>
</td>
</tr>
<tr>
<td colspan="3">&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td colspan="3"><div align="center">
<p><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Fax:&nbsp;</font></b></font> </p>
</div></td>

```



```

<td><div align="center"><font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif"><strong>&nbsp;</strong></font>
<input name="f1" type="text" id="te14" size="3" maxlength="3">
<strong><font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif">&nbsp;</font></strong><br>
<font color="#FFFFFF" size="1" face="Arial, Helvetica, sans-serif"><strong>LADA
(2 &ocute; 3 d&iacute;gitos)</strong></div> </td>
<td width="127"><div align="center"><font color="#FFFFFF" size="1"><strong><input name="f2" type="text" id="te22"
size="10" maxlength="20">
<font color="#FFFFFF" size="4" face="Arial, Helvetica, sans-serif">+</font><br>
<font face="Arial, Helvetica, sans-serif">N&uacute;mero (7 &uacute; 8
d&iacute;gitos)</font></strong></font></div>
</td>
<td width="104"><div align="center">
<input name="f3" type="text" id="f3" size="7" maxlength="15">
<font color="#FFFFFF" size="2" face="Arial, Helvetica, sans-serif"><strong><br>
<font size="1"> Extensi&ocute;n</font></strong></font><font size="1"> </font></div>
</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> </div>
</td>
<td colspan="3">&nbsp;</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> <font size=2><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">E-
mail:&nbsp;</font></b></font>
</div>
</td>
<td colspan="3">
<input type="TEXT" name="m" size="50">
</td>
</tr>
<tr>
<td colspan="3">
<div align="CENTER"> <font size=2><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-
serif">Web:</font></b></font>
</div>
</td>
<td colspan="3">
<input type="TEXT" name="w" size="50" value="http://">
</td>
</tr>
<tr>
<td colspan="3">
<div align="center"><font size=2><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif">Fecha
de nacimiento:</font></b></font></div>
</td>
<td colspan="3">
<select name="dgdia">
<option selected>---DIA---</option>
<option value="01">1</option>
<option value="02">2</option>
<option value="03">3</option>
<option value="04">4</option>
<option value="05">5</option>
<option value="06">6</option>
<option value="07">7</option>
<option value="08">8</option>
<option value="09">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>

```



```

Dirección de oficina (arriba)<br>
<input name=en type=radio
value=C>
Dirección de casa (abajo)</font></p>
</blockquote>
</td>
</tr>
</table>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr>
<td>
<div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Dirección&nbsp; <font size=1
color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">(o
apartado postal)<b>:</b></font> </div>
</td>
<td>
<input type="TEXT" name="d2" size="40">
</td>
</tr>
<tr>
<td>
<div align="CENTER"> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b></b></font> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">C.P.:&nbsp;</font></b></font> </div>
</td>
<td>
<input type="TEXT" name="cp2" size="40">
</td>
</tr>
<tr>
<td>
<div align="CENTER"> <font color=#000000
face="Arial, Helvetica, Times New Roman" size=2><b><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">
</font></b></font><font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF">Colonia:</font></b></font></div>
</td>
<td>
<input type="TEXT" name="co2" size="40">
</td>
</tr>
<tr>
<td>
<div align="CENTER"> <font color=#000000
face="Arial, Helvetica, Times New Roman" size=2><b><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">Ciudad
/ Población:</font></b></font> <font color=#000000
face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><font color="#FFFFFF"></font></b></font></div>
</td>
<td>
<input type="TEXT" name="cd2" size="40">
</td>
</tr>
<tr>
<td>
<div align="CENTER"> <font color=#000000
face="Arial, Helvetica, Times New Roman" size=2><b><font face="Verdana, Arial, Helvetica, sans-serif"
color="#FFFFFF">E</font><font face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">stado:</font><font
color="#FFFFFF">&nbsp;</font></b></font>
</div>
</td>
<td>
<select name="e2">

```

```

<option value="Estado" selected>-----Seleccione un Estado-----</option>
<option value="Ags.">Aguascalientes</option>
<option value="B.C.N.">Baja California Norte</option>
<option value="B.C.S.">Baja California Sur</option>
<option value="Camp.">Campeche</option>
<option value="Chis.">Chiapas</option>
<option value="Chih.">Chihuahua</option>
<option value="Coah.">Coahuila</option>
<option value="Col.">Colima</option>
<option value="D.F.">Distrito Federal</option>
<option value="Dgo.">Durango</option>
<option value="Gto.">Guanajuato</option>
<option value="Gro.">Guerrero</option>
<option value="Hgo.">Hidalgo</option>
<option value="Jal.">Jalisco</option>
<option value="M&eacute;x.">M&eacute;xico</option>
<option value="Mich.">Michoac&aacute;n</option>
<option value="Mor.">Morelos</option>
<option value="Nay.">Nayarit</option>
<option value="N.L.">Nuevo Le&oacute;n</option>
<option value="Oax.">Oaxaca</option>
<option value="Pue.">Puebla</option>
<option value="Qro.">Quer&eacute;taro</option>
<option value="Q.Roo">Quintana Roo</option>
<option value="S.L.P.">San Luis Potos&iacute;s</option>
<option value="Sin.">Sinaloa</option>
<option value="Son.">Sonora</option>
<option value="Tab.">Tabasco</option>
<option value="Tamps.">Tamaulipas</option>
<option value="Tlax.">Tlaxcala</option>
<option value="Ver.">Veracruz</option>
<option value="Yuc.">Yucat&aacute;n</option>
<option value="Zac.">Zacatecas</option>
</select>
</td>
</tr>
</table>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1><br>
Si usted trabaja en casa para una empresa, por favor escriba ambas
direcciones. Si la direcci&eacute;n de su casa es la misma de su negocio
por favor marque aqu&iacute;.</font>
<input name=sh
type=checkbox value=-1>
</p>
</blockquote>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF"></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td>
<p></p>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1><b> ¿Recibe alguna

```



```

<blockquote>
<p><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>2.&quest;C&uacute;al
es el giro principal de su empresa?</b></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p>
<select name="og2" size="1">
<option value="V" selected>-----Seleccione el giro de su empresa-----</option>
<option value="1">Automotriz</option>
<option value="2">Procesos/Manufactura</option>
<option value="3">Banca/Finanzas/Casa de bolsa</option>
<option value="4">Seguros/Fianzas</option>
<option value="5">Gobierno</option>
<option value="6">Medicina/Farmac&eacute;utica/Laboratorio</option>
<option value="7">Comunicaciones (Telecom/Telefon&iacute;a/ISP/Etc.)</option>
<option value="8">Transportaci&oacute;n/Mensajer&iacute;a</option>
<option value="9">Servicios p&uacute;blicos (Agua/Luz/Etc.)</option>
<option value="10">Construcci&oacute;n/Arquitectura/Ingenier&iacute;a</option>
<option value="11">Servicios de procesamiento de datos</option>
<option value="12">Comercio/Ventas/Renta/Distribuci&oacute;n. (NO
IT)</option>
<option value="13">Educa&iacute;n</option>
<option value="14">Agricultura/Ganader&iacute;a/Pesca</option>
<option value="15">Miner&iacute;a/Petr&oacute;leo/Qu&iacute;mica</option>
<option value="16">Viajes/Recreaci&oacute;n/Turismo/Hoteles</option>
<option value="17">Marketing/Publicidad</option>
<option value="18">Entretenimiento</option>
<option value="19">Investigaci&oacute;n y desarrollo</option>
<option value="20">Fabricaci&oacute;n de PC/Software/Accesorios/Consumibles</option>
<option value="21">Consultor&iacute;a de computadoras/redes</option>
<option value="22">Distribuidor de computadoras y equipo relacionado</option>
<option value="23">VAR de sistemas o integrador de redes</option>
<option value="24">Alimentos</option>
<option value="25">Relaciones P&uacute;blicas</option>
<option value="26">Servicios para negocios</option>
<option value="27">Prensa/Portales/Editorial</option>
<option value="28">Art&iacute;culos p/Oficina (Papeler&iacute;a/Muebles/Etc.)</option>
<option value="29">Maquinaria</option>
<option value="30">Aparatos El&eacute;ctricos</option>
<option value="31">Organizaci&oacute;n sin fines de lucro</option>
<option value="32">Consultor&iacute;a (no relacionado con IT)</option>
<option value="33">Despachos/Legal/Contable/Arrendamiento</option>
<option value="34">Bienes Ra&iacute;ces</option>
<option value="99">Otro</option>
</select>
</p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color="#FFFFFF"><b>Otro (especifique)</b></font>
<input type="TEXT" name="og2" size="50">
</p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">

```

```

<tr bgcolor="#FFFFFF">
<td>
<p></p>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1><b>3. ¿Cuál es
el número aproximado de empleados en su empresa?</b></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p>
<select name="e3">
<option value="v" selected>---Seleccione el número de empleados---</option>
<option value="01">Mas de 20,000</option>
<option value="02">10,000-19,999</option>
<option value="03">5,000-9,999</option>
<option value="04">1,000-4,999</option>
<option value="05">500-999</option>
<option value="06">100-499</option>
<option value="07">50-99</option>
<option value="08">25-49</option>
<option value="09">Menos de 25</option>
</select>
</p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="670" bgcolor="#336699" cellspacing="0" cellpadding="0">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1><b>4. ¿Cuál es
su inversión total anual proyectada para equipo de cómputo/redes/comunicaciones/</b></font><b><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1><font color="#FFFFFF">Servicios
externos/consulta</font></font><font face="Verdana, Arial, Helvetica, sans-serif" size="1" color="#ffffff">?
</font></b></p>
</blockquote>
</td>
</tr>
</table>
<table border="0" width="670" cellspacing="0" cellpadding="0" bgcolor="#336699" height="167">
<tr>
<td width="270" height="168">
<blockquote>
<p><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color="#ffffff">
<input type="radio" name="i4" value="AT">
</font><font color="#ffffff"> $ 10 millones o más<br>
</font></b></font><b><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input type="radio" name="i4" value="BT">
$ 5,000,000 a $9,999,999<br>
<input type="radio" name="i4" value="CT">

```



```

$ 2,500,000 a $ 4,999,999<br>
<input type="radio" name="i4" value="DT">
$ Un millón a $ 2,499,999<br>
<input type="radio" name="i4" value="ET">
$ 600,000 a $ 999,999<br>
<input type="radio" name="i4" value="FT">
$ 400,000 a $ 599,999<br>
<input type="radio" name="i4" value="GT">
$ 200,000 a $ 399,999<br>
<input type="radio" name="i4" value="HT">
$ 100,000 a $ 199,999 <br>
<input type="radio" name="i4" value="IT">
$ 50,000 a $ 99,999<br>
<input type="radio" name="i4" value="JT">
$ 25,000 a $ 49,999<br>
<input type="radio" name="i4" value="KT">
Menos de $ 25,000<br>
<input type="radio" name="i4" value="ZT">
Nada</font></b></p>
</blockquote>
</td>
<td width="400" height="168">
<blockquote>
<p><font face="Verdana, Arial, Helvetica, sans-serif" size="2" color="#ffffff"><b>CIFRAS
EN <font size="4">D&Oacute;LARES</font></b></font><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1> <b><font color="#ffffff" face="Verdana, Arial, sans-
serif" size="1">
</font></b></font></p>
</blockquote>
</td>
</tr>
</table>
<table border="0" width="670" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<p>&nbsp;</p>
<table width="450" border="0" bgcolor="#336699">
<tr>
<td><font color="#FFFFFF"><b><font size="4">Para suscriptores a cualquiera
de nuestras revistas</font></b></font></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>5.¿Por favor indique cuál de las siguientes opciones describe
mejor su función en la adquisición de servicios o productos de cómputo?</b></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p>
<select
name=a6>
<option selected>----Seleccione la opción que describe su
función----</option>

```

```

<option value="10">Determina necesidades/características</option>
<option value="11">Evalúa productos/vendedores/marcas</option>
<option value="12">Especifica productos/vendedores/marcas</option>
<option value="13">Autoriza el proceso final/aprobación de la adquisición</option>
<option value="14">Define objetivos del negocio/dirección/presupuestos</option>
<option value="19">No estoy involucrado</option>
</select>
</p>
</blockquote>
</td>
</tr>
<tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td height="2">
<blockquote>
<p><b><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1">6</font><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1">&quest;En qué funciones está usted involucrado? (marque todos los necesarios)</font></b></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#cccccc face="Verdana, Arial, Helvetica, sans-serif" size=2><b><font size=1> <font color="#FFFFFF">
<input name=f8a type=checkbox value=1,>
</font></font></b><font size="1" color="#FFFFFF">Intranets/Internet
<br>
<input name=f8b type=checkbox value=2,>
Redes/</font></font><font size="1" face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">>Telecomunicaciones</font><font face="Verdana, Arial, Helvetica, sans-serif" size="2" color="#FFFFFF"><font size=1>nicaciones<br>
<input name=f8c type=checkbox value=3,>
Desarrollo de aplicaciones <br>
<input name=f8d type=checkbox value=4,>
Soporte a PC/escritorio/remoto <br>
<input name=f8e type=checkbox value=5,>
Ingeniero de sistemas/integrador/consultoría <br>
<input name=f8f type=checkbox value=6,>
Administración de sistemas/software<br>
<input name=f8g type=checkbox value=7,>
Programador <br>
<input name=f8h type=checkbox value=8,>
Servicios de outsourcing en IT <br>
</font><font face="Verdana, Arial, Helvetica, sans-serif" size="2" color="#FFFFFF"><font size=1>
<input name=f8j type=checkbox id="f8j" value=a,>
Seguridad</font></font><font size=1> <br>
<input name=f8i type=checkbox value=9,>
</font></font><font size="1" face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFFF">No estoy involucrado en funciones IS/IT<br>
<b>OTRAS
FUNCIONES (por favor especifique)</b></font> <font color="#FFFFFF">

```

```

        <input maxlength=40
        name=f8o>
        </font></p>
    </blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
    <td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
    <tr bgcolor="#FFFFFF">
        <td></td>
    </tr>
    <tr>
        <td height="11">
            <blockquote>
                <p><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>7.¿Cu&aacute;ntos
                de estos sistemas est&aacute;n instalados en su empresa?</b></font><font color="#ffffff" face="Verdana, Arial, Helvetica,
                sans-serif" size="1"><b>
                (marque con n&uacute;mero todos los necesarios) </b></font></p>
            </blockquote>
        </td>
    </tr>
    <tr>
        <td>
            <blockquote><font color=#ffffff>
                <input type="text" name="s9a" size="7">
                <font face="Verdana, Arial, Helvetica, sans-serif" size="1">Mainframe/computadoras
                a gran escala<br>
                </font><font color=#ffffff>
                <input type="text" name="s9b" size="7">
                <font face="Verdana, Arial, Helvetica, sans-serif" size="1">Servidores
                UNIX<br>
                </font><font color=#ffffff>
                <input type="text" name="s9c" size="7">
                <font face="Verdana, Arial, Helvetica, sans-serif" size="1">Servidores
                PC<br>
                </font><font color=#ffffff>
                <input type="text" name="s9d" size="7">
                <font face="Verdana, Arial, Helvetica, sans-serif" size="1">PCs
                incluyendo estaciones de trabajo/port&atilde;tiles/handhelds </font></font></blockquote>
            </td>
        </tr>
    <tr bgcolor="#FFFFFF">
        <td></td>
    </tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
    <tr bgcolor="#FFFFFF">
        <td></td>
    </tr>
    <tr>
        <td height="11">
            <blockquote>
                <p><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>8.¿Qu&eacute;
                tipo de sistemas operativos se utilizan en su empresa?(marque todos
                los necesarios) </b></font></p>
            </blockquote>
        </td>
    </tr>
    <tr>
        <td height="188">
            <blockquote><font color=#ffffff><font face="Verdana, Arial, Helvetica, sans-serif" size="1"><br>
            <input name="o10p" type="checkbox" id="o10p" value="p,">

```

```

Windows 2003<br>
</font><font color=#ffffff><font face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name="o10a" type="checkbox" id="o10a" value="a,">
Windows 2000</font></font><font face="Verdana, Arial, Helvetica, sans-serif" size="1"> <br>
<input type="checkbox" name="o10b" value="b,">
Windows NT<br>
<input type="checkbox" name="o10c" value="c,">
Windows 95/98<br>
<input type="checkbox" name="o10d" value="d,">
Mac OS (Macintosh)<br>
<input type="checkbox" name="o10e" value="e,">
Sun OS/Solaris<br>
<input type="checkbox" name="o10f" value="f,">
UNIX<br>
<input type="checkbox" name="o10g" value="g,">
Linux<br>
<input type="checkbox" name="o10h" value="h,">
MVS/VM/MPE <br>
<input type="checkbox" name="o10i" value="i,">
OS/400<br>
<input type="checkbox" name="o10j" value="j,">
NetWare<br>
<input type="checkbox" name="o10k" value="k,">
Palm OS<br>
<input type="checkbox" name="o10l" value="l,">
Open VMS<br>
<input type="checkbox" name="o10m" value="m,">
Tru64<br>
<input type="checkbox" name="o10n" value="n,">
Windows CE<br>
<input type="checkbox" name="o10o" value="o,">
Windows XP<br>
Otros (especifique)
<input type="text" name="o10z" size="45">
<br>
</font></font></blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>9.¿Cuáles de los siguientes productos, servicios o tecnologías
planea aprobar, especificar, recomienda comprar o tiene influencia
en la compra? (marque todos los necesarios)</b></font> </p>
</blockquote>
</td>
</tr>
<tr>
<td height="65">
<blockquote>
<p><b><font face="Verdana, Arial, Helvetica, sans-serif" size="1" color="#FFFFFF">Hardware</font></b></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11a type=checkbox value=a,>
<font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1><font color="#FFFFFF">Mainframes/

```

```

</font></font> <font color="#FFFFFF"><font
  face="Verdana, Arial, Helvetica, sans-serif" size=1>computadoras
a gran escala </font></font><font
color=#ffffff><br>
<input name=c11b
type=checkbox value=b,>
</font><font color="#FFFFFF"><font
  face="Verdana, Arial, Helvetica, sans-serif" size=1>PCs/ estaciones
de trabajo</font></font><font
color=#ffffff><br>
<input
name=c11c type=checkbox value=c,>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font color="#FFFFFF"><font
  face="Verdana, Arial, Helvetica, sans-serif" size=1>PCs portátiles</font></font></font><font
color=#ffffff> </font><font
color=#ffffff><br>
<input
name=c11d type=checkbox value=d,>
Escanners<br>
<input
name=c11e type=checkbox value=e,>
Impresoras<br>
<input
name=c11f type=checkbox value=f,>
M&acute;dems <br>
<input
name=c11g type=checkbox value=g,>
Quemador de CD</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td height="65">
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Servicios</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11h type=checkbox value=h,>
<font
color=#ffffff>ASP, ISP<br>
<input
name=c11i type=checkbox value=i,>
eBusiness/eCommerce/Internet/Intranet<br>
<input name=c11j type=checkbox value=j,>
Outsourcing/Consultoría/Integración de servicios<br>
<input name=c11k type=checkbox value=k,>
Web hosting<br>
<input name=c11l type=checkbox value=l,>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff> Call center/Help desk</font></font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff></font></font><font
color=#ffffff><br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff

```

**ESTA TESIS NO SALE
DE LA BIBLIOTECA**

```

face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>eBusiness/eCommerce/Internet/Intranet</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11m type=checkbox value=m,>
<font
color=#ffffff>Customer Relationship Management (CRM)/eCRM<br>
<input name=c11n
type=checkbox value=n,>
Supply chain/Cadena de abastecimientos<br>
<input
name=c11o type=checkbox value=o,>
Sistemas de administración de contenido<br>
<input
name=c11p type=checkbox value=p,>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff>Streaming Media</font></font><font
color=#ffffff><br>
<input
name=c11q type=checkbox value=q,>
Aplicaciones web<br>
<input
name=c11r type=checkbox value=r,>
Administración de sitios web/Monitoreo de rendimiento<br>
<input
name=c11s type=checkbox value=s,>
Caching<br>
<input
name=c11t type=checkbox value=t,>
eProcurement<br>
<input
name=c11u type=checkbox value=u,>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff>Personalización/eMarketing/Automatización de fuerza
de ventas</font></font><font
color=#ffffff><br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td height="35">
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Redes/Comunicaciones</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11bd type=checkbox value=bd,>
<font color=#ffffff>LANs</font><font
color=#ffffff><br>
<input name=c11be
type=checkbox value=be,>
WANs<br>
<input
name=c11bf type=checkbox value=bf,>
Switches<br>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff>
<input
name=c11bg type=checkbox value=bg,>

```

```

Ruteadores</font></font><br>
<font
color=#ffffff>
<input
name=c11bh type=checkbox value=bh,>
Acceso remoto<br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td height="35">
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Almacenamiento</b></font></p>
<p><font color="#cccccc" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<font color="#FFFFFF">
<input name=c11v type=checkbox value=v,>
SAN (Redes de área de almacenamiento)<br>
<input name=c11w
type=checkbox value=w,>
Enterprise Storage <br>
<input name=c11x
type=checkbox value=x,>
Almacenamiento de escritorio<br>
</font></font><font face="Georgia, Times New Roman, Times, serif" size="1" color="#FFFFFF">
</font></p>
</blockquote>
</td>
</tr>
<tr>
<td height="55">
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Desarrollo de aplicaciones de software</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11y type=checkbox value=y,>
<font
color=#ffffff>XML<br>
<input name=c11z
type=checkbox value=z,>
Java<br>
<input
name=c11aa type=checkbox value=aa,>
Herramientas de desarrollo de aplicaciones<br>
<input
name=c11ab type=checkbox value=ab,>
Prueba de aplicaciones/Testing<br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Aplicaciones de software</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11ac type=checkbox value=ac,>
<font
color=#ffffff>ERP <br>
<input name=c11ad type=checkbox value=ad,>

```

```

Data Mining/Análisis estadístico/Inteligencia de negocio (BI)<br>
<input name=c11ae type=checkbox value=ae,>
Administración de bases de datos/Data Warehouse<br>
<input name=c11af type=checkbox value=af,>
Integración de aplicaciones empresariales/Web Services<br>
<input name=c11ag type=checkbox value=ag,>
Administración de documentos/Imaging<br>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff>
<input name=c11ah type=checkbox value=ah,>
</font></font><font
color=#ffffff>Finanzas/Facturación<br>
<input name=c11ai type=checkbox value=ai,>
Administración de activos<br>
<input name=c11aj type=checkbox value=aj,>
Groupware<br>
<input name=c11ak type=checkbox value=ak,>
Administración de sistemas<br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Seguridad</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11al type=checkbox value=al,>
<font
color=#ffffff>Software Anti-Virus<br>
<input name=c11am type=checkbox value=am,>
VPN/Firewalls <br>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff> </font></font><font
color=#ffffff>
<input name=c11an type=checkbox value=an,>
PKI<br>
<input name=c11ao type=checkbox value=ao,>
Tecnolog&iacute;as de Autenticaci&oacute;n y acceso<br>
<input name=c11ap type=checkbox value=ap,>
Detecci&oacute;n de intrusos/Vuln Assessment<br>
<input name=c11aq type=checkbox value=aq,>
Encripci&oacute;n<br>
<input name=c11ar type=checkbox value=ar,>
Seguridad de contenidos<br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Convergencia digital</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11as type=checkbox value=as,>
<font
color=#ffffff>Voz sobre IP<br>

```



```

<input name=c11at type=checkbox value=at,>
Video conferencia<br>
<input name=c11au type=checkbox value=au,>
PBX<br>
</font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff> </font></font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1><font
color=#ffffff>
<input name=c11av type=checkbox value=av,>
</font></font></font><font
color=#ffffff>Call centers para web <br>
</font></font></p>
</blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b>Tecnolog&iacute;a m&oacute;vil&iacute;nal&iacute;mbrica</b></font></p>
<p><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
<input name=c11aw type=checkbox value=aw,>
<font
color=#ffffff>Servicios inal&iacute;mbricos/celulares/radiolocalizadores
(pagers) <br>
<input name=c11ax type=checkbox value=ax,>
PANs (redes de &aacute;rea personal)<br>
<input name=c11ay type=checkbox value=ay,>
LANs inal&iacute;mbricas<br>
<input name=c11az type=checkbox value=az,>
Acceso fijo/punto a punto/MAN<br>
<input name=c11ba type=checkbox value=ba,>
WANs inal&iacute;mbricas<br>
<input name=c11bb type=checkbox value=bb,>
Plataformas de c&oacute;mputo m&oacute;vil (PDA, etc)<br>
<input name=c11bc type=checkbox value=bc,>
Aplicaciones de c&oacute;mputo m&oacute;vil<br>
</font></font></p>
<p><font face="Verdana, Arial, Helvetica, sans-serif" size="1"><b><font color="#FFFFFF">Otro
(especifique)
<input type="text" name="c11oo" size="50">
</font></b></font></p>
<p><font color="#FFFFFF"><b><font face="Verdana, Arial, Helvetica, sans-serif"><font size="2"><font size="1">
<input type="checkbox" name="c11zz" value="zz,">
Nada de lo anterior</font></font></font></b></font> </p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<p><br>
</p>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#357175">
<tr>
<td bgcolor="#FFFFFF"></td>
</tr>
<tr bgcolor="#336666">
<td bgcolor="#336699">
<blockquote>

```

```

<p><font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font
color=#fffff>N1. </font><font color=#fffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>¿Cuál de los siguientes
productos o servicios se están usando o se planean usar en su empresa?
(Marque los necesarios).</font></b></font></p>
</blockquote>
</td>
</tr>
</table>
<table width="600" border="0" bgcolor="#336699">
<tr>
<td>
<blockquote><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"
size="1">Topolog&iacute;as;</font></b></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name=nc1a type=checkbox value=1a,>
Ethernet/Fast Ethernet<br>
<input name=nc1b type=checkbox value=1b,>
</font><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1><font color="#FFFFFF">Gigabit
Ethernet</font></font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1"><br>
<input name=nc1c type=checkbox value=1c,>
FDDI <br>
<input name=nc1d type=checkbox value=1d,>
ATM<br>
<input name=nc1e type=checkbox value=1e,>
Frame Relay<br>
<input name=nc1f type=checkbox value=1f,>
SONET<br>
<input name=nc1g type=checkbox value=1g,>
Cable Modem<br>
<input name=nc1h type=checkbox value=1h,>
xDSL (ADSL, SDSL,HDSL) <br>
<input name=nc1i type=checkbox value=1i,>
Inalámbrico<br>
<input name=nc1j type=checkbox value=1j,>
T-1/T-3<br>
<input name=nc1k type=checkbox value=1k,>
OCx<br>
</font></p>
</blockquote>
</td>
</tr>
</table>
<table width="600" border="0" bgcolor="#336699">
<tr bgcolor="#336666">
<td bgcolor="#336699">
<blockquote><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font
color="#FFFFFF">Email/Groupware</font></b></font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name=nc2a type=checkbox value=2a,>
Lotus Notes/Domino<br>
<input name=nc2b type=checkbox value=2b,>
</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">Microsoft
Exchange /Mail</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1"><br>
<input name=nc2c type=checkbox value=2c,>
Novell GroupWise<br>
<input name=nc2d type=checkbox value=2d,>

```

```

Oracle Mail<br>
<input name=nc2e type=checkbox value=2e,>
Send Mail </font></blockquote>
</td>
</tr>
</table>
<table width="600" bgcolor="#336699" bordercolor="#FFFFFF">
<tr>
<td bgcolor="#336699">
<blockquote><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font size="1" color="#FFFFFF">Motores
de bases de datos</font></b></font></blockquote>
</td>
</tr>
<tr>
<td height="101">
<blockquote>
<p><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name=nc3a type=checkbox value=3a,>
Oracle<br>
<input name=nc3b type=checkbox value=3b,>
</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">Sybase
SQL Server</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1"><br>
<input name=nc3c type=checkbox value=3c,>
DB2<br>
<input name=nc3d type=checkbox value=3d,>
Informix <br>
<input name=nc3e type=checkbox value=3e,>
Microsoft SQL Server <br>
<input name=nc3f type=checkbox value=3f,>
Microsoft SNA Server <br>
<input name=nc3g type=checkbox value=3g,>
CA-Ingres</font></p>
</blockquote>
</td>
</tr>
<tr>
<td height="8">&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
<br>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td height="2"></td>
</tr>
<tr>
<td height="2">
<blockquote><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>T1</b></font><font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font
color=#ffffff>.</font><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>¿Cuál es el giro principal
de su empresa? (Marque todas las que apliquen)</font></b></font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font face="Verdana, Arial, Helvetica, sans-serif" size="1" color="#FFFFFF">
<input type="checkbox" name="t1a" value="LD,">
<b>Servicios de larga distancia<br>
<input type="checkbox" name="t1b" value="TL,">
Servicios de telefon&iacute;a local<br>
<input type="checkbox" name="t1c" value="CI,">
Servicios celular/inal&aacute;mbrico/PCS<br>
<input type="checkbox" name="t1ac" value="CC,">
Call Center/Contact Center<br>
<input type="checkbox" name="t1d" value="ID,">
Servicios de interconexi&oacute;n de datos<br>
<input type="checkbox" name="t1e" value="TC,">
Televisoras comerciales<br>

```

```

<input type="checkbox" name="t1f" value="TP,">
Televisi&ocute;n de paga (Cable, microondas)<br>
<input type="checkbox" name="t1g" value="ISP,">
Proveedor de servicios de internet (ISP)<br>
<input type="checkbox" name="t1h" value="HW,">
Proveedor de hosting para sitios web<br>
<input type="checkbox" name="t1i" value="SVA,">
Proveedor de servicios de valor agregado para redes<br>
<input type="checkbox" name="t1j" value="MR,">
Proveedor de servicios m&ocute;viles/radiolocalizadores/mensajer&iacute;a<br>
<input type="checkbox" name="t1k" value="SS,">
Proveedor de servicios satelitales<br>
<input type="checkbox" name="t1l" value="RP,">
Redes de datos p&uacute;blicas<br>
<input type="checkbox" name="t1m" value="SP,">
Servicios de infraestructura, energ&iacute;a el&eacute;ctrica, agua,etc.<br>
<input type="checkbox" name="t1n" value="IRS,">
Integrador de redes/sistemas<br>
<input type="checkbox" name="t1o" value="VAR,">
Revendedor de servicios/VAR<br>
<input type="checkbox" name="t1p" value="OUT,">
Outsourcing<br>
<input type="checkbox" name="t1q" value="XSP,">
Proveedores remotos de servicios (XSP)<br>
<input type="checkbox" name="t1r" value="MIT,">
Manufactura de equipo (IT)<br>
<input type="checkbox" name="t1s" value="DS,">
Desarrollador de software<br>
<input type="checkbox" name="t1t" value="IF,">
Instituci&ocute;n financiera<br>
<input type="checkbox" name="t1u" value="GOB,">
Dependencia del gobierno/normatividad<br>
<input type="checkbox" name="t1v" value="SCT,">
SCT<br>
<input type="checkbox" name="t1w" value="OG,">
Otras &aacute;reas del gobierno<br>
<input type="checkbox" name="t1x" value="CON,">
Consultor&iacute;a<br>
<input type="checkbox" name="t1y" value="IN,">
Investigaci&ocute;n y desarrollo<br>
<input type="checkbox" name="t1z" value="TRN,">
Transporte<br>
<input type="checkbox" name="t1aa" value="COM,">
Comercio<br>
<input type="checkbox" name="t1ab" value="DIS,">
Distribuci&ocute;n<br>
Otros (por favor especifique)</b></font> <b><font face="Verdana, Arial, Helvetica, sans-serif" size="1"
color="#FFFFFF">
<input type="text" name="t100" size="50">
</font></b></p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td height="5"></td>
</tr>
</table>
<br>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td>
<blockquote><font color="#ffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>T2</b></font><font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font
color=#ffff>. </font><font color=#ffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>¿Cuáles de los siguientes
productos, especifica, recomienda o compra? (marque todos los necesarios)

```

```

</font></b></font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1> <font color="#FFFFFF">
<b><br>
<input
name=t2a type=checkbox value=a,>
Equipo de transmisión <br>
<input
name=t2b type=checkbox value=b,>
Equipo de banda ancha <br>
</b></font><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1><font color="#FFFFFF"><b>
<input
name=t2c type=checkbox value=c,>
</b></font></font><font color="#FFFFFF"><b>Sistemas de redes inteligentes<br>
<input name=t2d type=checkbox value=d,>
Equipo de monitoreo y prueba <br>
<input
name=t2e type=checkbox value=e,>
Equipos y sistemas para TV por cable/servicios interactivos<br>
<input name=t2f type=checkbox value=f,>
Equipo de video </b></font></font></p>
<p><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name=t2g
type=checkbox value=g,>
Equipo interactivo<br>
<input name=t2h
type=checkbox value=h,>
Equipo inalámbrico/celular/PCS<br>
<input
name=t2i type=checkbox value=i,>
Sistemas de satélite/broadcast <br>
<input name=t2j type=checkbox value=j,>
Administración de redes/equipo para soporte a sistemas/computadoras<br>
<input name=t2k
type=checkbox value=k,>
Servidores/computadoras <br>
<input name=t2l
type=checkbox value=l,>
Software de aplicaciones para negocios<br>
<input name=t2m
type=checkbox value=m,>
Cableado y fibra óptica </b></font></p>
<p><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<input name=t2n
type=checkbox value=n,>
Sistemas OLTP<br>
<input name=t2o
type=checkbox value=o,>
UPS/Fuentes de poder <br>
<input name=t2p
type=checkbox value=p,>
Contenido/equipo de programación y servicio<br>
<input name=t2q type=checkbox value=q,>
Switches o ruteadores para backbone<br>
<input name=t2r type=checkbox value=r,>
Switches o ruteadores de acceso<br>
<input name=t2s type=checkbox value=s,>
Sistemas de redes &ocirc;pticas<br>
<input name=t2t type=checkbox value=t,>
Equipo para Call Center</font></b></p>
<p><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1"><br>
</font></b><b><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">Otros
(por favor especifique) </font> <font color="#FFFFFF">

```

```

        <input type="TEXT" name="t2oo" size="40">
    </font></b></p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
    <td></td>
</tr>
</table>
<p>&nbsp;</p>
<p><a name="otro"></a></p>
<table width="568" border="0" bgcolor="#336699">
    <tr>
        <td><font color="#FFFFFF"><b>Elija la revista de su preferencia y complete
        las preguntas adicionales; si desea solicitar más de un título, repita
        la operación donde corresponda</b></font></td>
    </tr>
</table>
<p><a href="#IW"></a> <font face="Arial, Helvetica, sans-
serif"><b></b></font>
    <a href="#BC"></a>
</p>
<p><font color=#ffffff
    face="Verdana, Arial, Helvetica, sans-serif"
    size=1><b><a name="IW"></a></b></font></p>
<table border="0" width="600" bgcolor="#6681C6">
    <tr>
        <td><font color="#FFFFFF"><b>Si desea recibir InformationWeek M&eacute;xico,
        por favor llene esta secci&oacute;n (Azul)</b></font></td>
    </tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#6681C6">
    <tr bgcolor="#FFFFFF">
        <td></td>
    </tr>
    <tr>
        <td height="11">
            <blockquote>
                <p><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>IW1.</b></font><font color="#ffffff"
                face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>¿C&uacute;al
                es la facturaci&oacute;n anual de su empresa? (Cifras en d&oacute;lares)</b></font></p>
            </blockquote>
        </td>
    </tr>
    <tr>
        <td height="124">
            <blockquote>
                <p><font color=#ffffff><font face="Verdana, Arial, Helvetica, sans-serif" size="1">
                <input type="radio" name="iw1" value="1">
                <b>M&aacute;s de $2,500 Millones USD<br>
                <input type="radio" name="iw1" value="2">
                De $500 a $2,500 Millones USD<br>
                <input type="radio" name="iw1" value="3">
                De $100 a $500 Millones USD<br>
                <input type="radio" name="iw1" value="4">
                De $20 a $100 Millones USD<br>
                <input type="radio" name="iw1" value="5">
                De $5 a $20 Millones USD<br>
                <input type="radio" name="iw1" value="6">
                De $500 Mil a $5 Millones USD<br>
                <input type="radio" name="iw1" value="7">
                Menos de $500 Mil USD<br>
                </b></font><font color=#ffffff><font face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>
                <input type="radio" name="iw1" value="8">
                No aplica</b></font></font></font></p>
            </blockquote>
        </td>
    </tr>

```

```

</tr>
<tr bgcolor="#FFFFFF">
  <td></td>
</tr>
</table>
<br>
<br>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#6681C6">
  <tr bgcolor="#FFFFFF">
    <td></td>
  </tr>
  <tr>
    <td height="11">
      <blockquote>
        <p><font color="#ffffff" face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>IW2.</b></font><font color="#ffffff"
face="Verdana, Arial, Helvetica, sans-serif" size="1"><b>¿Cuáles
de los siguientes productos o servicios aprueba, especifica, recomienda
comprar o tiene influencia en la compra? (marque todos los necesarios)</b></font>
</p>
      </blockquote>
    </td>
  </tr>
  <tr>
    <td height="46">
      <blockquote>
        <p><font color="#ffffff"></font><font
color=#cccccc face="Verdana, Arial, Helvetica, sans-serif"
size=1>
        <input name=iw2a type=checkbox value=a,>
        <font
color=#ffffff><b>Autos<br>
        <input name=iw2b type=checkbox value=b,>
        Renta de autos <br>
        <input name=iw2c type=checkbox value=c,>
        Seguros de autom&oacute;viles<br>
        <input name=iw2d type=checkbox value=d,>
        Seguros de Grupo<br>
        <input name=iw2e type=checkbox value=e,>
        Seguros m&eacute;dicos<br>
        <input name=iw2f type=checkbox value=f,>
        Seguros de vida<br>
        <input name=iw2g type=checkbox value=g,>
        Servicios de telefon&iacute;a de larga distancia<br>
        <input name=iw2h type=checkbox value=h,>
        Viajes<br>
        <input name=iw2i type=checkbox value=i,>
        Hoteles<br>
        <input name=iw2j type=checkbox value=j,>
        l&iacute;neas a&eacute;reas<br>
        <input name=iw2k type=checkbox value=k,>
        Servicios de mensajer&iacute;a<br>
        <input name=iw2l type=checkbox value=l,>
        Servicios de reproducci&oacute;n de documentos<br>
        <input name=iw2m type=checkbox value=m,>
        Asesor&iacute;as y consultor&iacute;as<br>
        <input name=iw2n type=checkbox value=n,>
        Convenciones<br>
        <input name=iw2o type=checkbox value=o,>
        Congresos<br>
        <input name=iw2p type=checkbox value=p,>
        Diplomados, maestr&iacute;as, posgrados en general<br>
        <input name=iw2q type=checkbox value=q,>
        Arrendamiento de inmuebles<br>
        <input name=iw2r type=checkbox value=r,>
        Aprobaci&oacute;n de uso de tarjetas corporativas<br>
        </b></font></font></p>
      </blockquote>
    </td>
  </tr>

```

```

<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
<br>
<br>
<table width="600" border="0" bgcolor="#6681c6">
<tr>
<td height="2">
<blockquote>
<p><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font color="#FFFFFF">Si
desea suscribirse a otro t&iacute;a;culo <a href="#otro">haga clic aqu&iacute;a; </a><br>
Si termin&oacute; <a href="#fin">haga click aqu&iacute;a; </a></font></b></font></p>
</blockquote>
</td>
</tr>
</table>
<p><br>
</p>
<p><a name="BC"></a></p>
<table border="0" width="600" bgcolor="#cc6600">
<tr bgcolor="#CC6600">
<td><font color="#FFFFFF"><b>Si desea recibir B-Secure, por favor llene
esta secci&oacute;n (Naranja)</b></font></td>
</tr>
</table>
<br>
<br>
<br>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#cc6600" height="46">
<tr bgcolor="#FFFFFF">
<td height="2"></td>
</tr>
<tr>
<td height="13">
<blockquote><font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color="#ffffff">B1</font></b></font><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>. Yo estoy interesado
en conocer las vulnerabilidades, riesgos y contingencias a las que pueda
estar expuesta mi empresa</font></b></font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p>
<input type="radio" name="sec1" value="-1">
<font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>S&iacute;a; </font></b></font>
<br>
<input type="radio" name="sec1" value="0">
<font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color="#ffffff">No</font></b></font></p>
</blockquote>
</td>
</tr>
</table>
<tr bgcolor="#FFFFFF">
<td height="5"></td>
</tr>
</table>
<br>
<br>
<table border="0" width="600" cellspacing="0" cellpadding="0" bgcolor="#cc6600" height="46">
<tr bgcolor="#FFFFFF">
<td height="2"></td>
</tr>
<tr>
<td height="13">
<blockquote><font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color="#ffffff">B2</font></b></font><font color=#ffffff

```



```

face="Verdana, Arial, Helvetica, sans-serif" size=1> . Yo en forma espec&iacute;fica:
recomiendo, apruebo, compro o influyo en la compra de hardware/software,
aplicaciones o servicios relacionados con la seguridad de mi empresa</font></b></font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote>
<p>
<input type="radio" name="sec2" value="-1">
<font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color=#ffffff
face="Verdana, Arial, Helvetica, sans-serif" size=1>S&iacute;cuta;</font></b></font>
<br>
<input type="radio" name="sec2" value="0">
<font
face="Verdana, Arial, Helvetica, sans-serif" size=1><b><font color=#ffffff>No</font></b></font></p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td height="5"></td>
</tr>
</table>
<br>
<table width="600" border="0" bgcolor="#CC6600">
<tr>
<td>
<blockquote>
<p><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font color="#FFFFFF">Si
desea suscribirse a otro t&iacute;cuta;tulo <a href="#otro">haga clic aqu&iacute;cuta;</a><br>
</font><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font color="#FFFFFF">Si
termin&oacute;cuta; <a href="#fin">haga click aqu&iacute;cuta;</a></font></b></font><font color="#FFFFFF">
</font></b></font></p>
</blockquote>
</td>
</tr>
</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699" height="197">
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
<tr>
<td>
<blockquote><font color=#fffff
face="Verdana, Arial, Helvetica, sans-serif"
size=1><b><a name="fin"></a>10.</b> ¿ Le gustaría recibir nuestro bolet&iacute;cuta;n
<b><i><font size="2">Netmedia en línea</font></i></b>? Netmedia Publishing
proporciona a sus suscriptores en forma gratuita los servicios de noticias
de la industria, abstractos de las versiones impresas de nuestras publicaciones,
anuncios de eventos importantes, directamente a su buzón de correo electrónico
(la dirección de e-mail suministrada al principio es necesaria para
darle este servicio). </font></blockquote>
</td>
</tr>
<tr>
<td>
<blockquote><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1>
<input CHECKED
name=q111 type=radio value=-1>
<font color="#FFFFFF" size="2">¡Si! Quiero recibir </font></font><font color="#FFFFFF" face="Verdana, Arial,
Helvetica, sans-serif" size="2">la
e-Newsletter <br>
<input name=q111 type=radio value=-1>
Quiero ver un ejemplo, me gustaría probar<br>
<input

```

```

name=q111 type=radio value="0">
No, por el momento no estoy interesado</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif"
size="1"><br>
</font><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
</font><font color=#cccccc
face="Verdana, Arial, Helvetica, sans-serif" size=1> </font></blockquote>
</td>
</tr>
<tr>
<td height="33">
<blockquote>
<p><font color="#FFFFFF" size="4"><b><font size="5">&iexcl;OBSEQUIE
UNA SUSCRIPCI&Oacute;N!<br>
Recomi&eacute;ndenos con sus colegas
</font></b></font></p>
<p><font color="#FFFFFF" face="Verdana, Arial, Helvetica, sans-serif" size="1">
<font size="2"><b>Envíe una forma de suscripción gratuita a:</b></font></font></p>
</blockquote>
</td>
</tr>
</table>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr>
<td width="25%">
<blockquote>
<p><font color=#cccccc><b><font
color=#ffffff>Nombre:</font></b></font></p>
</blockquote>
</td>
<td>
<input type="TEXT" name="q115" size="40">
</td>
</tr>
<tr>
<td width="25%">
<blockquote>
<p><font color="#FFFFFF"><b>E-mail:</b></font></p>
</blockquote>
</td>
<td>
<input type="TEXT" name="q116" size="40">
</td>
</tr>
</table>
<table border="0" width="500" cellspacing="0" cellpadding="0" bgcolor="#336699">
<tr>
<td>
<blockquote>
<p><font color=#ffffff face="Verdana, Arial, Helvetica, sans-serif"
size=2><b><i><font size="5">iMuchas gracias por su tiempo!</font></i></b>
</font></p>
<font
color=#ffffff face="Verdana, Arial, Helvetica, sans-serif" size=2>
<ul>
<li>Netmedia Publishing solo aceptará suscriptores que cumplan con
los criterios de calificación de la publicación
</ul>
</font>
<p align="CENTER">
<input type="SUBMIT" name="submit2" value="Enviar">
</p>
</blockquote>
</td>
</tr>
<tr bgcolor="#FFFFFF">
<td></td>
</tr>
</table>
</form>
</body>
</html>

```


Apéndice 5.- Script para generar la base de datos desde SQL Server.

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[FK_T_Sistema_T_Email]') and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE [dbo].[T_Sistema] DROP CONSTRAINT FK_T_Sistema_T_Email
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[FK_T_Telemarketing_T_Suscriptor]') and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE [dbo].[T_Telemarketing] DROP CONSTRAINT FK_T_Telemarketing_T_Suscriptor
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[CFL]') and OBJECTPROPERTY(id, N'IsView') = 1)
drop view [dbo].[CFL]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[A03 Ramo]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[A03 Ramo]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[CP]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[CP]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Captura]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Captura]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Catálogo de puestos]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Catálogo de puestos]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Compra (P11FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Compra (P11FU)]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[CompraIW (PIW2FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[CompraIW (PIW2FU)]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[CompraNC (PN1FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[CompraNC (PN1FU)]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Empleados]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Empleados]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[FL]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[FL]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Facturación (PIW1FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Facturación (PIW1FU)]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Fuente]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Fuente]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Funcion en la adquisicion (P6FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Funcion en la adquisicion (P6FU)]
GO
```

```

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Funciones (P8FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Funciones (P8FU)]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Giro]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Giro]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[IannualIT]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[IannualIT]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Responsabilidad en compras (P5FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Responsabilidad en compras (P5FU)]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Sistemas instalados (P9FU)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Sistemas instalados (P9FU)]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[Sistemas operativos (P10)]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Sistemas operativos (P10)]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Anunciante]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Anunciante]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Email]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Email]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Eventos]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Eventos]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_FL]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_FL]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Sistema]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Sistema]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Suscriptor]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Suscriptor]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[T_Telemarketing]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[T_Telemarketing]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[estatusllamada]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[estatusllamada]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[llama]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[llama]
GO

```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[llamada]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[llamada]
GO
```

```
if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[status]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[status]
GO
```

```
CREATE TABLE [dbo].[A03 Ramo] (
    [ramo] [smallint] NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CP] (
    [EF] [nvarchar] (30) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [del] [nvarchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [al] [nvarchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Captura] (
    [Captura] [smallint] NULL ,
    [Nombre] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Catálogo de puestos] (
    [Puesto] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [ClavePuesto] [int] NULL ,
    [Area] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [ClaveArea] [int] NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Compra (P11FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (70) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CompraIW (PIW2FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (70) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[CompraNC (PN1FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripción] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Empleados] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Cantidad] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[FL] (
    [ID] [int] NOT NULL ,
    [FLDir1] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FLDir2] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FLDir3] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FLCP] [nvarchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [SoHo] [bit] NOT NULL ,
    [FUFecha] [datetime] NULL ,
    [FUPuesto] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUGiro] [int] NULL ,

```

```

[FUEmpleados] [int] NULL ,
[FUIversion] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUFUN] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUSHa] [int] NULL ,
[FUSHb] [int] NULL ,
[FUSHc] [int] NULL ,
[FUSHd] [int] NULL ,
[FUSO] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUC] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUFACT] [int] NULL ,
[FUCIW] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUNCI] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUGTC] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[FUEQTC] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[Bsec1] [bit] NOT NULL ,
[Bsec2] [bit] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Facturación (PIW1FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Fuente] (
    [Fuente] [nvarchar] (4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Nombre] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Funcion en la adquisicion (P6FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Funciones (P8FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Giro] (
    [ramo] [smallint] NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[InuaIT] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Cantidad] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Responsabilidad en compras (P5FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Sistemas instalados (P9FU)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (70) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Sistemas operativos (P10)] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,

```

```

[Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[T_Anunciante] (
    [ID] [int] NOT NULL ,
    [uan] [datetime] NULL ,
    [anprin] [bit] NOT NULL ,
    [agente] [int] NULL ,
    [ultimallam] [datetime] NULL ,
    [IYW] [bit] NOT NULL ,
    [LBS] [bit] NOT NULL ,
    [LPA] [bit] NOT NULL ,
    [LEX] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[T_Email] (
    [ID_Email] [int] NOT NULL ,
    [Email] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Email2] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [mailBS] [bit] NOT NULL ,
    [mailIW] [bit] NOT NULL ,
    [mailTC] [bit] NOT NULL ,
    [noeb] [bit] NOT NULL ,
    [noen] [bit] NOT NULL ,
    [nopr] [bit] NOT NULL ,
    [Bol Txt] [bit] NOT NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[T_Eventos] (
    [ID] [int] NOT NULL ,
    [alas50] [int] NULL ,
    [aCIO] [int] NULL ,
    [aBIF] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [aSEG] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [aCA] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

```

```

CREATE TABLE [dbo].[T_FL] (
    [ID_FL] [int] NOT NULL ,
    [Dir1_FL] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Dir2_FL] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Dir3_FL] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [CP_FL] [nvarchar] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [SoHo] [bit] NOT NULL ,
    [FUFecha] [datetime] NULL ,
    [FUPuesto] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUGiro] [int] NULL ,
    [FUEmpleados] [int] NULL ,
    [FUInversion] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUFUN] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUSHa] [int] NULL ,
    [FUSHb] [int] NULL ,
    [FUSHc] [int] NULL ,
    [FUSHd] [int] NULL ,
    [FUSO] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUC] [nvarchar] (200) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUFACT] [int] NULL ,
    [FUCIW] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUNCI] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUGTC] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [FUEQTC] [nvarchar] (150) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Bsec1] [bit] NOT NULL ,
    [Bsec2] [bit] NOT NULL
) ON [PRIMARY]
GO

```



```

CREATE TABLE [dbo].[T_Sistema] (
    [ID_Sistema] [int] NOT NULL ,
    [status] [nvarchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [desde] [datetime] NULL ,
    [captura] [smallint] NULL ,
    [fuente] [nvarchar] (4) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Memo] [ntext] COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [verif] [datetime] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[T_Suscriptor] (
    [ID_Suscriptor] [int] NOT NULL ,
    [Titulo] [nvarchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Nombre] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Empresa] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Puesto] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Dir1] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Dir2] [nvarchar] (100) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Dir3] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [CP] [nvarchar] (6) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Tel] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Fax] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Web] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [IW] [bit] NOT NULL ,
    [BS] [bit] NOT NULL ,
    [Giro] [int] NULL ,
    [Giro2] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [fnac] [datetime] NULL ,
    [sexo] [nvarchar] (1) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [eimp] [int] NULL ,
    [acuse] [int] NULL ,
    [ultimo] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[T_Telemarketing] (
    [ID_Telemarketing] [int] NOT NULL ,
    [flamada] [datetime] NOT NULL ,
    [status_telemarketing] [nvarchar] (4) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [memo_telemarketing] [ntext] COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[estatusllamada] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[llama] (
    [Captura] [smallint] NULL ,
    [Nombre] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[llamada] (
    [Clave] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[status] (
    [status] [nvarchar] (2) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [descripcion] [nvarchar] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL
) ON [PRIMARY]
GO

```

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
```

```
CREATE VIEW dbo.CFL
```

```
AS
```

```
SELECT  dbo.T_Sistema.*, dbo.T_Suscriptor.*, dbo.T_FL.*, dbo.T_Email.*
```

```
FROM    dbo.T_Email INNER JOIN
```

```
        dbo.T_Sistema ON dbo.T_Email.ID_Email = dbo.T_Sistema.ID_Sistema INNER JOIN
```

```
        dbo.T_Suscriptor ON dbo.T_Sistema.ID_Sistema = dbo.T_Suscriptor.ID_Suscriptor INNER JOIN
```

```
        dbo.T_FL ON dbo.T_Suscriptor.ID_Suscriptor = dbo.T_FL.ID_FL
```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

Bibliografía

- [1] MOLINER MARÍA: *Diccionario de uso del español*. Editorial Gredos. Madrid, España (1998).
- [2] Webopedia: Online Computer Dictionary for Computer and Internet Terms and Definitions
<http://www.webopedia.com>
- [3] FORD, KIM, SPANIER, STEVENSON: *Tecnologías de Interconectividad de redes*. Editorial Prentice-Hall. México (1998).
- [4] INIESTA BONILLA, SÁEZ GONZÁLEZ, SÁNCHEZ PÉREZ: *Bases de datos de Marketing. Clasificación, uso y aplicaciones en la estrategia competitiva actual*.
Universidad de Almería. España (2000).
http://www.esic.es/documentos/editorial_em.pdf
- [5] Docs.sun.com: Sun Global Glossary
Sun Product Documentation
<http://docs.sun.com/app/docs/doc/805-4368/6j450e60c?a=view>
- [6] NameSecure : Glossary for terms related to domain name registration at namesecure.com
WEB SERVICES GLOSSARY
http://www.namesecure.com/en_US/index.jhtml?cat=null&subCat=glossary
- [7] Niveles de RAID
<http://www.smdata.com/NivelesRAID.htm>
- [8] Base de datos - Wikipedia en español
http://es.wikipedia.org/wiki/Base_de_datos
- [9] Object-oriented database management system - a Whatis.com definition - see also: ODBMS, OODBMS
http://searchdatabase.techtarget.com/sDefinition/0,,sid13_qci213671,00.html
- [10] Transaction Log Backups (Administering SQL Server (SQL Server))
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/admsql/ad_bkprst_565v.asp
- [11] SQL Server: Optimizing Database Performance Through Indexes
Kevin Kline, Baya Pavlashvili.
<http://www.awprofessional.com/articles/article.asp?p=28285>
- [12] SILBERSCHATZ, GALVIN, GAGNE: *Operating system concepts*.
Editorial: Wiley, USA
- [13] Glossary of Internet and Computer Terms
<http://www.5starsupport.com/info/glossary.html>
- [14] SILBERSCHATZ, KORTH, SUDARSHAN: *Database System Concepts*
editorial: MC Graw Hill USA.

[15] Ethernet
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ethernet.htm

[16] Protocolos TCP/IP
<http://usuarios.lycos.es/janjo/janjo1.html>

[17] Deitel y Deitel: *Cómo programar en JAVA*.
editorial: Prentice Hall. 1998 México, D.F.

[18] About the Java Technology
<http://Java.sun.com/docs/books/tutorial/getStarted/intro/definition.html>

[19] The Java Language Environment: Contents
<http://Java.sun.com/docs/white/langenv>

[20] Introducción a Javascript
<http://www.desarrolloweb.com/articulos/490.php?manual=20>

[21] Introducción a la programación en PHP
<http://www.desarrolloweb.com/articulos/303.php?manual=12>

[22] Microsoft training & certification
Maintaining a Microsoft Windows Server 2003 Environment Workbook
Course: 2275

[23] Diseño e implantación de sistemas de información y procesamiento de datos para empresas
Héctor Alejandro Sánchez Rodríguez
<http://www.monografias.com/trabajos14/implantacion-datos/implantacion-datos.shtml>

[24] Object Database vs. Object-Relational Databases
Steve McClure
<http://www.ca.com/products/jasmine/analyst/idc/14821E.htm>