UN ANALISIS REGIONAL DE LA ARQUITECTURA CLIENTE/SERVIDOR

TESIS

MAESTRIA EN ADMINISTRACION DE SISTEMAS DE INFORMACION

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

POR

ALMA GLORIA FLORES TREJO

AGOSTO DE 1995

UN ANALISIS REGIONAL DE LA ARQUITECTURA CLIENTE/SERVIDOR

TESIS

MAESTRIA EN ADMINISTRACION DE SISTEMAS DE INFORMACION

INSTITUTO TECNOLOGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

POR

ALMA GLORIA FLORES TREJO

AGOSTO DE 1995

UN ANÁLISIS REGIONAL DE LA ARQUITECTURA CLIENTE/SERVIDOR

TESIS

MAESTRÍA EN ADMINISTRACIÓN DE SISTEMAS DE INFORMACIÓN

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

POR

ALMA GLORIA FLORES TREJO

AGOSTO DE 1995

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

DIVISIÓN DE GRADUADOS E INVESTIGACIÓN
PROGRAMA DE GRADUADOS EN INFORMÁTICA

Los miembros del comité de tesis recomendamos que la presente tesis de la Lic. Alma Gloria Flores Trejo sea aceptada como requisito parcial para obtener el grado académico de Maestra en Administración de Sistemas de Información.

Comité de tesis:

Dr. David Angel Alanís Dávila
ASESOR PRINCIPAL

Ing. Francisco Alberto Geada Gutiérrez. MA, MC
SINODAL

Ing. Benjamín Ernesto Hernández Sepúlveda. MC
SINODAL

Dr. Carlos Scheel Mayenberger
Director del Programa de Graduados en Informática

AGOSTO DE 1995

UN ANÁLISIS REGIONAL DE LA ARQUITECTURA CLIENTE/SERVIDOR

POR

ALMA GLORIA FLORES TREJO

TESIS

Presentada a la División de Graduados e Investigación
Este Trabajo es Requisito Parcial
para obtener el Título de
Maestra en Administración de Sistemas de Información

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

AGOSTO DE 1995

DEDICATORIA

A DIOS, por creer en él....por darme la oportunidad de vivir y existir.

A San Juan Lagos, por creer que hasta lo imposible es posible.

A mi mamá, quien me dió la vida y luchó por mí, ...quien me dió todo lo que necesitaba: su amor, alegría, tiempo, paciencia, dedicación, ... por enseñarme a luchar por conseguir lo que uno anhela....por inculcarme ciertos valores.... por ella soy lo que soy....por tantas y tantas cosas....mil gracias.

A mi abuelita Luisa (q.e.p.d.), por su amor y cariño hacia mi madre y a mí....quien hasta en sus últimos momentos de vida demostró su alegría, optimismo y fortaleza.

Al Sr. Antonio Gorordo Domínguez y Sra. Josefina Arias de Gorordo, por apoyar y ayudar siempre a mi madre. Por brindarme su cariño, por confiar en mí para lograr ser lo que hasta ahora soy.....por su ayuda económica y desinteresada logré estudiar y ser una persona de bien para la sociedad.....por ser unos padres para mí.....en fin, no tengo palabras para agradecerles *todo* lo que han hecho por nosotras.....muchas gracias.

A Pepa, Toño, Jorge, Beto y Teté, por quererme y, apoyarme como si fuera una hermana más para ellos.

A Alberto Gorordo Arias, por haberme apoyado para conseguir la beca del tec y así empezar esta gran carrera profesional.

A José Luis, por darme su apoyo incondicional y cariño, por su sencillez, por valorarme, por saber que puedo contar con él en cualquier momento.....simplemente ....por estar en mi camino en el momento preciso de mi vida.

AGRADECIMIENTOS

Al ITESM Campus Tampico, por darme la oportunidad de trabajar en mi área y poder cursar esta maestría.

Al Ing. Javier Ponce, quien me brindó la oportunidad de trabajar en Servicios Escolares del Campus Tampico para desarrollarme profesionalmente.

Al Ing. Francisco Geada, quién confió en mí y me dió la oportunidad de laborar en el depto. de informática del Campus Tampico. Por cumplir una gran función para lograr mi maestría.....ser mi sinodal. Por contar con su amistad y ser un gran ejemplo en mi vida profesional.....gracias por ayudarme y apoyarme siempre.

Al Ing. Benjamín Hernández, por conocerlo y haber trabajado juntos, ...por haberme apoyado en mi crecimiento profesional.....por darme su tiempo para lograr la terminación de mi tesis, siendo mi sinodal.

Al Dr. David Alanís Dávila, por haber aceptado ser mi asesor de tesis, ya que sin su ayuda no hubiera logrado esto.....gracias por su gran apoyo.

A todos los que fueron mis maestros y compañeros de trabajo del ITESM Campus Tampico.....quienes me vieron crecer profesionalmente desde que ingresé al Campus a preparatoria.....ya que siempre los recuerdo con cariño.

A mis tíos, Pablo, Jael, Delta, Virginia, Nora, Guadalupe.....quienes siempre han estado con mi madre y conmigo, dándome todo su cariño y apoyo, para seguir adelante siempre.

A todos mis primos y sobrinos.....en especial a Jorge Alberto Flores Sánchez (q.e.p.d.) por su sencillez, por su gran amistad y cariño por nuestra familia.....por su valentía.

A mis padrinos, Francisco y Benita.....por su enorme cariño y singular simpatía.

Al Ing. Ponciano Betancourt, por ser un gran amigo y compañero de generación....por sus consejos....por darme su apoyo en momentos difíciles....por ayudarme en el desarrollo de mi tesis.

Al Ing. Javier Narro y al Ing. Arístides Palma, por ser grandes maestros, amigos y compañeros de maestría....por sus consejos.

Al P. Quintín Balderrama López, S.J., por darme la oportunidad de trabajar y convivir en el Instituto Cultural Tampico demostrando mi desempeño profesional.....por ser una gran persona.

A toda la comunidad jesuita, así como a profesores y personal del I.C.T. quienes me brindaron su calidez y apoyo desde mi ingreso al colegio.

A la Comunidad Magis, por encontrar grandes amistades e inculcarme valores que desconocía.....en especial a Pancho, Peque, Ofelia, Alma, Consuelo, René, Conchita, P. José Luis Ornelas.

A mis grandes amigas....Adriana Hernández, Alma Gómez, Alhelí Echazarreta, Estela Meza, con quienes he pasado momentos inolvidables de mi vida, por enseñarme y demostrarme lo que es una gran amistad.....por contar con ellas siempre.

A una gran amiga.....a distancia.....Lupita Díaz De León....por sus consejos y apoyo para terminar mi tesis...por su valiosa ayuda y colaboración durante mis cursos de maestría.

A Margarita, Carlos, Bertha, Honorio, Mauricio, Pepe.....por conocerlos y brindarme su amistad durante mis estancias en Monterrey.

Al Ing. Antonio Santos.....por darme la oportunidad de conocerlo y ser una gran persona a quien admiro por tener una gran fortaleza y espíritu de crecimiento profesional.....por creer en mi labor profesional.....por ser una persona muy humana.

A Betty, Mario, Ernesto por colaborar y trabajar conmigo....por su amistad...por demostrar su gran profesionalismo.

A todos los que fueron mis alumnos, ya que me inculcaron una gran virtud....la capacidad de enseñar.....por brindarme grandes amistades como: Gracia, Fernando, Marissa, Betty.

A todas las personas que laboran en Sinapsis y en especial a la Dra. Morín por todas sus atenciones y apoyo en el transcurso de la maestría y de mi defensa.

A quienes asistieron y me apoyaron durante mi defensa de tesis.....en Tampico Tam.: mi mamá, mi asesor: Dr. David Alanís; mis sinodales: Ing. Francisco Geada, Ing. Benjamín Hernández; mis tutores: José Antonio Gorordo Dmz., Josefina Arias de Gorordo; C.P. José Luis Manrique, Ing. Patricia Martínez, Ing. Patricia Osti, Lic. Imelda Pontvianne, Lic. Alhelí Echazarreta, Químico Francisco Martínez, Lic. Alma Gómez, Lic. Ma. del Consuelo Arámbula, Ing. Orlando Figón, tía: Sra. Delta Flores, Emma González, Ing. Antonio Santos, Lic. Gracia Pineda, Lic. Marysol Saldaña, Ing. Ana Laura Hinojosa. En Monterrey N.L.: Mayra Quintanilla, Rosenda, Ing. Guadalupe Díaz De León, Dra. Ma. Elena Morín.

A las empresas que colaboraron y ayudaron en el desarrollo de investigación de esta tesis.

A todas mis amistades que no menciono pero que siempre las tengo presentes.

TABLA DE CONTENIDO

Página

Capítulo 1. Introducción 1

Capítulo 2. Arquitectura Cliente/Servidor 4

2.2 Características de Cliente/Servidor 4
2.3 Beneficios de Cliente/Servidor 8
2.4 Downsizing 12
2.5 Rightsizing 14
2.6 Interfaces Gráficas a Usuarios 16
2.6.1 Definición 16
2.6.2 Características 18
2.6.3 Necesidad de Estándares 19
2.6.4 Ambientes GUI 20
2.6.4.1 Windows 3.x 20
2.6.4.2 Presentation Manager 23
2.6.4.3 Motif 24
2.6.4.4 Open Look 25

Capítulo 3. Desarrollo de Aplicaciones Cliente/Servidor 27

3.1 Factores que afectan al Software 27
3.2 Hardware/Network 29
3.3 Soporte y Servicio 33
3.4 Metodologías para implantación de sistemas 34
3.4.1 Tradicional 34
3.4.2 Estructurado 37
3.4.3 LEAD 44

Capítulo 4. Situación Actual en México 51

4.1 La informática en México 51
4.2 Cultura informática, base de la competitividad 52

Capítulo 5 Análisis de la Situación Actual en Tampico con Cliente/Servidor 59

5.1 Sondeo de la situación actual 59
5.2 Metodología y Análisis 61

Capítulo 6. Estrategia de Implantación 79

6.1 Elementos a considerar 79

Conclusiones 85
Proyectos Futuros 86
Anexo 1. Plataformas bajo Cliente/Servidor 87
Anexo 2. Encuesta 102
Referencias 110
Vita 113

LISTA DE FIGURAS

Página

Figura 1. Sistemas Cliente/Servidor 2

Figura 2. Proceso Maestro/Esclavo del Servidor 5

Figura 3. Comunicación RPC 7

Figura 4. Comunicación de intercambio entre procesos utilizando "nonblocking" 7

Figura 5. Diseño de Ventana: Interface Gráfica a Usuarios 17

Figura 6. Diseño Ventana: "GUI" de captura 18

Figura 7. Diferencias entre DDE y OLE 22

Figura 8. El Ciclo de Vida del Método Tradicional 35

Figura 9. El Ciclo de Vida del Método Estructurado 38

Figura 10. Un modelo de cuál es la función del sistema 40

Figura 11. El "área gris" entre el sistema y el ambiente 40

Figura 12. Diagrama parcial de Contexto 41

Figura 13. Lista de acontecimientos 41

Figura 14. Componentes de un diagrama de estructura 42

LISTA DE GRÁFICAS

Página

Gráfica 1. Nuevas Tecnologías en Telecomunicaciones 54

Gráfica 2. Mejoras que se buscan con las Nuevas Tecnologías 54

Gráfica 3. Nuevas Tecnologías en Informática 54

Gráfica 4. Sistemas Operativos que se están instalando 55

Gráfica 5. Implantación de Tecnologías Front-End para atender al Cliente 55

Gráfica 6. Número de Sistemas Operativos utilizados 55

Gráfica 7. Interoperabilidad bajo Sistemas Abiertos 56

Gráfica 8. Sistemas Operativos en funcionamiento 56

Gráfica 9. Arquitectura dominante en el Sector Financiero 56

Gráfica 10. Mainframe tiene Futuro 57

Gráfica 11. Outsourcing es una Alternativa 57

Gráfica 12. Outsourcing en diversas áreas 57

Gráfica 13. Elementos Clave en la Decisión de Compra 58

Gráfica 14. Puesto que ocupa actualmente 61

Gráfica 15. Número de Empleados que tiene la empresa actualmente 62

Gráfica 16. "Software" de desarrollo que han utilizado para las aplicaciones 62

Gráfica 17. "Software" de desarrollo bajo la tecnología Cliente/Servidor utilizado para las aplicaciones 63

Gráfica 18. Tiempo de uso del "software" bajo la tecnología Cliente/Servidor 63

Gráfica 19. Indique las herramientas con las que empezó a desarrollar "software" para aprender la tecnología Cliente/Servidor 64

Gráfica 20. Porcentajes de la Toma de Decisiones para la adquisición de la Herramienta Cliente/Servidor 65

Gráfica 21. Proveedores para el desarrollo de las aplicaciones 65

Gráfica 22. Cantidad de computadoras "Cliente" en base a su RAM 67

Gráfica 23. Cantidad de computadoras "cliente" en base a su procesador 67

Gráfica 24. Plataformas donde corren los Manejadores de Bases de Datos 69

Gráfica 25. Servidor de Base de Datos utilizado 69

Gráfica 26. Satisfacción por el desempeño del producto 70

Gráfica 27. Facilidad de uso de la herramienta 70

Gráfica 28. Promedio de habilidades que los usuarios finales tienen 71

Gráfica 29. Frecuencia con que utilizan los usuarios finales los graficadores, consultas, reporteadores 72

Gráfica 30. Productos que se evaluaron para la selección de la herramienta enfocada al Usuario Final 73

Gráfica 31. Características que hicieron posible la selección de la Herramienta orientada a Usuarios Finales 73

Gráfica 32. Opiniones sobre la dificultad de llevar a cabo la capacitación del personal, para aprender el uso de la Herramienta de desarrollo bajo la Tecnología Cliente/Servidor 74

Gráfica 33. Opiniones sobre la dificultad para llevar a cabo la capacitación a los usuarios finales, para aprender el uso de las nuevas aplicaciones 75

LISTA DE TABLAS

Página

Tabla 1. Secciones de la encuesta realizada a desarrolladores de aplicaciones bajo Cliente/Servidor 60

Tabla 2. Criterios de evaluación para la selección de los proveedores 66

Tabla 3. Relación de estaciones "Cliente" para desarrollo y uso de aplicaciones Cliente/Servidor 66

Tabla 4. Tipos de servidores usándose actualmente para herramientas Cliente/Servidor 68

Tabla 5. Aspectos a mejorar de la herramienta Cliente/Servidor 71

Tabla 6. Herramientas complementarias 72

Tabla 7. Ventajas y desventajas en Cliente/Servidor 76

Tabla 8. Pasos para la estrategia de implantación 86

RESUMEN

En esta tesis se presentan los resultados obtenidos mediante un sondeo de diez empresas en la ciudad de Tampico Tamaulipas, sobre la arquitectura Cliente/Servidor.

Los alcances de esta tesis se delimitan a los siguientes dos objetivos:

1) Conocer en Tampico el uso de herramientas Cliente/Servidor para el desarrollo de aplicaciones, tomando como base el apoyo de bibliografía actualizada, para llevar acabo el sondeo en donde se aplicaron encuestas a las empresas que desarrollan "software" bajo Cliente/Servidor.

2) Conocer el tipo de metodología o estrategia que siguen estas empresas para el desarrollo de sus aplicaciones, dando lugar a proponer una estrategia de implantación adecuada al medio, y a las necesidades que las empresas mostraron en el resultado de la encuesta.

CAPÍTULO 1
INTRODUCCIÓN

Se dice que la arquitectura Cliente/Servidor puede ser un tanto confusa, puesto que el concepto es relativamente nuevo, ya que promete la revolución del procesamiento de datos en las organizaciones. La confusión puede darse por la gran revolución tecnológica actual y la cantidad de productos que la acompañan, dando lugar a un sin número de preguntas, tales como: ¿Cuáles productos son cliente/servidor y cuáles no? ¿Los recursos de redes compartidas como NOVELL y LAN MANAGER son verdaderamente cliente/servidor? ¿Deberán todos los clientes tener interfaces gráficas? ¿Puede una aplicación ser cliente/servidor sin haber sido construida con productos cliente/servidor?.

Se dice también, que la década de los 90's es la "era de la información", esto debido a que nos encontramos en medio de un proceso de evolución que se prolongará todavía por algunos años y que cambiará la forma en que obtenemos y utilizamos la información almacenada electrónicamente.

El principal motivo detrás de esta evolución es la necesidad que tienen las organizaciones de realizar sus operaciones más eficientemente, debido a la creciente presión competitiva a la que están sometidas, lo cual se traduce en la necesidad de que su personal sea más productivo, que se reduzcan los costos y gastos de operación, al mismo tiempo que se generan productos y servicios más rápidamente y con mayor calidad.

Bajo este contexto, es necesario establecer una infraestructura de procesamiento de información que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y proporcionar mejor servicio a clientes.

La arquitectura Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones, dando como consecuencia el desempeño de un papel importante en este proceso de evolución.

Debido a lo anterior, se definirán los componentes de esta arquitectura: cliente, servidor y red. Cliente es una estación que brinda servicios a los

usuarios y hace peticiones al servidor. El servidor, es una estación que procesa la información requerida por el cliente dando como resultado una respuesta a dicha petición. Una red es el medio de comunicación entre el cliente y el servidor.

La figura 1 describe en forma sencilla y completa un sistema Cliente/Servidor: [RENAUD, 1993]

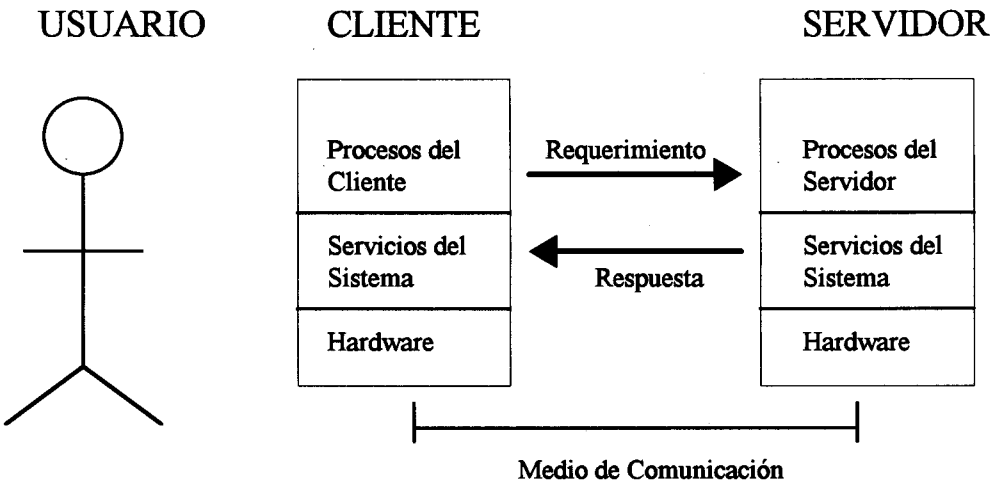


Figura 1. Sistemas Cliente/Servidor

Lo anterior no implica que otros clientes o servidores no puedan correrse bajo este sistema, ya que no existen restricciones para las comunicaciones múltiples.

Como podemos apreciar la Arquitectura Cliente/Servidor, ha sido el factor de cambio en la mayoría de las organizaciones; esto debido a las ventajas que proporciona aunado con las necesidades que reflejan las empresas.

En el capítulo 2, "Visión Cliente/Servidor", presenta en forma general los aspectos más relevantes que tienen relación con Cliente/Servidor, los cuales son: características, beneficios, "Downsizing", "Rightsizing", y las interfaces gráficas a usuarios.

En el capítulo 3, llamado "Desarrollo de Aplicaciones Cliente/Servidor", se dan a conocer los puntos claves a considerar en el desarrollo de aplicaciones, tales como: "software", "hardware", redes, soporte, servicio, metodologías para implantar sistemas.

En el capítulo 4, denominado "Situación Actual en México", se indica cómo se encuentra actualmente el país en cuanto a cultura informática se refiere, esto se muestra por medio de un estudio en el sector financiero-bancario.

En el capítulo 5, "Análisis de la Situación Actual en Tampico con Cliente/Servidor", se explica cómo se encuentra actualmente Tampico en cuanto a los recursos que utilizan bajo Cliente/Servidor. Esto, mediante la aplicación de una encuesta a diez empresas de la localidad, las cuales desarrollan sus propias aplicaciones o son proveedores de otras empresas.

En el capítulo 6, "Estrategia de Implantación", se da a conocer una serie de pasos o elementos a considerar para el desarrollo de aplicaciones Cliente/Servidor, esto en base al capítulo 5 y a las metodologías explicadas en el capítulo 3.

En el capítulo 7, "Conclusiones", se indican las conclusiones de esta investigación y las proyecciones a futuro de esta tecnología.

CAPÍTULO 2
ARQUITECTURA CLIENTE/SERVIDOR

2.2 Características de la Arquitectura Cliente/Servidor.

A continuación se explicarán las principales características de Cliente/Servidor: [RENAUD, 1993]

- Atributos del Cliente
- Atributos del Servidor
- Atributos de Comunicación

Atributos del Cliente: el proceso del cliente es proactivo, emite requerimientos al servidor. Un cliente puede interactuar con un servidor o con múltiples servidores para realizar su trabajo, ya que existen procesos que deben checarse en otros servidores, dependiendo de la aplicación que sea.

A nivel aplicación, el cliente es responsable de mantener y procesar el diálogo con el usuario. Esto incluye la ejecución de lo siguiente:

* manejo de pantalla
* menús o interpretación de comandos,
* entradas de datos y validación,
* proceso de ayuda y,
* recuperación de errores

En aplicaciones gráficas, debe ejecutar:

* manejo de ventanas,
* entrada de datos con "mouse" y teclado,
* control de cajas de diálogo y,
* sonido y video (en aplicaciones multimedias)

Es importante saber que, para administrar todas las interacciones del usuario, el cliente encubre en forma efectiva al sevidor y la red, dando la ilusión que la aplicación se está ejecutando localmente en su estación, sin el uso de otros procesos, máquinas o redes.

Atributos del Servidor: el proceso del servidor es reactivo, provocado por la llegada de requerimientos de sus clientes. Generalmente, un servidor está siempre corriendo, dando servicios a muchos clientes. Estos servicios pueden ser dados directamente por el servidor o indirectamente por procesos esclavos. Por ejemplo, el servidor de base de datos Oracle siempre está corriendo aceptando todos los requerimientos de los clientes, cuando un requerimiento llega, el servidor Oracle produce un proceso esclavo dedicado a manejar el requerimiento, de esta forma el servidor maestro recibe otros requerimientos inmediatamente. La figura 2 muestra el uso de los procesos maestro/esclavo de un servidor. [RENAUD, 1993]

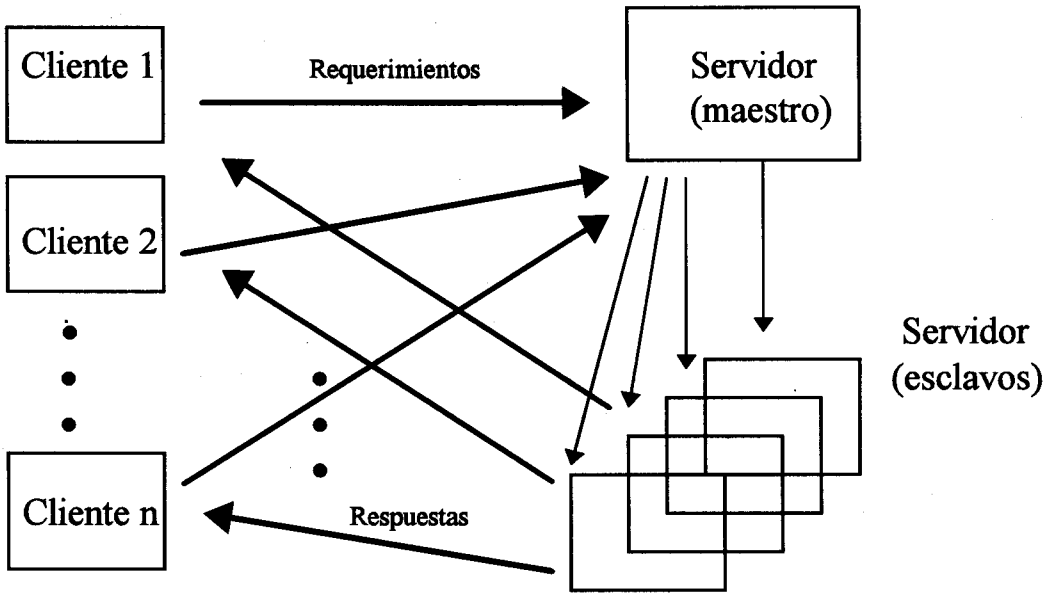


Figura 2. Proceso Maestro/Esclavo del Servidor

El procesamiento típico de un servidor incluye todos los procesos asociados con accesos; almacenamiento y organización de datos compartidos; actualización previa a datos almacenados; y cualquier administración de otros recursos compartidos. Los recursos compartidos pueden ser datos, poder del UCP (Unidad Central de Procesamiento), almacenamiento en disco o cinta, capacidad de impresión, comunicaciones, nivel de despliegue y administración de la memoria.

Atributos de Comunicación: usualmente hay muy pocas relaciones entre clientes y servidores. Es la responsabilidad del cliente localizar e iniciar el intercambio con el servidor. Si sólo hay un servidor, es muy fácil hacerlo.

En grandes redes, sin embargo, el localizar el servidor apropiado no es muy sencillo.
El estilo de comunicación entre el cliente y el servidor es transaccional y cooperativo. El aspecto transaccional significa que el servidor regresa solamente los resultados relevantes a los requerimientos del cliente. Esto es, la cantidad de datos enviada es la menos necesaria para que el cliente realice su trabajo. Esto significa que hay poco "overhead" sobre la red generada por la aplicación. Algunos aspectos importantes a considerar son: el tráfico en la red y el tiempo de respuesta del servidor. Un ejemplo importante es el que mencionó el Ing. Benjamín Hernández, Director de Informática del ITESM Campus Tampico,...."el caso de INTERNET es un tema importante de comunicación a considerar, ya que existen algunos criterios de cobro para proveer información, los cuales son: 1) el tiempo de conexión, cuyo cobro es bajo y 2) el cobro por el volúmen de información obtenida, en donde el tráfico es muy alto"...

El aspecto cooperativo, significa que el proceso ocurre entre los dos: el cliente y el servidor. Existen 2 formas de darse la comunicación. Una por medio de Llamadas Procedurales Remotas ("Remote Procedure Calls", RPC) y la otra por medio de intercambio entre procesos utilizando el "nonblocking".
Las RPC's, trabajan de la siguiente forma: el cliente hace una llamada al servidor pidiendo requerimientos y espera mientras se procesa el requerimiento, el servidor atiende la llamada y empieza a procesar la petición, el servidor regresa lo pedido al cliente y éste continua su procesamiento.

La figura 3 muestra este tipo de comunicación: [RENAUD, 1993]

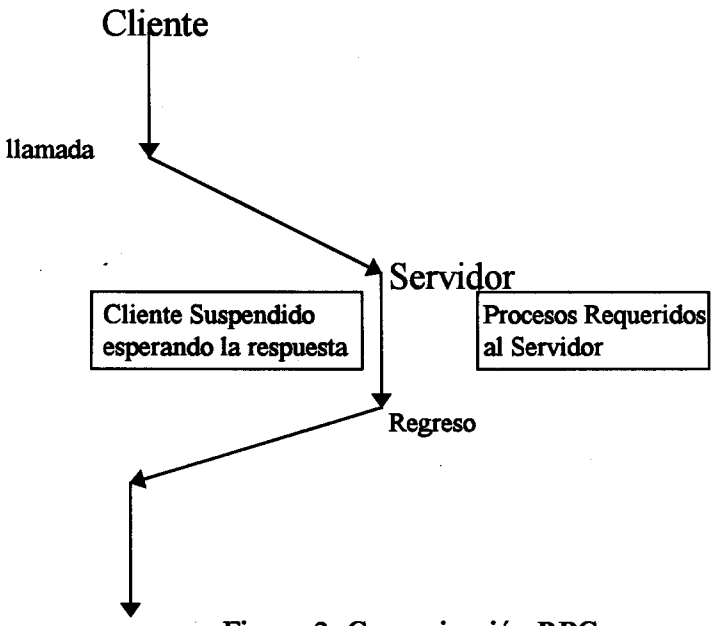


Figura 3. Comunicación RPC

En el caso del intercambio entre procesos, sólo ocurre que mientras el servidor atiende la petición del cliente éste realiza alguna otra actividad. Esto depende mucho del sistema operativo que se utilice en el cliente, ya sea multitareas o multiprocesos.
La figura 4 muestra este tipo de comunicación: [RENAUD, 1993]

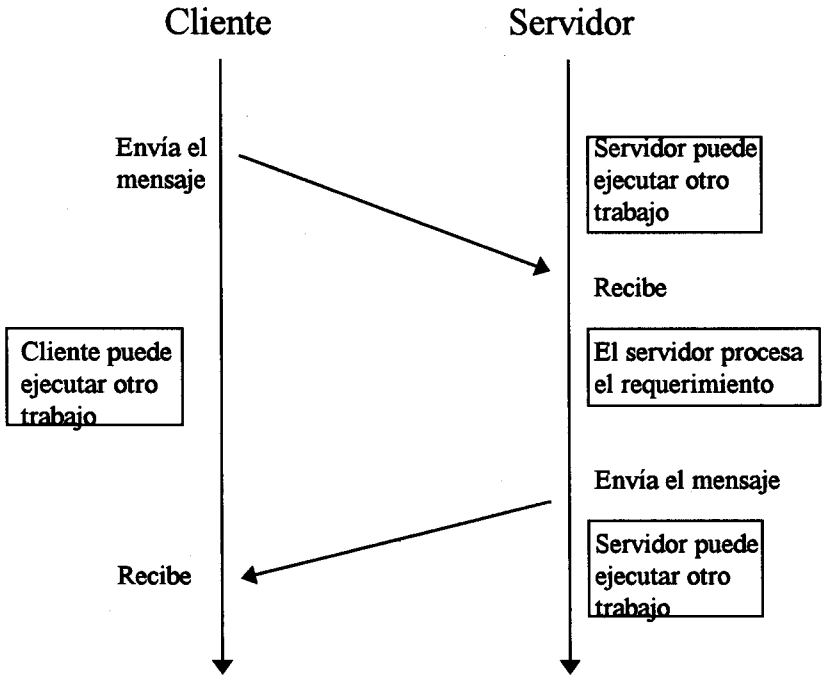


Figura 4. Comunicación de intercambio entre procesos utilizando "nonblocking"

2.3 Beneficios de Cliente/Servidor

Existen pocas personas que tienen desacuerdos en cuanto a que la implementación de Cliente/Servidor pueda dar buenos resultados en la actualidad y en ahorros futuros, pero esta nueva tecnología no puede ser justificada bajo un análisis de costo/beneficio solamente. Sino que existen algunos otros parámetros que indican los beneficios que ofrece esta tecnología, los cuales son: [TRAVIS, 1993]

Ahorro en dinero: este ahorro puede realizarse al migrar aplicaciones basadas en "mainframes" a ambientes Cliente/Servidor. Los ambientes "mainframe" son costosos de mantener, tanto por el "hardware", el "software", y el "staff" requerido para mantener y desarrollar aplicaciones las cuales son realmente caras. En cambio, con poco "staff" puede mantenerse la plataforma Cliente/Servidor.
Ahora bien, por ejemplo, el costo de mantenimiento de un servidor Cliente/Servidor no se compara con el costo de un servidor "mainframe". Incluso el hecho de cambiar un servidor Cliente/Servidor por otro, es más fácil hacerlo que el de cambiar un "mainframe", lo cual resultaría excesivamente caro. El caso de PETROCEL en Tampico, es un claro ejemplo de esto, ya que, en una entrevista realizada al Ing. Félix Lavín, Gerente de Sistemas de dicha empresa, comentó lo siguiente...."se reemplazó una IBM (Mini) por una SPARC debido a los altos costos de mantenimiento que se tenía, además de que si emigramos hacia Cliente/Servidor nos será más fácil el desarrollo de aplicaciones en la SPARC"...

En cuanto al desarrollo de aplicaciones, en Cliente/Servidor se desarrolla en menos tiempo que en ambientes "mainframe". Ya que el "front-end" (programación sobre o al frente de la aplicación) y el "back-end" (programación interna o detrás de la aplicación) pueden ser realizados como procesos en forma independiente, sin necesidad de tener todo en la misma aplicación. En la empresa Consultores en Informática Empresarial de Tampico, se realizó una entrevista con el Ing. Ponciano Betancourt, socio de la empresa y Director de Tecnologías y Desarrollo de Sistemas Computacionales de la misma, en donde indicó un claro ejemplo del ahorro en costos.... "ahora que emigramos a Cliente/Servidor podemos darnos cuenta el gran ahorro en costos que tenemos al desarrollar

aplicaciones a nuestros clientes, ya que cuando trabajábamos bajo herramientas típicas como DATAFLEX ó CLARION, entre otras, nuestro desempeño y productividad era lento en comparación con las herramientas de GUPTA que hoy utilizamos, y esto a su vez, se ve reflejado en los precios de las aplicaciones que les vendemos a los clientes; porque, existe mayor rapidez al realizar cambios o crear nuevas aplicaciones. Por lo que, puedo decir que nuestro aumento de productividad con herramientas Cliente/Servidor y del ahorro en costos es casi de un 40%"...

Incremento de la Productividad: tanto los usuarios como los desarrolladores son más productivos utilizando las herramientas de Cliente/Servidor. Esto debido a que, los usuarios se involucran más en el desarrollo de los procesos y el control de la aplicación. Se tienen accesos transparentes a los datos que los usuarios requieren para realizar sus trabajos, teniendo un estándar y una interface fácil de usar para el manejo de la información.

La productividad del usuario final se debe prácticamente a la facilidad de uso que se obtiene del manejo de ventanas, tales como, Microsoft Windows basados en DOS ver 3.x, Presentation Manager de IBM basados en OS/2, y Open Look y Motif basados en UNIX. Estos tipos de interfaces son manejados por "mouse", teniendo ventajas por los tipos de colores, y la presentación de grupos de datos a través de módulos, ventanas y el uso de menús o íconos.
Algunos estudios han demostrado que los usuarios presentan mayor productividad utilizando Interface Gráfica a Usuarios ("Graphical User Interface", GUI), tanto por el manejo de gráficos como la rapidez con que pueden obtener la información necesaria para realizar su trabajo.

La productividad del desarrollador se debe, en gran parte, a que las herramientas Cliente/Servidor utilizan básicamente "Structured Query Language" (SQL), la cual es una herramienta poderosa que ayuda a reducir el código de programación.

Flexibilidad y Escalabilidad: es importante para toda organización el poder emigrar a nuevas tecnologías teniendo pocos problemas para lograrlo. Esto es, por ejemplo, que una aplicación no tenga que ser rediseñada para usar una nueva interface o para moverla de plataforma. O bien, el hecho de actualizar un servidor y que genere un gran impacto en las aplicaciones.

Cliente/Servidor es modular. Ya que al agregar nuevos componentes o reemplazar los existentes es tan directo como agregar o reemplazar componentes en un sistema de sonido (estéreo). Las organizaciones fácilmente pueden actualizar componentes o emigrar a nuevas tecnologías sin cambiar las especificaciones de la aplicación.

Los usuarios pueden cambiar sus propias interfaces: GUI, Windows o Motif, pantallas texto o el formato de hoja electrónica de cálculo; esto debido a que, los usuarios tienen más control sobre su trabajo con sus aplicaciones y sus datos.

En base a que los estándares de las empresas creadoras de Cliente/Servidor evolucionan, las aplicaciones pueden correr en múltiples ambientes y en diferentes plataformas como si estuvieran corriendo en un mismo lugar, ya que las aplicaciones pueden ser soportadas fácilmente de un ambiente a otro. Es importante indicar que es una gran vantaja el hecho de desarrollar aplicaciones para diferentes tipos de servidores que tengan un SQL Server.

Utilización de Recursos: la primer implementación de Cliente/Servidor en una organización puede no requerir equipo nuevo. Una de las grandes características de Cliente/Servidor es que es capaz de ligar el equipo existente con las aplicaciones.

Los sistemas operativos de servidores de 32 bits, tales como, UNIX, Microsoft Windows, y OS/2 versión 2.0 de IBM; pueden mejorar el desempeño de los servidores, permitiéndoles realizar más manipulación de datos y la administración de las funciones de los datos. Las rutinas que son compartidas o que pueden ser optimizadas pueden ser pre-compiladas y almacenadas en el servidor. Estos procedimientos de almacenamiento deben ser llamados por las aplicaciones que los requieran y ejecutar el código compilado.

La arquitectura Cliente/Servidor está diseñada para usarse en forma eficiente en redes, minimizando cualquier embotellamiento o tráfico de información.

Control Centralizado: durante un tiempo existió la descentralización de información debido a la seguridad de la misma, sin embargo era muy difícil compartir la información o el utilizar estándares de control y de procedimientos. Debido a esto, se llegó a la centralización. Sin embargo en muchas organizaciones requieren tener estos dos tipos de control, y la arquitectura Cliente/Servidor facilita la combinación de ambos.

La parte de descentralización del sistema computacional consiste en el proceso hecho por el cliente la cual es responsabilidad de la empresa y del usuario. La parte de centralización del sistema consiste en la ejecución del proceso en el servidor la cual es responsabilidad de los profesionales de sistemas de información, ya que estos hacen las ligas con el servidor, el cual mantiene una base de datos inteligente con reglas y códigos de seguridad . El resultado final de esto, es que los sistemas de información obtienen control centralizado sobre los datos y los usuarios se enfocan a sus requerimientos de la empresa.
Un aspecto importante es que el respaldo y la recuperación de información son procedimientos centralizados, evitando así la pérdida de información o la duplicidad de la misma.

Sistemas Abiertos: para que la arquitectura Cliente/Servidor sea efectiva y convincente para cualquier organización, deberá soportar múltiples ambientes. Debe soportar a los diversos tipos de "hardware", múltiples plataformas GUI, múltiples sistemas operativos, múltiples Administradores de Sistemas de Bases de Datos ("Data Base Management Systems", DBMS), múltiples protocolos de comunicación y múltiples sistemas operativos de red. Un ejemplo de esto, es el caso del ITESM Campus Tampico, en donde hace tan sólo 1 año cambiaron su herramienta de desarrollo a PROGRESS bajo UNIX y que actualmente están analizando la posibilidad de cambiar a PROGRESS con "Windows". El soporte en "hardware" sería el mismo (equipos 486), así como los protocolos de comunicación (ethernet), lo único que cambiaría es el incremento en RAM en algunas estaciones cliente y el ambiente gráfico de trabajo para los usuarios finales.

2.4 Downsizing.

Las nuevas tendencias que han asumido los gerentes de sistemas se basan en el proceso distribuido. El crecimiento de las redes distribuidas cliente/servidor ofrecen a los usuarios varias alternativas para mezclar y complementar los elementos tradicionales de la computación.

El "Downsizing", es aquello que al no dar un valor agregado a la producción deberá eliminarse. [RED, #25,1993]

Uno de los objetivos que suceden en el área de informática al hacer "Downsizing", es el mover las aplicaciones que ocurren en un "mainframe" o minicomputadora a un servidor de base de datos de red. Los datos se almacenan localmente para aprovechar el poder de procesamiento de las computadoras personales y evitar así los costos de mantener datos en minicomputadoras y "mainframes".

Otro de los objetivos de hacer "Downsizing" es migrar los procesos y datos a equipos y programas que se encuentren más cerca de los usuarios, que posean interfaces gráficas y que substituyan a las terminales tontas.

"Downsizing" es un concepto más global del procesamiento distribuido. "Downsizing" considera el uso de bases de datos en red de alto rendimiento y estaciones de trabajo gráficas para hacer más atractivas y productivas las aplicaciones críticas de la empresa. Estas bases de datos pueden ser utilizadas en: aplicaciones tipo hoja de cálculo, procesadores de palabra y herramientas de búsqueda y reporte.

"Downsizing", ha generado discusiones políticas y emocionales, cambiando carreras y formas. Los gerentes de sistemas, quienes tenían el poder de la informática de la empresa en sus manos, están viendo como éste se diluye en cada departamento.

La tendencia es evitar el uso de "mainframes", pero no todas las compañías lo han hecho. Muchas utilizan sus poderosísimos "mainframes" como superservidores para sus redes metropolitanas y de área amplia.

Para poder establecer "Downsizing" es importante considerar lo siguiente: [RED, #25,1993]

1. Seleccionar la plataforma adecuada; es importante considerar tanto "hardware", "software", sistema operativo, administración e interoperabilidad.
2. Establecer las bases para la interoperabilidad; es importante el preparar la plataforma para que todos los equipos se comuniquen.
3. Seleccionar las aplicaciones; se deben evaluar las aplicaciones que podrían ser "Downsizing".
4. Seleccionar las nuevas aplicaciones; es necesario evaluar cargas de trabajo y flujo de información para que las aplicaciones resuelvan problemas particulares.

El efectuar "Downsizing" implica mucho trabajo, ya que el departamento de sistemas establece un contacto estrecho con el personal que se encarga del servidor, aplicaciones y datos, para lograr un resultado satisfactorio. Por lo que es importante reconocer que, el nuevo sistema no debe satisfacer a la gente de sistemas sino a los usuarios.

Es conveniente mencionar el caso de SECOFI, que optó por el "Downsizing". En 1988 trabajaban bajo una arquitectura cerrada y con una incipiente infraestructura de comunicaciones, contaban con 66 terminales, y 185 especialistas en informática. Todas las actividades informáticas estaban centralizadas, y existía una falta de capacitación de personal. Ante esto decidieron establecer el Programa de Modernización Informática, el cual consistió en dotar infraestructura de cómputo a las unidades administrativas; ampliación de infraestructura de comunicaciones, para intercambios de información. A partir de este momento tenían dos opciones a considerar: UNIX o redes. Optaron por redes ya que tenía las siguientes ventajas: manejo de ambiente gráfico, independencia con los proveedores (por ser abierto) y portabilidad. [COMPUTER WORLD, Num. 387, 1993]

A pesar de lo positivo del "Downsizing", existen algunos aspectos negativos del mismo, como por ejemplo: [COMPUTER WORLD, Num. 387, 1993]

* Se disponen pocas herramientas para la conversión de aplicaciones y datos. La conversión ha sido ante todo un esfuerzo de reprogramación.

* Muchas aplicaciones que corren en sistemas abiertos SMP (Sistemas Abiertos de Multiproceso Simétrico) exigen coexistencia con el "mainframe" y acceso a los datos manejados en el "mainframe". Los datos se extraen y descargan por la noche, un proceso prolongado y tedioso. La coexistencia exige demasiado trabajo.
* Resulta difícil la solución de problemas y la sintonización de aplicaciones para bases de datos en sistemas SMP, porque aún no se cuenta con herramientas de servicio. Generalmente, la solución a los innumerables problemas es una restauración de la base de datos.
* Los sistemas se caen con mucha frecuencia. Se muestran muy sensibles y no son tan robustos como los sistemas "mainframe".
* Los recursos de "hardware" requeridos para soportar la carga de trabajo, como poder de procesamiento y capacidad de disco, han sido mayores de lo anticipado.

"Downsizing" se debe presentar cuando las necesidades de las empresas lo requieran, para lo cual es conveniente el considerar la plataforma adecuada, el "hardware", "software", seleccionar las aplicaciones que estén funcionando y que requieran cambiar, así como también las nuevas aplicaciones.

Existe un concepto adicional e importante que va ligado al "Downsizing" el cual es el "Rightsizing", en donde a continuación se explicará.

2.5 Rightsizing.

El hablar de "Rightsizing" es hablar del punto medio entre "Downsizing" y "Upsizing". En donde "Downsizing", es el migrar aplicaciones que están en una plataforma cara a una barata. "Upsizing", significa tener expectativas de crecimiento hacia plataformas más grandes. "Rightsizing", significa utilizar mejor los recursos con los que hasta hoy cuenta la empresa . Es decir, pretende una estrategia evolutiva: ir del "mainframe" hacia abajo con equipos más pequeños y hacia arriba con equipos más grandes. [COMPUTER WORLD, Num. 387, 1993]

El paso más importante al iniciar "Rightsizing", es el establecer un buen programa de concientización y de apoyo por parte de toda la organización. Este programa debe comprender los siguientes objetivos: [UNIX, 1993]

* reducción de costos,
* asignación de responsabilidades,
* alta calidad,
* eliminar el "software" obsoleto y no obsoleto y,
* uso de "software" comprado

El "Rightsizing" debe ser un esfuerzo de equipo de todo el departamento de informática, logrando con ello: reducción de costos y creando ventajas competitivas. El "Rightsizing", no es una planeación de Sistemas de Información (SI) es una planeación completa de toda la organización.

Para obtener beneficios a corto plazo se requieren establecer objetivos fundamentales. El primer objetivo es absoluto: todo fuera del "mainframe". Esto puede sonar extremista, pero es realizable en la mayoría de las empresas.

El staff de SI no debe ser polarizado. Por ejemplo, el staff no debe ser dividido en el grupo UNIX y el grupo "mainframe". Todos deben tener el mismo objetivo.

Otro concepto básico de "Rightsizing", es proveer fácil acceso a la aplicación. Asumiendo que los beneficios de reducción de costos de adquisición de software son parte del plan de "Rightsizing".

Otro aspecto importante, es el encontrar un producto que se aproxime a las necesidades operacionales de la organización que puede ser modificada. El ambiente de "Rightsizing" debe ser tan abierto como sea posible para permitir la evaluación de diversos productos candidatos. En la práctica se debe establecer un Ambiente Estándar de Operación (AEO). Bajo el enfoque AEO la organización SI busca un lenguaje de desarrollo y una base de datos estándares.

Una organización puede utilizar dos enfoques para efectuar "Rightsizing":

1) Contratar nuevo personal con conocimientos para el "Rightsizing". Este enfoque trae experiencia inmediata pero crea algunos problemas:

* aliena la existencia del personal SI
* añade costo cuando el objetivo es reducirlo
* el reclutamiento de un nuevo equipo de desarrollo está sujeto a que emigren con la competencia

2) El enfoque ideal es el reentrenamiento del personal existente en las nuevas herramientas.

Algunos puntos estratégicos a considerar para implementar "Rightsizing", son los siguientes:

* considerar el "software" a convertir a las necesidades de la empresa
* entrenamiento de personal adicional para la instalación

Hoy en día las organizaciones han realizado "Downsizing" para aprovechar los recursos de Cliente/Servidor, otras tratan de aplicar "Rightsizing" para aprovechar los mismos recursos con que cuenta, en forma óptima. Sin embargo, cada organización deberá hacer un balance sobre qué es lo que más le conviene realizar, y evitar el cambiar o emigrar hacia algo que tal vez sólo le dé gastos innecesarios en lugar de inversiones futuras.

Dentro de lo que es Cliente/Servidor nace un concepto nuevo e interesante, lo cual hace atractivo el uso de ciertas herramientas para el desarrollo de aplicaciones orientadas a usuarios finales, las cuales son las interfaces gráficas a usuarios, que a continuación se mencionarán.

2.6 Interfaces Gráficas a Usuarios

2.6.1 Definición

Las interfaces gráficas presentan al usuario información en forma de ventanas, las cuales son áreas rectangulares sobre la pantalla de la computadora en donde se utiliza el "mouse" (ratón) para poder seleccionar la información. [TRAVIS, 1993]

Dentro de estas ventanas se encontrarán pequeños dibujos o íconos alusivos a la información que se esté visualizando. La figura 5 y 6, muestran un ejemplo de esto:

Catálogo de Cuentas :

Figura 5. Diseño de Ventana: Interface Gráfica a Usuarios

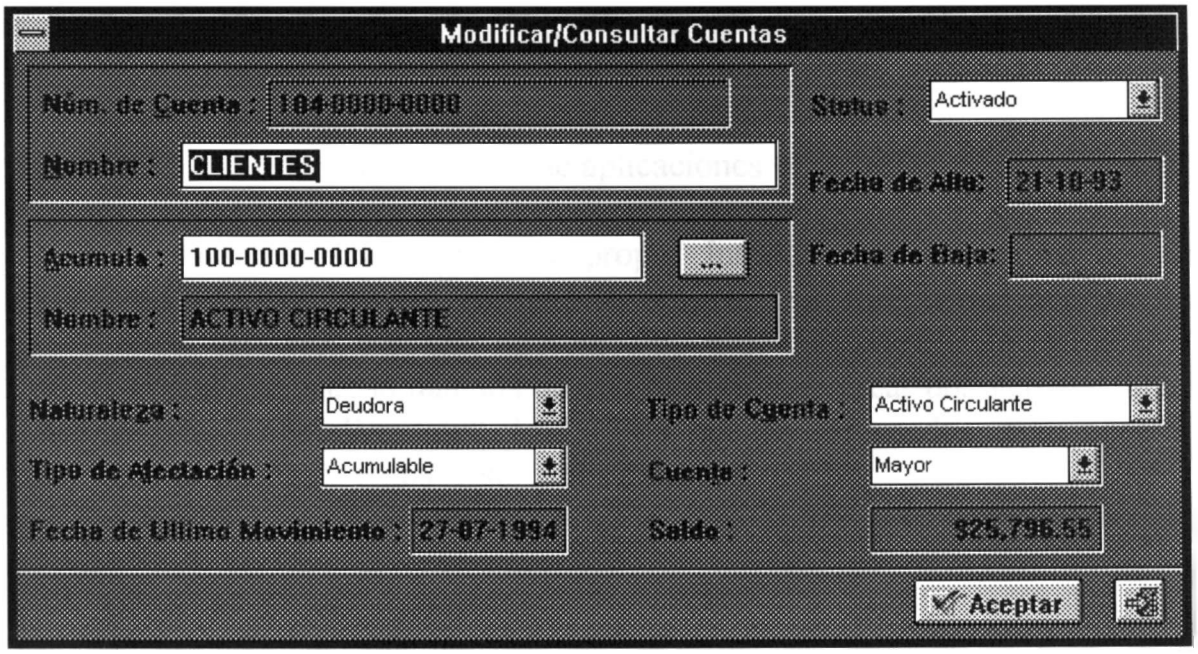


Figura 6. Diseño Ventana: "GUI" de captura

2.6.2 Características básicas.

El utilizar GUI's brinda ventajas al usuario por tener las siguientes características: cambios de color, tamaños de íconos, cambios de posición de una ventana, poder abrir una ventana nueva, miniaturizar o maximizar ventanas, utilizar ventanas con "scroll", entre otras.

Una gran diferencia con el uso de GUI y los sistemas tradicionales basados en pantallas texto, es que el usuario siempre mantiene y maneja el control del sistema ya que puede cambiar de aplicación en cualquier momento. El uso de GUI está orientado a usuarios no computacionales, a quienes les facilita la interacción con el sistema de información.

Los GUI's son controlados a través de manejadores de eventos, algunos de estos son:

* eventos de "mouse",
* eventos de teclado,
* eventos de actualización de ventanas,

* eventos de cambios en tamaños de ventanas,
* eventos de activar/desactivar ventanas,
* eventos de menús y,
* eventos de inicio/fin de aplicaciones

Cada ambiente GUI tiene su propia Interface de Programación de Aplicaciones (API). Un API consiste de un grupo de rutinas que son usadas para proveer servicios y el unir diferentes tipos de "software". Un ejemplo de esto, es cuando se hace la llamada de una función para crear una ventana dicha función sería "CreateWindow". Los API's pueden ser usados: para accesar servidores de bases de datos y servicios de redes.

2.6.3 Necesidad de estándares.

Las interfaces consistentes entre los usuarios y las aplicaciones representan el requerimiento llave para los sistemas abiertos. Este requerimiento es importante desde el momento en que las interfaces afectan tanto a los desarrolladores de sistemas como a los usuarios finales, esto, debido a que los programas o las interfaces pueden esconderse para que el usuario final no vea lo que se realizó sin embargo, la presentación de la aplicación es visible para cualquier persona, sea desarrollador o usuario del sistema.

La gran variedad de interfaces puede confundir a los usuarios y desarrolladores, ya que cada interface nueva a utilizar o desarrollar, requiere el reentrenamiento de éstos. Una nueva GUI puede causar el rehacer toda una aplicación en una plataforma nueva, esto debido a que, la gran mayoría de aplicaciones con GUI no son portables a otros ambientes GUI's.

Debido a esto, un estándar GUI con un simple API y un estándar "look and feel" puede tener un gran impacto en la productividad de los desarrolladores y de los usarios.

La aceptación de las industrias, el soporte en "hardware" y "software" por parte de los proveedores y los usuarios finales de las organizaciones opinan que deberán establecerse estándares para los GUI's. Un estándar GUI para

sistemas abiertos deberá satisfacer los siguientes requerimientos: [BERSON, 1992]

Portabilidad: tanto las aplicaciones como los usarios deben ser portables a través de cualquier plataforma, reduciendo la necesidad de reentrenamiento y la facilidad de aprendizaje.

Herramientas de desarrollo: cualquier GUI debe tener un conjunto de herramientas de desarrollo que brinden lo siguiente: facilidad de desarrollo de aplicaciones y construcción de aplicaciones sobre cualquier plataforma.

Flexibilidad: un estándar GUI debe ser flexible para dar el poder suficiente de agregar: nuevos despliegues en pantalla y dispositivos de entrada/salida

Internacionalización: esta va muy ligada con la portabilidad de la aplicación, e incluye: lenguajes del país, números, unidades monetarias, formatos de fecha y hora, símbolos especiales y los mensajes especiales que la cultura del país utilice.

Independencia de la plataforma: para que realmente sea abierto y estándar, un GUI debe diseñarse para operar independientemente del sistema operativo o de la plataforma de "hardware" en que esté corriendo. En forma similar, en un ambiente de red, el estándar de un GUI debe operar independientemente de los protocolos de red.

2.6.4 Ambientes GUI

2.6.4.1 Windows 3.x

Dos de las fuerzas "Microsoft Windows" son el poder de la administración de la memoria y su protocolo "Dynamic Data Exchange" (DDE), la cual soporta ligas entre programas.

"Windows" utiliza dos tipos de combinaciones de memoria: local y global. La local es específica para la aplicación. La global puede ser accesada por cualquier programa en el sistema. La memoria local es más rápida y más eficiente pero limitada a trozos de 64 kbytes, una limitante del sistema operativo DOS.

Existen algunas características importantes de este ambiente:
[TRAVIS, 1993]

"Dynamic Link Libraries" (DLL): "Windows" soporta "Dynamic Link Libraries" (DLL), las cuales son rutinas de código modulares, estas mejoran el "performance" de la ejecución y el mantenimiento de una aplicación.
Un cambio en algún módulo requiere el volver a ligar ("linkear") las librerías que forman parte de este. Los DLL pueden ser cargados con las aplicaciones que los necesiten.

"Dynamic Data Exchange" (DDE): esta provee el intercambio de información en forma automática entre aplicaciones. Los DDE pueden ser usados para ligar un grupo de celdas entre dos programas, como Excel y Word de Microsoft para Windows. Esto es, si la parte de la hoja de cálculo o una gráfica de los datos de la hoja de cálculo es cargada dentro de un documento de Word y los números de la hoja cambian, el área correspondiente al documento es actualizada automáticamente.

"Object Linking and Embedding" (OLE): la tecnología OLE de Microsoft muestra a los usuarios el construir documentos combinados y enfocar sus datos (palabras, números y gráficas) más que sus aplicaciones. Un documento es tratado como una colección de objetos, más que un archivo. Cada objeto guarda una liga a la herramienta que se usó para crearlo. Un doble "click" (con el "mouse") sobre el objeto empieza a originar la aplicación, mostrando al usuario la modificación al objeto. La figura 7, muestra las diferencias entre DDE y OLE.

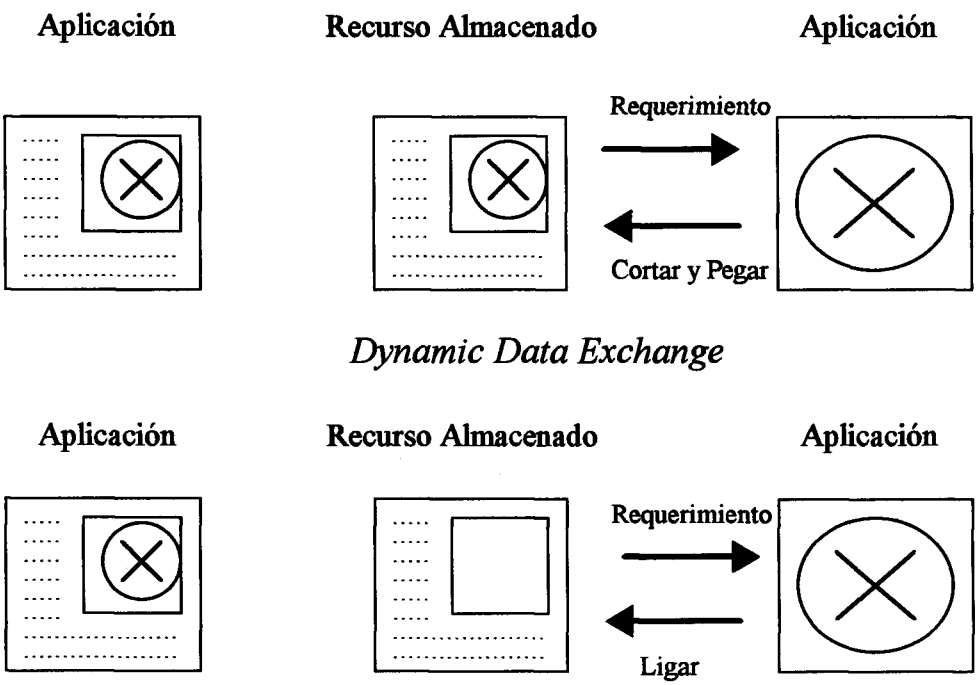


Figura 7. Diferencias entre DDE y OLE.

Cuando el dato es ligado usando OLE, "Windows" sólo almacena un apuntador al dato. La vista del dato ligado es automáticamente refrescada cuando el dato original es modificado. Sin embargo, si la aplicación ligada está fuera de línea de la aplicación fuente, el dato no puede ser accesado por la aplicación ligada. Cuando el dato es incrustado, "Windows" almacena una copia del dato en la aplicación. El dato no es refrescado hasta que el usuario haga doble "click" sobre el dato para empezar a originar la aplicación.

Ventanas API's: la base de la interacción de "Windows" en red es el "driver Winnet", el cual es un conjunto de API's que hacen el puente entre el "front-end" de "Windows" con el sistema operativo de la red. Para utilizar "Winnet", el cual forma parte de "Device Driver Kit" de "Windows", un desarrollador puede escribir aplicaciones de red que no son específicas de un sistema operativo de red.

Los proveedores agregan sus propios enfoques, como Novell que tiene sus propias utilerías de "Windows"; LAN Manager usa "Winnet API's" dentro de servidores de red.

Actualizaciones de "Windows": Microsoft ha anunciado una versión de manejo de objetos para "Windows", llamada Cairo. Este utiliza un sistema de archivo objeto que almacena datos y componentes de aplicaciones como objetos y ligas entre objetos. Cairo utiliza un recuperador de datos "front-end" que permite a los usuarios buscar información usando atributos como el tópico o el autor. Cairo incluye un modelo de objeto nuevo, servicios de directorio y seguridad distribuida.

Aunque las versiones de Cairo han sido originalmente planeadas para DOS y Windows NT, Microsoft decidió lanzar el sistema sobre Windows NT. Debido a esto, los desarrolladores de "sofware" han presentado objeciones al respecto y Microsoft ha estado tratando de liberar uno en Windows para DOS.

Nueva Tecnología "Windows": el sistema operativo Windows NT de 32-bits de Microsoft puede eliminar las limitantes de DOS. Sin embargo, Microsoft afirma que Windows NT no está diseñado para reemplazar "Windows" para máquinas de escritorio; está diseñado como un sistema operativo para servidores y estaciones de trabajo.

2.6.4.2 Presentation Manager

OS/2 de IBM y Presentation Manager son usualmente consideradas como paquetes distribuidos. Como una parte de OS/2, Presentation Manager ha construido el soporte para características de OS/2, como multitareas, librerías con ligas dinámicas, y los "pipes" para comuncar procesos entre máquinas.

Contrario a las creencias populares, las aplicaciones efectivas de Presentation Manager no pueden ser hechas para DOS, Windows, o Macintosh.

Para correr bajo Presentation Manager, las aplicaciones son registradas y asignadas a un manejador de bloques, el cual es usado para ligar tipos de ventanas a la aplicación.

Las clases de Presentation Manager y los objetos relacionados a las ventanas y sus tipos, no necesariamente son clases orientadas a objetos. Cada clase de ventana tiene un nombre que lo identifica.

La cola de mensajes para la aplicación y las ventanas necesarias para esta, son construidas.

Porque OS/2 es multilecturas, cada aplicación tiene un apuntador primario y puede tener apuntadores secundarios. El sistema de mensajes se obtiene por medio de la entrada de los usuarios (vía "mouse" o teclado) y dirige el mensaje a la máquina primaria de la aplicación relevante.

Presentation Manager traduce esta entrada al sistema de mensajes, los cuales son almacenados en la cola y procesados al mismo tiempo con el ruteador interno.

El ruteador interno direcciona los mensajes a la cola de mensajes apropiada de la aplicación.

Cuando el mensaje es recibido por una aplicación, éste se limpia y se destruye de la cola de mensajes y del manejador de bloques, Presentation Manager notifica que la aplicación no está activa para la sesión.

Multilecturas es una de las principales características de OS/2 y representa una gran diferencia con Windows.

2.6.4.3 Motif

Motif de Open Software Foundation (OSF) es implementado para usar un simple "Application Programming Interface" (API) que pueda ser soportado en todas las plataformas. Motif, basado en la tecnología de ventanas de DEC, fue mejorado para soportar el estilo de Presentation Manager. Motif está basado en los estándares de OSF y está considerado como una tecnología que permite ser abierta.

Motif incluye las siguientes herramientas:

"User Interface X Toolkit": contiene los objetos gráficos que usa Motif.

"User Interface Language" (UIL): lenguaje utilizado para describir los aspectos visuales de una aplicación, tales como menús, formas, etiquetas, y botones.

Los desarrolladores crean un archivo texto que contiene una descripción de cada objeto y con sus funciones asociadas. El archivo texto es compilado por el UIL dentro de un archivo fuente, el cual es cargado en el "runtime".

"Window Manager": muestra las ventanas a ser movidas, ampliadas, y para reducir los íconos. el "Window Manager", le da a Motif la apariencia tridimensional.

2.6.4.4 OpenLook

OpenLook fue diseñado por Sun Microsystems y AT&T. OpenLook Window Manager es requerido para comunicaciones entre clientes.

Uno de los siguientes tres API's pueden ser usados para desarrollar aplicaciones con OpenLook:

"NeWS Development Environment": aplicación de la plataforma de SUN, que emula un interpretador "PostScript" para soportar un sistema de ventaneo. El cual consiste en que el cliente y el servidor interactúan para producir el GUI de OpenLook.

"Xt+": aplicación de la plataforma de AT&T, que contiene todos los objetos gráficos utilizados por OpenLook.

"Xview": aplicación de SUN SPARC y Digital VAX, para utilizarse en 80386, Motorola 680x0.

El uso de interfaces gráficas, es un factor de cambio importante para el usuario ya que marca la pauta de una forma de trabajar más simple, fácil y sencilla, entre otras.

Esto nos indica un cambio radical en el uso de aplicaciones actuales a las de hace unos cuantos años atrás, en donde el usuario dependía totalmente de alguna persona de sistemas. Con aplicaciones nuevas, donde predominan las

interfaces gráficas, y el usuario es capaz de crear sus propias aplicaciones sin necesidad de conocer a fondo algún lenguaje computacional.

Existe una gran cantidad de "software" que funcionan bajo diversas plataformas, las cuales brindan sus propias interfaces gráficas. Ejemplos de este "software" con sus características básicas, requerimientos de "hardware", así como sus proveedores se encuentran en el Anexo 1.

CAPÍTULO 3
DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR

3.1 Factores que afectan al Software

La planeación estratégica, el desarrollo, y cualquier otro soporte para las aplicaciones son procesos vitales que pueden brindar enormes beneficios en términos de ahorro de costos a las compañías, el poco tiempo para mercadear nuevos productos, y las grandes satisfacciones de los clientes, entre otros. Mientras que hay otros eventos que influyen y comprimen el tiempo planeado, la acción real es en el desarrollo de aplicaciones. En esta área es donde ciertos factores afectan al "software" y donde presentan más impacto hoy en día. [SMITH, 1992]

A continuación se hará mención de estos factores:

Aumento de los costos del staff de tecnología: debido al incremento de la demanda por el desarrollo de sistemas, y que el grupo de personas que se dedican al área técnica no se dan abasto para cubrir la demanda, da como resultado que el costo de los técnicos aumente más rápido que la inflación. Muchas organizaciones encuentran que los profesionales en tecnología, cambian de trabajo cada tres o cuatro años, lo cual trae como consecuencia capacitaciones y entrenamientos a personal nuevo.

Presión por construir ventaja competitiva para liberar sistemas: hay una gran presión de las organizaciones por tomar ventaja de la nueva tecnología para crear ventaja competitiva. Esto puede crearse al brindar servicios innovadores antes que la competencia.

En algunos estudios realizados, todavía existen compañías desarrolladoras de "software" que del 80% al 90% de los sistemas tradicionales que realizan están dedicados a dar mantenimiento, y sólo del 20% o el 10% restante lo dedican a desarrollar nuevas aplicaciones.

Para muchas organizaciones, el implementar sistemas no sólo incrementa eficiencia y efectividad, sino que también transforma procesos fundamentales para crear ventaja competitiva como *algo absolutamente fundamental para sobrevivir.* Aquellas organizaciones que no puedan encontrar nuevas formas para refinar los procesos de sus negocios y mantener la cadena del valor, rápidamente fallarán a lado de otras compañías que sí lo hagan.

Necesidad para mejorar la productividad de profesionales en tecnología: El "Index Group" reporta que la tecnología CASE (Computer-Aided Software Engineering) y otras tecnologías que desarrollan "software" con rapidez son mencionadas por el 70% de los altos ejecutivos de Tecnología de Información que examinaron como las más críticas tecnologías de implementación. El mercado CASE está creciendo a una tasa de 30% por año, y se predice que para 1995 tenga un mercado de $5 billones de dólares.

Esta nueva producción de herramientas de "software" ayudan a las organizaciones a responder más rápidamente, acortando el tiempo que toma el crear nuevas aplicaciones o dar mantenimiento.

Las compañías necesitan innovaciones, sistemas de misión crítica para desarrollar rápidamente, con alta productividad, y con un staff de profesionales que trabaje junto con el usuario durante la fase de diseño. El modelo de desarrollo cliente/servidor, provee el significado de desarrollo de prototipo horizontal de una aplicación cuando se está diseñando. La presentación visual en una estación y el colaborar en la etapa de diseño, le da al usuario un cambio total de visualización y creación de las aplicaciones que en los métodos tradicionales.

Necesidad para emigrar a otras plataformas y reingeniería de sistemas existentes: en ocasiones el mantener una plataforma es más costosa que el cambiar a otra, dando lugar al "downsizing". Por lo que es necesario estudiar bajo costo-beneficio el cambio a realizar.

Por otro lado, el ciclo de vida de las aplicaciones es de 5 a 15 años, mientras que la de la tecnología es más corta, de 1 a 3 años. Los

grandes avances pueden dar lugar la reingeniería de las aplicaciones existentes.

Necesidad para una interface común a través de plataformas: en 1991 el "Information Week" estudió que 157 ejecutivos de Tecnología de Información revelaron que el gran criterio de compra de "software" (más que el criterio de *facilidad de utilización*) es el uso común de interfaces entre plataformas.

Incremento en el desarrollo de aplicaciones por los usuarios: debido a la gran cantidad de software orientado al usuario y a la gran iniciativa por capacitarse en computación, hoy en día cualquier usuario puede realizar sus propias aplicaciones sujetas a sus necesidades, por lo que el departamento de Tecnología de Información está funcionando como utilidad, para brindar otros servicios.

3.2 Hardware/Network

Antes de seleccionar el "hardware" para usuarios finales, muchas organizaciones deben definir estándares para las clases de usuarios. Este conjunto de estándares simplifica la selección del "hardware" apropiado para los usuarios. [SMITH, 1992]

Existen algunos aspectos a considerar cuando se selecciona: la estación cliente, tipo de procesador, capacidad del coprocesador, estructura interna del bus, el tamaño de la unidad base, entre otros.

Algunos puntos importante sobre el "hardware" son los siguientes:

Unidades de procesamiento a nivel computadora personal: las aplicaciones cliente/servidor varían considerablemente en los requerimientos de procesamiento del cliente y en sus demandas de I/O (input/output) del procesador del cliente y el servidor. En general, las estaciones clientes que soportan direccionamiento en modo protegido deben ser comprados. Esto implica que el usar

procesadores de 32 bits, quizá con un bus I/O de 16 bits si el requerimiento de I/O es lento, significa que el cliente no requiere enviar y recibir grandes cantidades de datos, como las imágenes de más de 100 kbytes.

Macintosh: el sistema operativo MAC 7 es altamente visual y provee la mejor productividad cuando responde a operaciones "GUI". Se recomienda Motorola 68030, 6 Mbytes de RAM, 40 Mbytes en disco.

Notebooks: el uso de este tipo de equipos cada vez es más usual, sin embargo la velocidad de I/O del disco es lenta, lo cual hace que se instale más RAM y el crear "drives" virtuales. Por ejemplo, si alguna aplicación corre un "GUI" remoto, es deseable instalar "software" que comprima el "GUI" y la comunicación del modem V.32 a 9600 bps o V.32bis a 14400 bps usando comprensión V.42 y V.42bis, respectivamente. El efectivo "throughput" es dos o tres veces el "baud rate" por la comprensión.

Pen: los clientes basados en Pen proveen la habilidad de operar aplicaciones usando una pluma para apuntar y seleccionar o escribir sin necesidad de "mouse" o teclado. Frecuentemente la usan para verificar, seleccionar, en aplicaciones donde las listas de selección son disponibles.

UNIX Workstation: estas estaciones son usadas cuando el cliente requiere procesamientos extensivos. En muchas aplicaciones de UNIX, usan las terminales X conectadas al servidor UNIX. Estas estaciones clientes tienen más poder de procesamiento que una computadora personal cliente.

X-Terminal: proveen la habilidad de ejecutar sólo servicios de presentación en las estaciones. Estos servicios son provistos por otras de UNIX, Windows 3.x, OS/2, o servidores VMS. Base de datos, comunicaciones, y servicios de aplicaciones son provistas por el mismo u otros servidores de la red. La memoria mínima de estas estaciones usando cliente/servidor es de 4 a 8 Mbytes de RAM, dependiendo del número de ventanas abiertas.

Los requerimientos del servidor varían de acuerdo a la complejidad de las aplicaciones y la distribución del trabajo. Esto es a causa de los dispositivos multiusuarios, y al número de usuarios activos.

Los requerimientos de almacenamiento permanente son específicos y requeridos por las aplicaciones. Esto es, la cantidad de almacenamiento en disco, los puntos de desempeño y rentabilidad deben considerarse ampliamente. Algunos aspectos a considerar en cuanto a almacenamiento de datos son los siguientes:

Disco Magnético: los dispositivos de almacenamiento en disco usan el controlador SCSI-2 ("Small Computer System Interface"). Este provee el mejor desempeño en ambientes basados en estándares, gran capacidad, y rentabilidad de disco.

Discos Espejo: cuando las aplicaciones requieren alta rentabilidad, puede ser apropiado el usar una configuración que soporte discos espejo. Con esta configuración, los datos se escriben automáticamente en dos discos, y la gran ventaja surge cuando falla un disco el otro queda trabajando siendo esto transparente al usuario.

RAID- Arreglo de discos: el alto desempeño y la alta disponibilidad pueden llevar a cabo el uso de RAID ("Redundant Array of Inexpensive Drives"). Esto muestra archivos de datos físicamente separados en discos, donde los datos se espejean como parte de la configuración. Esta tecnología provee un gran desempeño porque muchas operaciones I/O pueden procesarse al mismo tiempo.

Cintas: es la más popular y de bajo costo en cuanto a almacenamiento magnético, con propósitos de uso para respaldos.

Discos Ópticos: esta tecnología provee la ventaja del gran volumen de almacenamiento.

CD-ROM: el "compact disk" de memoria sólo lectura ("Compact Disk- Read Only Memory") son usados para almacenar información que sólo sea para leer. Es más rentable que los discos magnéticos o las cintas.

WORM: Escribir una vez, leer varias veces ("Write Once Read Many") son "drives" ópticos para almacenar información. Se utiliza para archivar datos que no podrán modificarse.

Erasable Optical: estos "drives" ópticos borrables, son usados como una alternativa para discos magnéticos cuando la velocidad de acceso no importa y el volumen de datos almacenados es grande.

Los procesadores de cliente y servidor son asignados a una red local a través de NIC's ("Network Interface Cards"). Estas proveen la conectividad física en la red y el soporte del protocolo para el envío y recepción de mensajes. Los siguientes protocolos de red son los más comunes hoy en día:

Token Ring: son productos originales de IBM los cuales se usan en computadoras personales y bajo UNIX. Este protocolo se utiliza en las comunicaciones donde existe gran volumen de información a transferir o en impresiones de imágenes. Token Ring para red local opera a 4 o 16 Mbps (Mega bits por segundo).

Ethernet: esta ofrece un precio muy competitivo y su desempeño es excelente cuando no excede al 305 de su capacidad. Ethernet opera a 10 Mbps.

FDDI: ("Fiber Distributed Data Interchange") es un protocolo definido para comunicaciones de alta velocidad. Esta opera a 100 Mbps. Es necesaria en aplicaciones con grandes imágenes, grandes transferencias de archivos o multimedia usando "full-motion video".

CDDI: ("Copper Distributed Data Interchange") provee soporte de comunicaciones FDDI sobre líneas de cobre. También opera a 100 Mbps.

Es importante el mencionar a los dispositivos de protección del equipo, tales como: UPS ("Uninterruptible Power Supplies") y protectores de voltaje.

3.3 Soporte y Servicio

El hecho de tener tecnología cliente/servidor o cualquier otra, implica ciertos aspectos a considerar; ya que, tan importante es el tener el "hardware" adecuado e instalado como el "software" (vistos anteriormente), sin embargo la mayoría de los problemas que tienen los usuarios son causados por la falta de soporte y servicio a sus necesidades. En este aspecto se deben tomar en cuenta los siguientes puntos: [SMITH, 1992]

Disponibilidad: es la habilidad de que el sistema esté disponible para procesar información y de realizar el trabajo cuando se le requiera. Con la tecnología de hoy en día, las minicomputadoras y los "mainframe's" deben proveer 99.5% de disponibilidad.

Rentabilidad: las aplicaciones requieren rentabilidad a ser protegidas de re-escritura y el compartir memoria para accesar sólo tareas autorizadas. La seguridad debe implementarse al mostrar accesos a recursos a usuarios autorizados.

Servicios: tanto las mini's como los "mainframe's" proveen un servicio de diagnóstico para conocer los puntos (o localidades) en donde ocurren fallas.

Distribución del "software": depende de la plataforma sobre la cual se esté trabajando. En ambientes mini's o "mainframes" comparten la ejecución del "software" a partir de una librería; en el modelo cliente/servidor, el "software" reside en los servidores; en UNIX la complejidad crece debido a las diversas plataformas de "hardware" que son usadas. Muchas organizaciones instalan el "software" en cada una de las diversas plataformas de "hardware" que se tengan.

Desempeño: factor importante en el uso de cualquier plataforma, ya que esto marca el buen funcionamiento de "hardware".

Administración de red: algunas herramientas para administrar red son: IBM NetView, AT&T UNMA y DEC EMA; proveen un nivel remoto de monitoreo en tiempo de respuesta y carga de la red.

Ayuda de escritorio ("Help Desk"): la forma más eficiente y efectiva de dar soporte a usuarios cliente/servidor es a través del uso de "help desk". Este es un conjunto de sistemas y procedimientos usados por un grupo de personas técnicas o de staff que brindan ayuda a usuarios finales en ciertas áreas. Este tipo de soporte puede proveerse usando PC's remotas, ayuda telefónica, o asistencia personal.

Administración remota de sistemas: los administradores de redes locales deben conectarse remotamente para administrar las estaciones de los usuarios que tienen problemas. Algunos productos son: DCAF de IBM, PolyMod2 de Memsoft, Remote OS de Menlo.

Seguridad: aspecto más relevante en el uso de cualquier aplicación. Esta debe estar presente desde el momento de asignar una cuenta a un usuario con su "password" hasta el diseño de cada aplicación.

3.4 Metodologías para implantación de sistemas

3.4.1 Tradicional

Una de las metodologías más utilizadas al principio de la era de los sistemas de información, era el método tradicional el cual lo forman las siguientes etapas: [SENN, 1987] [BURCH,1991] [KENDALL,1991]

1) Identificación de Problemas, Oportunidades y Objetivos
2) Determinación de Requerimientos
3) Análisis
4) Diseño
5) Desarrollo y Documentación
6) Prueba y Mantenimiento
7) Implantación y Evaluación

La Figura 8 muestra una diagramación de estos pasos:

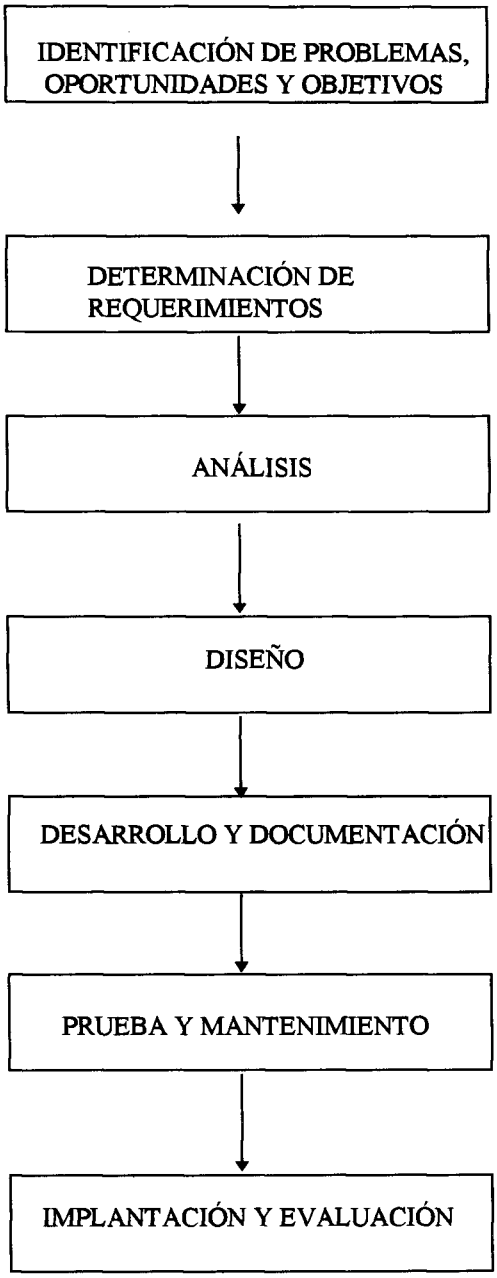


Figura 8. El Ciclo de Vida del Método Tradicional

1) Identificación de Problemas, Oportunidades y Objetivos: en esta primera etapa el analista se involucra en la identificación de problemas, de las oportunidades y de los objetivos. Esta etapa requiere que el analista observe de forma objetiva lo que ocurre en una empresa. Las oportunidades son aquellas situaciones que el analista

considera que pueden perfeccionarse mediante el uso de los sistemas de información computarizados. En cuanto a la identificación de los objetivos, el analista deberá descubrir lo que la empresa intenta realizar, para determinar si el uso de los sistemas de información apoyaría a la empresa para alcanzar sus metas, el encaminarla a problemas u oportunidades específicas.

2) Determinación de Requerimientos: el analista trabaja directamente con los usuarios involucrados, para identificar los requerimientos de información que necesita, para ello pueden utilizarse algunas herramientas como: el muestreo, el estudio de datos, formas usadas por la empresa, la entrevista, los cuestionarios, la observación de la conducta de quién toma las decisiones, del ambiente, entre otros.

3) Análisis: consiste en analizar las necesidades propias del sistema. Con ayuda de herramientas y técnicas especiales facilitan esta etapa, las cuales son: diagramas de flujos de datos, diccionarios de datos, lenguaje estructurado, árboles o tablas de decisión, etc.
Durante esta fase, el analista prepara una propuesta del sistema que resume todo lo que ha encontrado, presenta un análisis costo/beneficio de las alternativas y plantea las recomendaciones de lo que deberá realizarse.

4) Diseño: en esta etapa el analista usa la información que recolectó con anterioridad y elabora el diseño lógico. Diseña procedimientos precisos de captura de datos, con el fin de que los datos que se introducen en el sistema sean correctos. Utiliza técnicas de diseño de formas y de pantallas. También se diseñan las pantallas para tener una interfaz adecuada a las necesidades del usuario y el diseño de los archivos o la base de datos.

5) Desarrollo y Documentación: el analista trabaja con los programadores para desarrollar todo el "software" original, transmitiéndoles los requerimientos de programación. El analista también colabora con los usuarios para desarrollar la documentación indispensable del sistema, incluyendo los manuales de procedimientos. La documentación le dirá al usuario cómo operar el sistema y qué hacer en caso de presentarse algún problema.

6) Prueba y Mantenimiento: el sistema debe probarse antes de utilizarse. El programador realiza algunas pruebas por su cuenta, otras se llevan a cabo junto con el analista de sistemas. El mantenimiento del sistema empieza en esta etapa, ya que en ocasiones es necesario realizar algunas operaciones especiales para que el sistema opere satisfactoriamente, un ejemplo de esto sería el depurar archivos cada "x" periodo de tiempo para evitar se llenen de información innecesaria.

7) Implantación y Evaluación: el analista ayuda a implantar el sistema de información, ofreciendo capacitación y entrenamiento a usuarios finales. La evaluación la dictamina el usuario final al utilizar el sistema ya terminado.

3.4.2 Estructurado

Un cambio radical surgió a raíz de que en el método tradicional el diseñador sólo tenía contacto con el analista, el cual le describía las especificaciones y no tenían contacto directo con los usuarios. En este cambio surge el método estructurado, se crea un tipo totalmente distinto de ciclo de vida del proyecto. Este método lo forman nueve pasos, los cuales son los siguientes: [YOURDON,1993]

1) Encuesta
2) Análisis
3) Diseño
4) Implantación
5) Generación de Pruebas de Aceptación
6) Control de Calidad
7) Descripción de Procedimientos
8) Conversión de Bases de Datos
9) Instalación

La figura 9 muestra una diagramación de estos pasos:

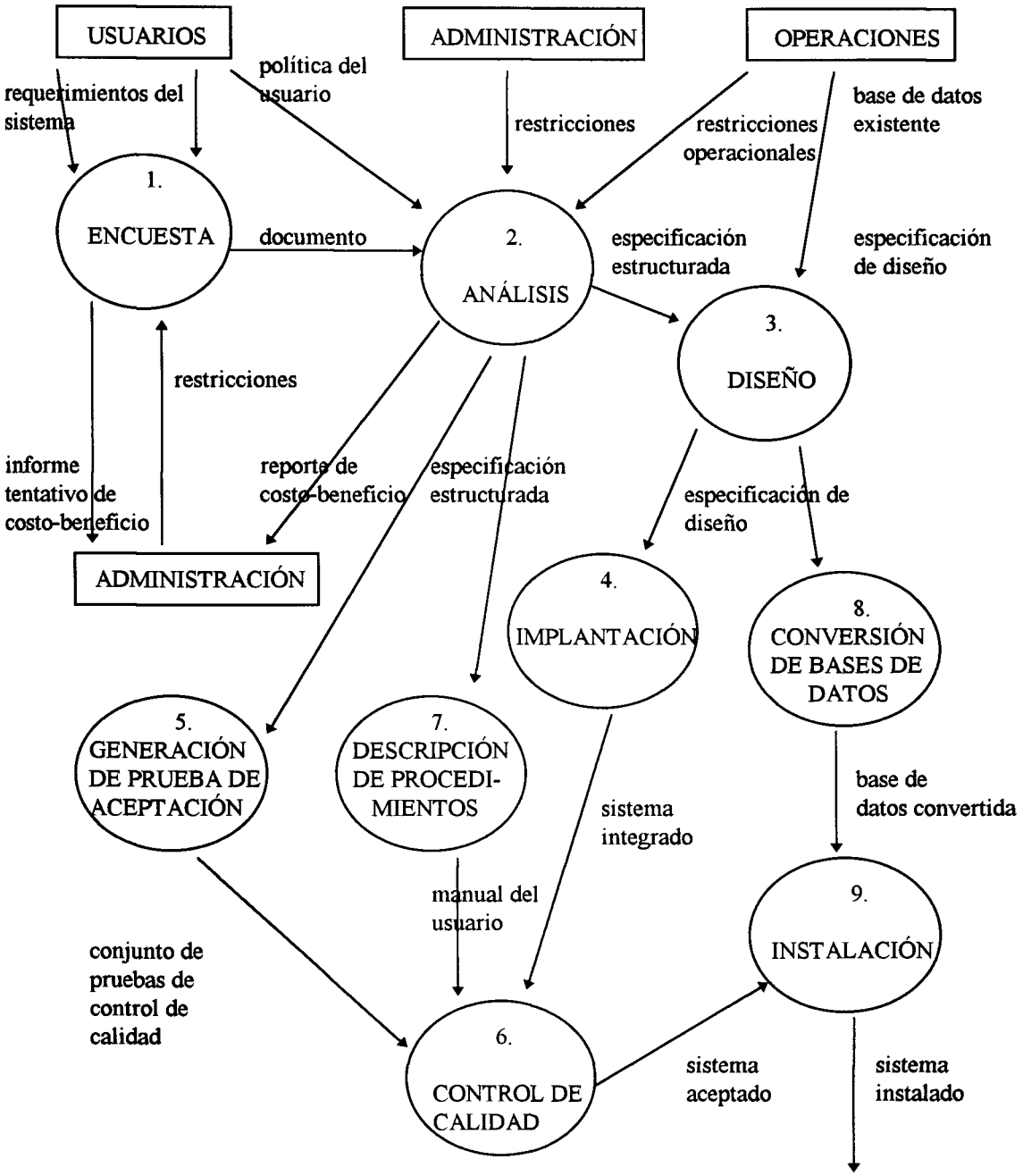


Figura 9. El Ciclo de Vida del Método Estructurado

1) Encuesta: es el estudio inicial, empieza cuando el usuario solicita que una o más partes de su sistema se automaticen. Los principales objetivos de la encuesta son:

- *Identificar a los usuarios responsables y crear un "campo de actividad" inicial del sistema.* Esto puede comprender la conducción de una serie de entrevistas para determinar qué usuarios estarán comprendidos en el proyecto propuesto.
- *Identificar las deficiencias actuales en el ambiente del usuario.* Esto en general comprenderá la lista de funciones que hacen falta o que se están llevando a cabo insatisfactoriamente en el sistema actual.
- *Establecer metas y objetivos para un sistema nuevo.* Esto puede ser también una simple lista narrativa que contenga las funciones existentes que deben reimplantarse, las nuevas que necesitan añadirse y los criterios de desempeño del nuevo sistema.
- *Determinar si es factible automatizar el sistema y de ser así, sugerir escenarios aceptables.* Esto implicará algunas estimaciones bastante rudimentarias y aproximadas del costo y el tiempo necesarios para construir un sistema nuevo y los beneficios que se derivarán de ello.
- *Preparar el esquema que se usará para guiar el resto del proyecto.* Este esquema incluirá toda la información que se lista anteriormente, además de identificar al administrador responsable del proyecto. También pudiera describir los detalles del ciclo de vida que seguirá el resto del proyecto.

2) Análisis: el propósito principal de esta etapa es el transformar sus dos entradas principales, las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujos de datos, diagramas entidad-relación, diagramas de transición de estado, entre otras.
   Esta etapa de análisis la forman 3 modelos: esencial, ambiental, comportamiento.

   - Modelo Esencial: es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario, diciendo lo mínimo posible acerca de cómo se implantará. La figura 10 muestra un modelo esencial más apropiado de lo que la función el sistema debe realizar sin importar su implantación final.

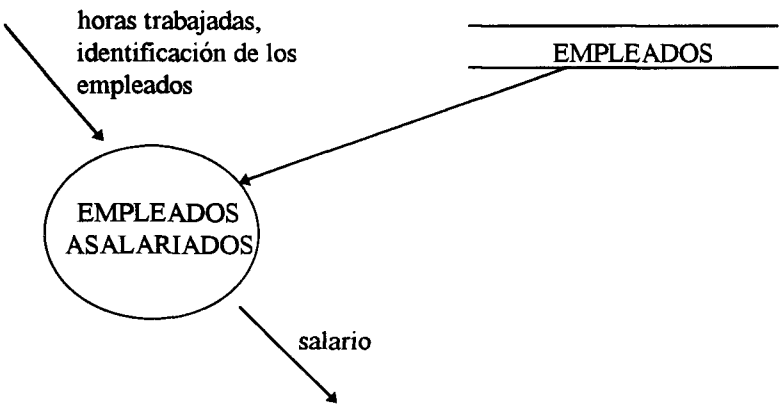


Figura 10. Un modelo de cuál es la función del sistema

- Modelo Ambiental: define la frontera entre el sistema y el resto del mundo (el ambiente en el cual existe el sistema). El usuario tiene una buena idea de la frontera general entre el sistema y el ambiente. Pero siempre existe un "área gris" que está abierta a negociaciones, las cuales pueden ser: *el usuario no está seguro, el usuario no había pensado, el usuario tenía algunas ideas preconcebidas que está dispuesto a reflexionar,* entre otras.
  La figura 11 muestra este tipo de modelo y el "área gris".

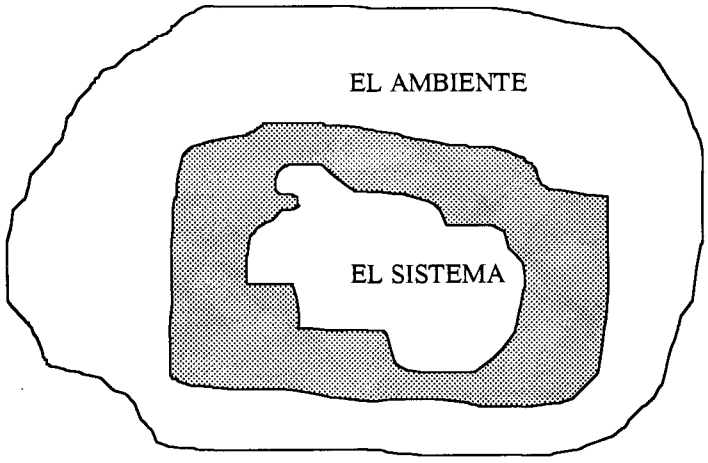


Figura 11. El "área gris" entre el sistema y el ambiente

El modelo del ambiente consta de tres componentes: declaración de propósitos, diagrama de contexto y lista de acontecimientos.

√ Declaración de Propósitos: es una declaración textual breve del propósito del sistema, dirigida al nivel administrativo superior, la administración de los usuarios, y otros que no están directamente involucrados con el desarrollo del sistema. El siguiente párrafo es un ejemplo de esto:

"El propósito del Sistema de Procesamiento de Libros Ajax es manejar todos los detalles de los pedidos de libros de los clientes, además del envío, facturación y cobro retroactivo a clientes con facturas vencidas. La información acerca de los pedidos de libros debe estar disponible para otros sistemas, tales como mercadeo, ventas y contabilidad."

√ Diagrama de Contexto: es un caso especial del diagrama de flujo de datos, en donde una sola burbuja representa todo el sistema. La figura 12 muestra un diagrama parcial de contexto.

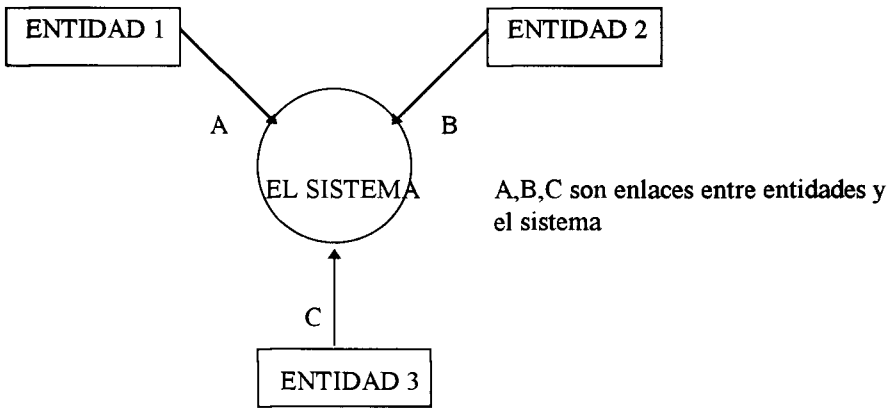


Figura 12. Diagrama parcial de Contexto

√ Lista de Acontecimientos: es una lista narrativa de los estímulos que ocurren en el mundo exterior a los cuales el sistema debe responder. En la figura 13 se muestra una lista de acontecimientos para un sistema de pedido de libros:

1. Un cliente hace un pedido
2. Un cliente cancela un pedido
3. La administración pide un reporte de ventas
4. Llega un pedido de reimpresión de un libro a la bodega

Figura 13. Lista de acontecimientos

- Modelo de Comportamiento: describe el comportamiento que del sistema se requiere para que interactúe de manera exitosa con el ambiente. Lo forman principalmente: la construcción de los DFD (Diagrama de Flujo de Datos), DER (Diagrama Entidad-Relación) y el Diccionario de Datos.

3) Diseño: esta actividad se dedica a asignar porciones de la especificación a procesadores adecuados y a labores apropiadas dentro de cada procesador. Dentro de cada labor, la actividad de diseño se dedica a la creación de una jerarquía apropiada de módulos de programas y de interfaces entre ellos para implantar la especificación creada en la actividad 2. Además, esta actividad se ocupa de la transformación de modelos de datos de entidad-relación en un diseño de base de datos.
En esta fase se involucra el diagrama de estructura, que muestra la organización jerárquica de módulos dentro de una tarea. La figura 14 muestra los principales componentes de este diagrama.

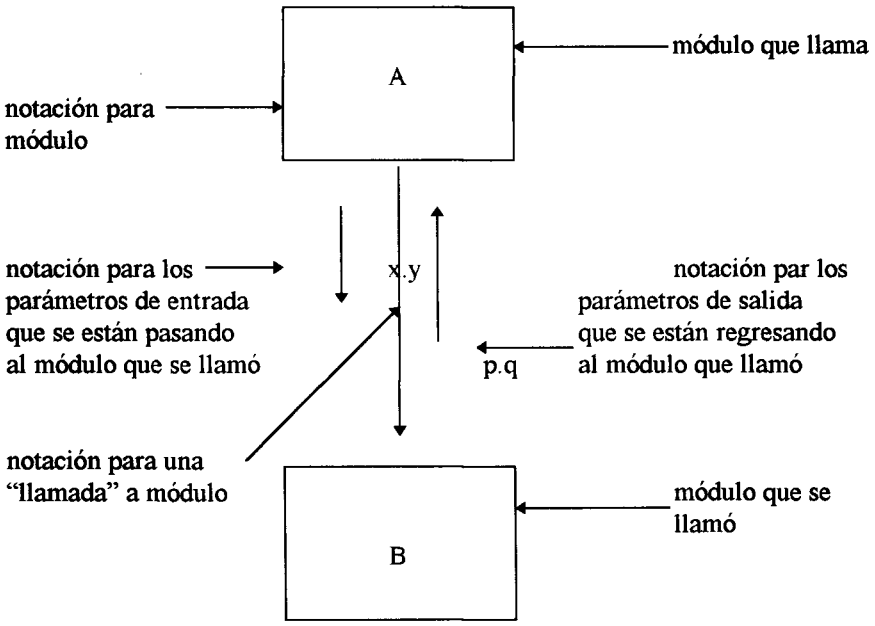


Figura 14. Componentes de un diagrama de estructura

4) Implantación: esta actividad incluye la codificación y la integración de módulos en un esqueleto progresivamente más completo del sistema final. Por eso, esta actividad incluye tanto programación estructurada como implantación descendente.

5) Generación de Pruebas de Aceptación: la especificación estructurada debe tener toda la información necesaria para definir un sistema que sea aceptable desde el punto de vista del usuario. Por eso, una vez generada la especificación, puede comenzar la actividad de producir un conjunto de casos de prueba de aceptación desde la especificación estructurada.

6) Control de Calidad: es la prueba final o la prueba de aceptación. Esta actividad requiere como entradas los datos de la prueba de aceptación generada en la actividad 5 y el sistema integrado producido en la actividad 4.

7) Descripción de Procedimientos: esta actividad involucra la creación de un manual del usuario, en donde se describe formalmente las partes del sistema, así como la descripción del cómo interactuarán los usuarios con la parte automatizada del nuevo sistema.

8) Conversión de Bases de Datos: en algunos proyectos es necesario convertir bases de datos al nuevo sistema, si es así, se requiere como entrada la base de datos actual del usuario, al igual que la especificación del diseño producida por medio de la actividad 3.

9) Instalación: es la actividad final; sus entradas son el manual del usuario producido en la actividad 7, la base de datos convertida que se creó en la actividad 8 y el sistema aceptado y producido por la actividad 6. En esta fase se involucra la capacitación y el entrenamiento a usuarios.

3.4.3 LEAD

LEAD es un estándar para "Logical Entity Action Design". El modelo es un conjunto de patrones que relacionan elementos del modelo del negocio con elementos de la interface a usuarios. El término "modelo del negocio" con el modelo de datos (entidades lógicas) y las reglas del negocio (acciones) para la aplicación. [CLIENT, 1994]

El modelo LEAD incluye una estructura de administración de proyectos que integra características de metodologías como Análisis Estructurado, Metodología de Ingeniería de Información, "Joint Application Design" (JAD) y Desarrollo de Aplicaciones Rápidas (RAD, "Rapid Application Development").

El proyecto LEAD se desglosa en nueve fases:

1) Colección de Requerimientos
2) Análisis Preliminar
3) Sesión JAD
4) Prototipeo
5) Especificación
6) Planeación
7) Desarrollo
8) Trayectoria (Seguimiento)
9) Mantenimiento

1) Colección de Requerimientos: en esta etapa debe conducirse con cuidado y sólo a un alto nivel. Esta se utiliza sólo para el análisis preliminar y no debe ser altamente detallado. Contiene tres aspectos: Administración, Documentos Fuente y Entrevistas.
   La Administración se enfoca a entrevistas administrativas, para conocer el alcance del proyecto, y determinar entidades externas.
   Los Documentos Fuente, son formas y reportes que se utilicen en la organización y que sean relevantes e importantes para el desarrollo de la aplicación.
   Las Entrevistas, deberán hacerse a las personas que realizan los documentos fuente, de tal forma que se conozca el cómo y el para qué sirve la información.

2) Análisis Preliminar: se utiliza una herramienta CASE como "Popkin's System Architect" para documentar el sistema. Generando un grupo de diagramas para representarlo. Los diagramas incluyen:

- Diagramas ER, entidades en la base de datos y sus relaciones con otras
- DFD, flujos de datos
- ESTD, determina el ciclo de vida de las entidades llave
- Diagrama de Contexto, visualiza el alcance del proyecto

Los diagramas no deberán hacerse tan detallados, sólo basta con que sean comprensibles y entendibles.

- Diagramas ER (Entidad Relación): se utilizan los documentos fuente para crear un diagrama preliminar ER. Estos identifican todos los atributos usados en la organización y se pueden interpretar para determinar sus relaciones. Se identifica el dominio de cada atributo y las llaves primarias o extranjeras. Las llaven pueden ser determinadas desde el contexto del negocio, considerando cómo los empleados identifican objetos de otros.
  El grupo de entidades dentro de los objetos lógicos de alto nivel, crean diagramas de Relaciones de Objetos (OR) para identificarlos y mostrar sus relaciones. Esto se utiliza durante el JAD para introducir a los usuarios a los conceptos de Diagramas ER sin empezar con un documento abrumadoramente detallado como el diagrama ER a nivel atributos.
  Al final de esta fase, se tienen tres documentos:

  * Diagrama OR, objetos de alto nivel y sus relaciones.
  * Nivel Entidad ERD, entidades y sus relaciones.
  * Nivel atributo ERD, atributos y dominios.

- DFD, Diagrama de Flujo de Datos: el crear DFD's es para mostrar el flujo de los documentos fuente. Relacionando los flujos de datos de cada documento fuente. Son usados como herramienta de comunicación para validar los requerimientos de la organización.

- Diagrama de Entidad de Estado de Transición (ESTD, "Entity State Transition Diagram"): identifica las entidades llaves que tienen un ciclo de vida. Identificando cada uno de los estados en su ciclo de vida (ej. nuevo, precio, confirmado, etc.). Determina los eventos que causan los cambios de estado para la entidad.

- Diagrama de Contexto: se crea en base a las entrevistas administrativas, el ESTD y los DFD's.

3) Sesión JAD ("Joint Application Design"): Esta fase no sólo es importante para completar las especificaciones, sino para terminar todo el proyecto. Primero, se requiere de un comité para que atienda las sesiones JAD. Segundo, se les da a conocer a los desarrolladores los requerimientos de los usuarios. Tercero, se establece una relación de trabajo entre usuarios y desarrolladores; se establecen canales de comunicación no tan formales durante el transcurso del proyecto.

   Existen dos fases:

   * Fase I - Validación y Mejoramiento de Diagramas, el propósito de esta fase es el validar y mejorar la documentación creada en la fase de análisis. Empezar con el Diagrama de Contexto para mostrar el alcance del proyecto, enseguida con los ERD's, empezando con el alto nivel ORD. Antes de mostrar el Diagrama ER a nivel Atributos, revise los conceptos de dominio con los usuarios para que ellos validen los cambios. Después, checar los DFD's, y ESTD.

   * Fase II - Demostración del Prototipo, se construye un prototipo en base al Diagrama ER y ESTD's de la sesión JAD.

4) Prototipeo: usando la librería LEAD, se genera la base de datos, el "script" de prueba de datos y el prototipo de la aplicación en SQLWindows. Regrese a la sesión JAD para revisar el prototipo. Después de obtener retroalimentación, realice modificaciones a los documentos y después al prototipo.

   Las herramientas a utilizar son:

* Exportar los documentos CASE, las herramientas CASE tienen la capacidad de exportar documentos a archivos. La librería LEAD lee los archivos exportados para generar la base de datos, prueba de datos y el prototipo de la aplicación.

* El generador de datos, crea una base de datos vacía y nueva.

* El generador de prototipos, una vez hecha la base de datos se utiliza esta herramienta para generar el prototipo.

5) Especificación: una vez terminado el prototipo, las especificaciones de los detalles del proyectos son requeridos. Este trabajo puede ocurrir en tres áreas:

* Aumento en las formas, consiste en determinar cambios a la forma del plan y agregar procesamientos especializados a la forma.

  - Forma del Plan y Campos Dependientes.- identificar cualquier campo adicional que requiera la forma. Esto puede ser algún campo de otras entidades que sea sólo de lectura para propósitos de información.

  - Procesamientos especializados.- existen tres funciones utilizadas para manejar los procesamientos especiales requeridos para las formas:

    InitNew()- se ejecuta cuando una nueva forma es creada. Se utiliza para usar valores de "default".
    InitOpen()- se ejecuta cuando una forma es abierta. Puede usarse para campos dependientes, para habilitar o deshabilitar menús o controles de edición o algo parecido.
    Validate()- se ejecuta antes de que una forma sea grabada, las reglas del negocio puede ser que se hagan pruebas antes de grabar las modificaciones.

* Acciones del negocio, cada acción del negocio es identificada en los ESTD's. La magnitud del desarrollo está en la codificación de las acciones del negocio. Esta codificación está estructurada como sigue:

- Pre-condiciones, determinan las condiciones requeridas para permitir ejecutar las acciones del negocio.

- "User Input", determinan las entradas requeridas por el usuario, así como el método para obtener información.

- Post-condiciones, determinan las acciones requeridas para completar la acción del negocio. Están a alto nivel, se encuentran en los DFD's.

* Reportes, necesitan ser revisados y especificados. Hay tres factores:

- "User Input", parámetros para el reporte.
- Entidades fuente, identifican entidades y atributos que pueden aparecer
- Procesamiento especializado, identifican campos calculados para el reporte.

6) Planeación: hasta que el proyecto esté bien especificado éste deberá planearse, los siguientes elementos deben considerarse:

* Fase del Proyecto, partir el proyecto en pequeñas fases a completar. De esta forma los usuarios visualizan mejor todo el sistema y se visualizan rápidamente los problemas.

* Experiencia del Staff, en aplicaciones Cliente/Servidor tiene un gran significado en la planeación del proyecto. Si el tiempo está distribuido para aprender y realizar errores, un proyecto puede realizarse con personal con menos experiencia.

* "Software" métrico, este modelo incluye un conjunto de "software" métrico. Medidas cuantitativas son usadas para determinar requerimientos de recursos para el proyecto, lo cual incluye:

- número de entidades y relaciones
- complejidad de acciones del negocio y reportes
- experiencia del Staff

Las métricas están basadas en investigaciones de "software" métrico y la experiencia en construir aplicaciones LEAD.

7) Desarrollo: el proyecto puede desglosarse en partes, de tipos de tareas correspondientes al paso 5 (Especificación).

8) Trayectoria (seguimiento): la trayectoria del proyecto es importante para identificar problemas en cada área.

9) Mantenimiento: a modificaciones y cambios de la aplicación, en forma fácil y sencilla ya que sólo se debe modificar en la parte correspondiente del diseño de LEAD, editando o agregando nuevas acciones al sistema.

En resumen, se debe considerar en forma importante al: "software", "hardware" y soporte y servicio, al desarrollar aplicaciones Cliente/Servidor.

Algunos aspectos importantes del "software" son los costos del personal para mantenerlos y capacitarlos; el emigrar a otras plataformas para mejorar la productividad; el crear estándares, y considerar que los usuarios finales cada vez desarrollan sus propias aplicaciones, por lo que se debe considerar el adquirir "software" de acuerdo a sus necesidades.

También, se deben considerar los requerimientos de "hardware" para el servidor como para la estación cliente, esto de acuerdo a las necesidades de la organización, la plataforma de desarrollo y el presupuesto disponible para lograrlo.

El soporte y servicio, aspecto relevante para lograr excelentes resultados en el desarrollo de aplicaciones Cliente/Servidor; ya que es necesario e indispensable el poder contar con personal capacitado, dispuesto a brindar ayuda en el momento que se requiera, así como la seguridad y el desempeño que se tenga con el "hardware"/"software".

Existen algunas Metodologías que ayudan a realizar sistemas computacionales, de las cuales podemos mencionar: tradicional, estructurado y el LEAD.

El tradicional, muestra en forma sencilla y simple el cómo llegar a realizar un sistema sin emplear muchas herramientas que profundicen la obtención de información.

El estructurado, muestra una serie de herramientas que ayudan a ampliar el panorama de construcción de un sistema dando lugar a evitar el que se omita cierta información relevante.

LEAD, es una metodología utilizada para aplicaciones bajo Cliente/Servidor en donde involucra combinaciones de las dos metodologías anteriores y el prototipeo, además de agregar herramientas para los conceptos de "front-end" y "back-end".

CAPÍTULO 4
SITUACIÓN ACTUAL EN MÉXICO

4.1 La informática en México

La historia de la informática en nuestro país, ha pasado por diferentes etapas, las cuales mencionaremos en forma global. [DIECK, 1994]

La computadora llega a México en la década de los 60's. Sus aplicaciones se limitan a la automatización de funciones operativas dentro de las empresas. El personal que formaba el área de informática eran técnicos en procesamiento de datos.

La era del proceso de datos en México comienza con la llegada de la computadora y se extiende hasta la primera parte de los 70's.

Las empresas comienzan a observar el potencial de la computadora y tratan de justificar su alto costo inicial, es así como se prolifera su uso. Los 70's se caracterizan por un alto incremento en el desarrollo de aplicaciones en su mayoría operacionales. En esta década, adquieren importancia las carreras profesionales del área de sistemas, por lo que la informática se pone de moda. En las empresas se hace muy común la existencia de grandes departamentos de sistemas.

La era de los sistemas de información gerenciales surge hasta casi finales de los 70's.

En 1980, llegan las primeras microcomputadoras al país. Esto fue un suceso importante, pues se considera un gran paso para la descentralización de las funciones de informática.

A fines de los 80's, aparecen las primeras redes locales, y se vuelven más comunes las aplicaciones gerenciales.

En la década de los 90's, las redes y las telecomunicaciones se vuelven un producto o servicio altamente demandado. Es notoria la aplicación de la

informática para crear otros servicios, esto se ve muy claro en las empresas cuyo giro comercial es el servicio, como por ejemplo: el sector bancario, el sector educacional, y el sector industrial, entre otros.

La era de innovación surge en 1990. Pero la era de integración y reestructuración de negocios también ya se está haciendo aparente. Probablemente esto se deba a la necesidad que tienen las empresas para seguir operando en un mercado global tan competido.

En los 90's, se marca con énfasis la Tecnología de Información (TI). La cual es la unión de computadoras, telecomunicaciones, estaciones de trabajo, dispositivos de almacenamiento masivo, y tecnologías de imágen. [KEEN, 1991]

Este énfasis, debido a que era necesario ver todas las herramientas de la informática como un todo no por partes como se había estado haciendo desde un principio.

La TI es un término que involucra el "hardware", las aplicaciones y los servicios usados por las organizaciones para proveer a sus miembros de: datos, información y conocimiento. [LUFTMAN, LEWIS, ALDACH, 1993]

Las fechas y etapas mencionadas, no ocurren de igual manera en todas las organizaciones. Existen algunas que brincan etapas, otras que extienden el tiempo en alguna de ellas.

Las organizaciones no deben de cerrarse a su propia evolución, deben de observar a su alrededor y comenzar a apreciar, si es que aún no lo han hecho, que la Tecnología de Información no sólo es un apoyo para la automatización de sus funciones operacionales, sino que es una valiosa herramienta para innovar y rediseñar, es un arma para competir.

4.2 Cultura informática, base de la competitividad

Para analizar lo que sucede en México en cuanto a cultura informática se refiere, podemos citar el sector financiero-bancario como punto de referencia.

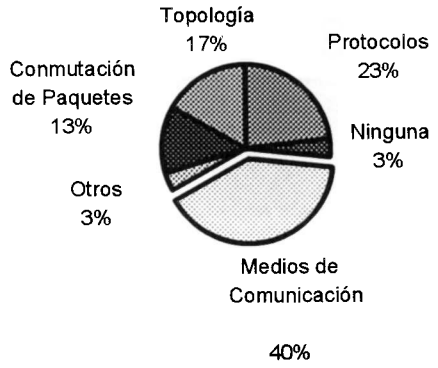
Con este caso en particular, nos daremos cuenta tanto de la cultura informática como de la competitividad que se genera.

Recientemente fue publicado un artículo de investigación en la revista RED titulado: "Soluciones tecnológicas a los retos de los bancos". En éste, se cuestiona a la tecnología como el único elemento de solución para la competitividad de los bancos. [RED, Num. 47,1994]

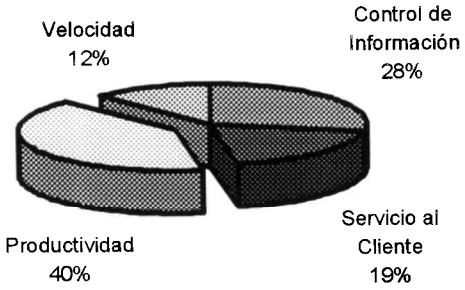
La revista citada seleccionó una muestra variada compuesta por bancos, casas de bolsa, instituciones de crédito y dependencias financieras; siendo éstas las siguientes: [RED, Num. 47,1994]

* Multibanco Mercantil Probursa
* Inverlat
* Bank of America
* Carnet
* Comisión Nacional de Valores
* Nacional Financiera
* American Express
* Casa de Bolsa Bursamex
* Banamex
* Banco Nacional de Comercio Exterior
* Bancrecer
* CBI
* Banobras
* Banca Cremi
* Banquers Trust Company
* Unión
* Serfín
* Casa de Bolsa Value
* City Bank
* Internacional
* Quadrum
* Bancomer

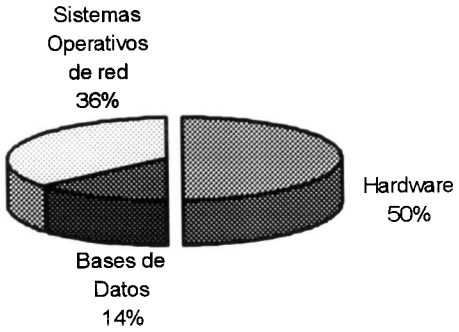
Con la colaboración de estas instituciones financieras, se realizó la investigación que concluyó con los resultados que a continuación se describen.



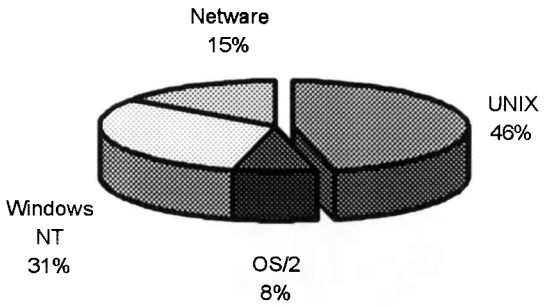
Gráfica 1. Nuevas Tecnologías en Telecomunicaciones



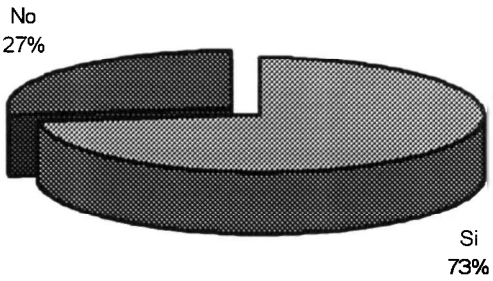
Gráfica 2. Mejoras que se buscan con las Nuevas Tecnologías



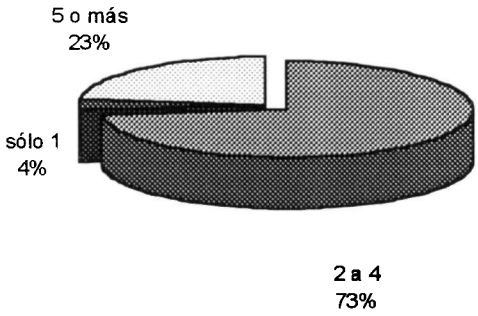
Gráfica 3. Nuevas Tecnologías en Informática



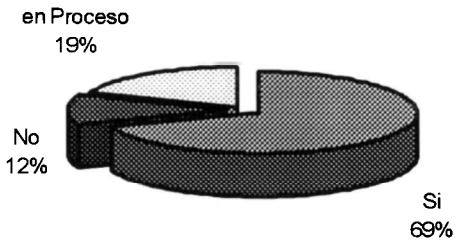
Gráfica 4. Sistemas Operativos que se están instalando



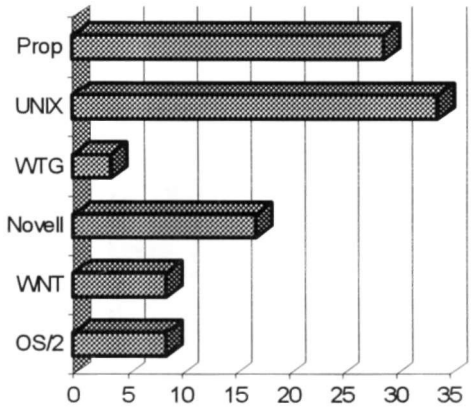
Gráfica 5. Implantación de Tecnologías Front-End para atender al Cliente



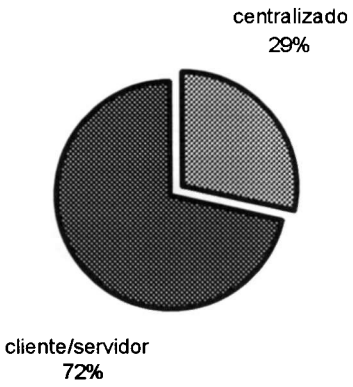
Gráfica 6. Número de Sistemas Operativos utilizados



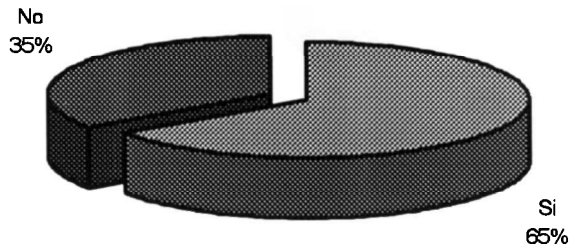
Gráfica 7. Interoperabilidad bajo Sistemas Abiertos



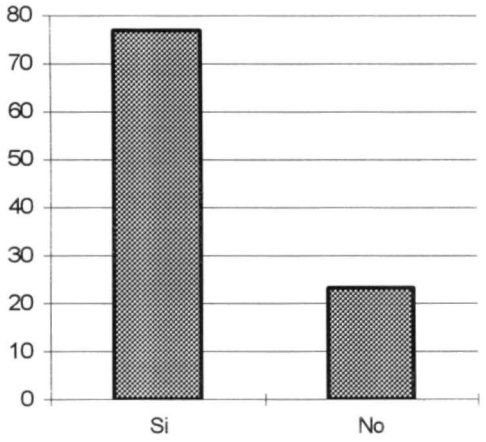
Gráfica 8. Sistemas Operativos en funcionamiento



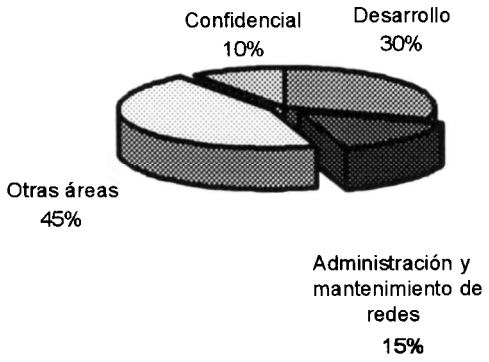
Gráfica 9. Arquitectura dominante en el Sector Financiero



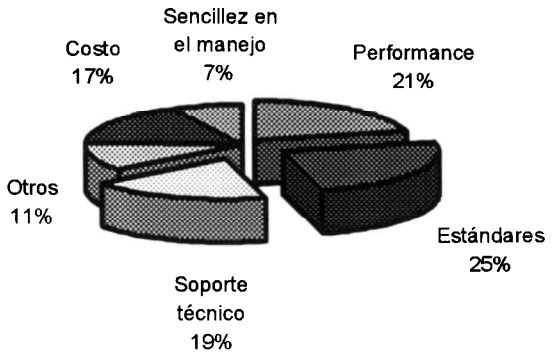
Gráfica 10. "Mainframe" tiene Futuro



Gráfica 11. "Outsourcing" es una Alternativa



Gráfica 12. "Outsourcing" en diversas áreas



Gráfica 13. Elementos Clave en la Decisión de Compra

Los resultados arrojados por estas encuestas, presentan directamente las siguientes directivas y tendencias:

- √ Incremento en el uso de servicios de telecomunicaciones
- √ Uso de tecnologías de conmutación para contar con mayor capacidad de transmisión
- √ Crecimiento de la arquitectura Cliente/Servidor
- √ Predominio de UNIX y sistemas abiertos
- √ Uso del "mainframe" como un superservidor
- √ Procesamiento centralizado de la información
- √ Incremento de soluciones "front-end" ("software" y "hardware")

CAPÍTULO 5
ANÁLISIS DE LA SITUACIÓN ACTUAL EN TAMPICO CON CLIENTE/SERVIDOR

5.1 Sondeo de la situación actual

Se realizó un sondeo en la zona de Tampico Tamaulipas, en diez empresas que están desarrollando "software" bajo herramientas Cliente/Servidor. Algunas empresas se dedican a la venta de aplicaciones siendo su giro principal el dar asesoría y desarrollo de sistemas, y a la vez dan lugar a que sus empresas cliente sigan con sus propios desarrollos; otras, tienen giro industrial, educativo o comercial, y desde un inicio han realizado sus propias aplicaciones.

Las empresas encuestadas para realizar el sondeo son las siguientes:

1) Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Tampico
2) Consultores en Informática Empresarial, S.A. (CIE)
3) Informática y Control S.A. de C.V.
4) Diseño Profesional de Sistemas (DIPROS)
5) Sociedad Industrial S.A. (Grupo Continental)
6) Autotransportes Mante S.A. de C.V.
7) NHUMO S.A. de C.V. (Negro de Humo)
8) Industrias NEGROMEX S.A.
9) Miramar Technology S.A.
10) Universal Computers S.A.

El objetivo principal de la elaboración de la encuesta, es conocer el tipo de herramientas bajo la arquitectura Cliente/Servidor que utilizan las empresas en Tampico; así como el porqué trabajar bajo Cliente/Servidor y el tipo de metodología que aplican para el desarrollo de sus sistemas.

El diseño de la encuesta aplicada se encuentra en el Anexo 2.

La tabla 1 explica las once secciones en que se divide la encuesta.

Sección	Descripción	Definición
I	Información sobre la Empresa	conocer el puesto que tiene la persona a encuestar y el número de empleados que tiene
II	"Software" de Desarrollo	conocer el "software" con el que ha trabajado para desarrollo de sus aplicaciones; el que está orientado a cliente/servidor; las razones que hicieron posible el cambio de herramienta y las personas que tomaron la decisión del cambio
III	Proveedores	saber qué proveedores existen y sus criterios de evaluación
IV	"Hardware"	conocer el tipo de "hardware" que utilizan para estaciones cliente como para servidores
V	Bases de Datos y Plataformas	saber qué tipos de manejadores de bases de datos utilizan y las plataformas
VI	Requerimientos y Opiniones	conocer qué más requieren del "software" y las opiniones que tienen sobre el mismo
VII	Usuarios Finales	saber si tienen habilidades computacionales, la frecuencia de uso de ciertas herramientas y el cómo llegaron a seleccionar el "software" orientado a usuario final
VIII	Capacitación	conocer si fue difícil dar capacitación al personal de sistemas y al usuario final, indicando los motivos, así como el tiempo que se llevó dar dicha capacitación
IX	Ventajas/Desventajas	indicar ventajas y desventajas sobre sus herramientas de cliente/servidor
X	Metodologías	conocer las metodologías que utilizan para desarrollar sus aplicaciones cliente/servidor
XI	Comentarios	conocer en general qué tanto conocimiento tiene la persona encuestada (de sistemas) sobre ciertos tópicos

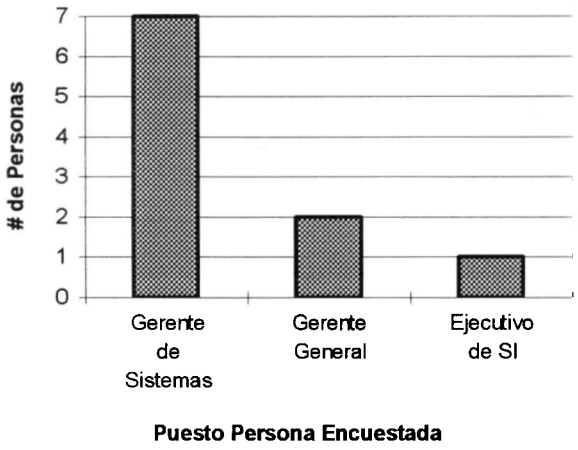
Tabla 1. Secciones de la Encuesta realizada a Desarrolladores de Aplicaciones bajo Cliente/Servidor

Cabe mencionar que la mayoría de las preguntas hacen referencia a más de una respuesta, por lo que para ello se indican el número de empresas que las seleccionaron.. En otras sí fue posible el indicar porcentajes, de tal forma que al graficar esto da un total del 100%.

5.2 Metodología y Análisis

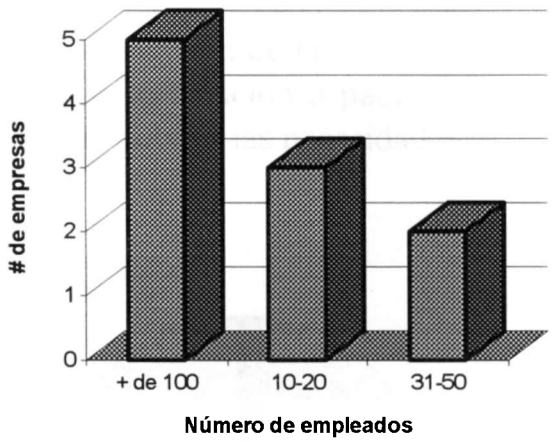
El análisis de la encuesta se hará por cada una de las secciones, presentando en forma gráfica o textual el resultado de cada pregunta correspondiente a la sección.

Sección I. Información sobre la empresa.



Gráfica 14. Puesto que ocupa actualmente

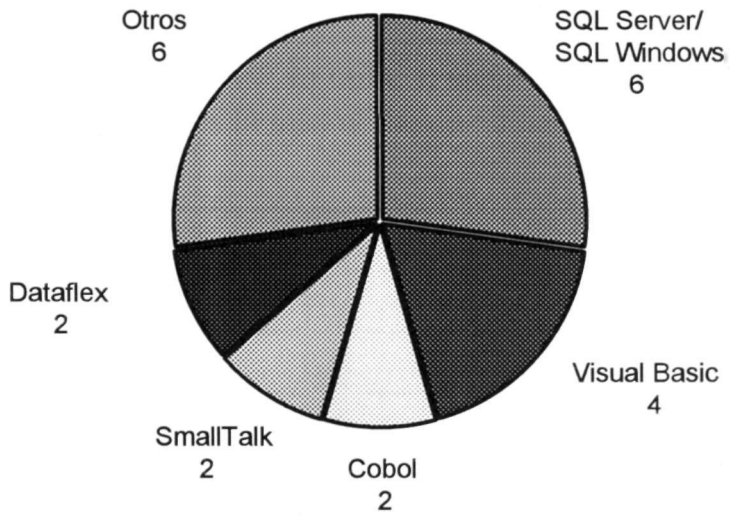
En esta gráfica se visualiza que de las diez personas de empresas encuestadas, siete son gerentes de sistemas, siendo su actividad principal el ser responsables de desarrollo de aplicaciones. Esto no significa que el giro de los gerentes generales sea diferente ya que en este caso muy en particular son los dueños de las empresas y tienen más de una responsabilidad. El ejecutivo de sistemas de información es responsable de desarrollo de aplicaciones y el brindar asesoría y capacitaciones al personal de la empresa.



Gráfica 15. Número de Empleados que tiene la empresa actualmente

La gráfica resume que de las diez empresas, cinco de ellas tienen más de 100 empleados laborando actualmente. Lo cual indica que son empresas grandes las que están utilizando herramientas Cliente/Servidor.

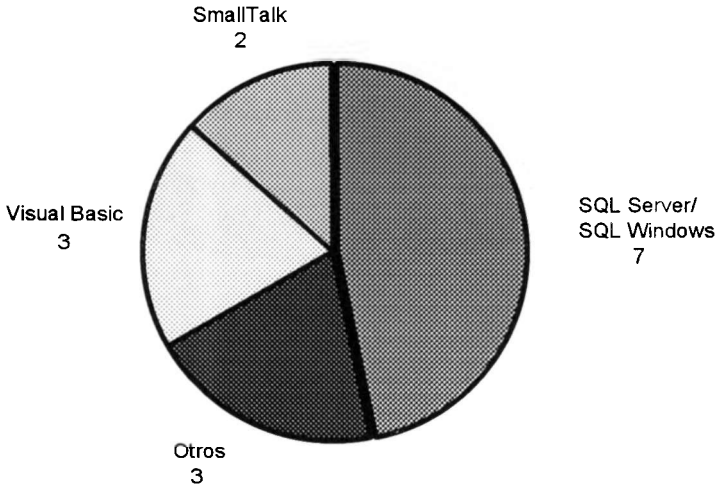
Sección II. "Software" de Desarrollo.



Gráfica 16. "Software" de desarrollo que han utilizado para las aplicaciones

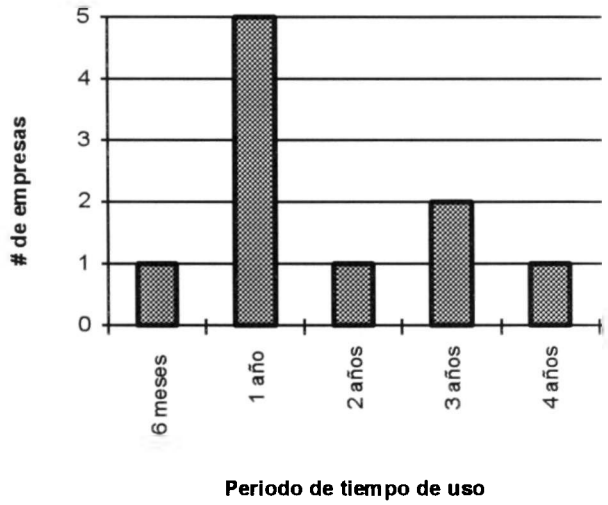
Esta gráfica determina el número o cantidad de empresas que utilizaron algún "software" en especial. Por lo tanto, el número mayor de "software" utilizado para realizar aplicaciones cae en "SQL Server/SQL Windows" y en "otros", sin embargo el más representativo es el primero ya que seis empresas

de las diez lo han utilizado. En el caso de "otros", el número es menos significativo debido a que la mayoría de las empresas siempre utilizan, no uno sino diversos lenguajes de programación o paquetes manejadores de archivos para indagar qué herramienta cubrirá las necesidades de la empresa.



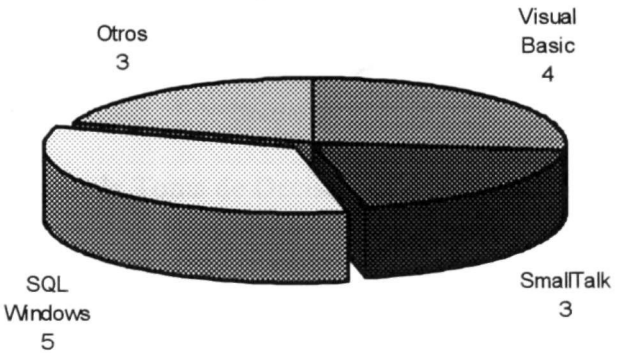
Gráfica 17. "Software" de desarrollo bajo la tecnología Cliente/Servidor utilizado para las aplicaciones

"SQL Server/SQL Windows" lo utilizan siete de las diez empresas, lo cual indica que esta herramienta es la más utilizada actualmente en Tampico.



Gráfica 18. Tiempo de uso del "software" bajo la tecnología Cliente/Servidor

El periodo mayor de tiempo de uso del "software" es de un año, ya que cinco empresas caen en este periodo de tiempo. Es importante indicar que en Tampico pocas empresas tienen más tiempo de utilización de herramientas Cliente/Servidor.

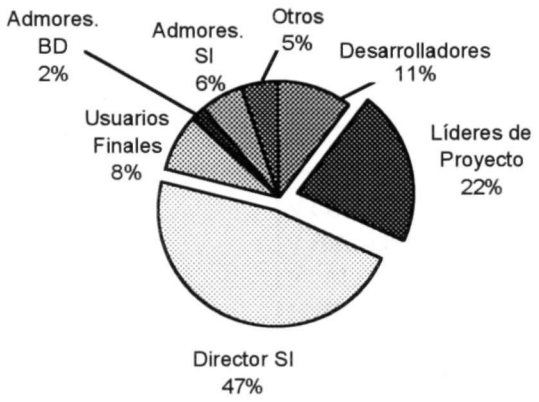


Gráfica 19. Indique las herramientas con las que empezó a desarrollar "software" para aprender la tecnología Cliente/Servidor

"SQL Windows" ha sido la herramienta que más han utilizado las empresas para el inicio de sus desarrollos, esto representa cinco de las diez empresas. Sin embargo es importante notar que del sondeo realizado representa sólo la mitad de las empresas encuestadas.

Existen varias razones por las cuales las empresas decidieron utilizar la herramienta bajo la tecnología Cliente/Servidor. A continuación se mencionarán indicando la prioridad en forma ascendente, siendo el número 1 la más alta prioridad y el número 8 la más baja.

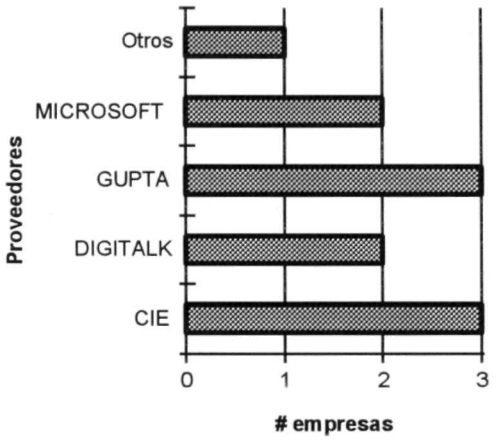
1) Alta Productividad de los desarrolladores
2) Vanguardia Tecnológica
3) Soporte total en Múltiples Bases de Datos
4) Soporte Total en Windows
5) Facilidades para programar en equipo
6) Capacidad de Prototipeo Rápida
7) Ahorro en Costos
8) Interfaces Orientadas a Objetos



Gráfica 20. Porcentajes de la Toma de Decisiones para la adquisición de la Herramienta Cliente/Servidor

De las diez empresas encuestadas, el Director de Sistemas de Información es el que tomó la decisión para la adquisición de la herramienta, esto puede indicar que tal vez tenga la misma toma de decisiones para adquirir cualquier otro tipo de "software".

Sección III. Proveedores



Gráfica 21. Proveedores para el desarrollo de las aplicaciones

"CIE" (Consultores en Informática Empresarial) representa ser el proveedor que más empresas cliente tiene en Tampico. En la parte correspondiente a "otros", significa que algunas empresas han desarrollado sus propias aplicaciones.

La tabla 2 muestra algunos criterios de evaluación que tuvieron los encuestados sobre los proveedores.

PROVEEDORES	OPINIONES
CIE	√ Conocimiento del "Software" de GUPTA √ Habilidad para desarrollo de sistemas √ Al adquirir las aplicaciones, se adquieren la propiedad de los códigos fuentes √ Capacidad en el uso de la tecnología Cliente/Servidor √ Experiencia en sistemas √ Tienen bases fuertes sobre aplicaciones Cliente/Servidor, tales como: "hardware", "software", personal capacitado, etc. √ Soporte Técnico
DIGITALK	√ Gran Versatilidad √ Requerimientos de "hardware" aceptables
GUPTA	√ Liderazgo a nivel mundial √ Cumple con las expectativas de las necesidades de los clientes

Tabla 2. Criterios de evaluación para la selección de los proveedores

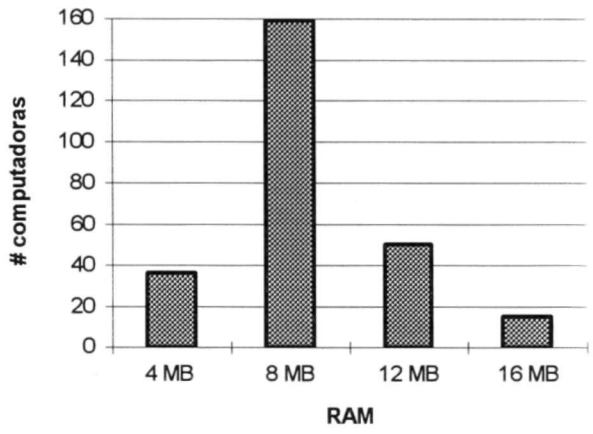
Sección IV. "Hardware"

La tabla 3, muestra la cantidad de equipos que se tienen invertidos como estaciones Cliente para desarrollar y utilizar aplicaciones Cliente/Servidor .

PROCESADORES	4 MB RAM	8 MB RAM	12 MB RAM	16 MB RAM	TOTAL
286		26			26
386		31			31
486, 25 MHZ		28	10		38
486, 33 MHZ	13	3	13		29
486, 50 MHZ	13	52	10	5	80
486, 66 MHZ	10	9	7	10	36
PENTIUM		10	10		20
TOTAL	36	159	50	15	260

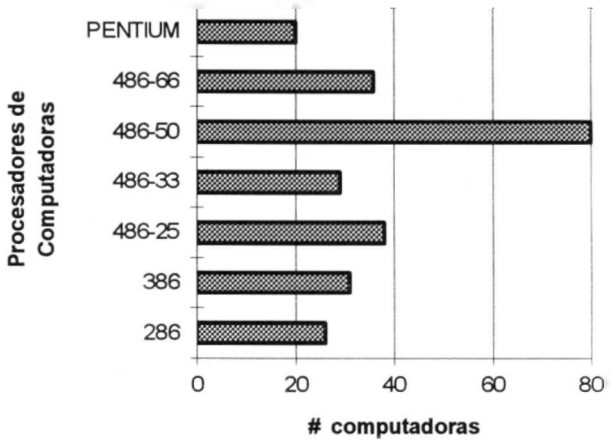
Tabla 3. Relación de estaciones "Cliente" para desarrollo y uso de aplicaciones Cliente/Servidor.

Las gráficas 22 y 23, muestran la información detallada de la tabla anterior en forma concisa.



Gráfica 22. Cantidad de computadoras "Cliente" en base a su RAM

En esta gráfica se puede visualizar que se tienen más equipos con una memoria base de 8 MB de RAM.



Gráfica 23. Cantidad de computadoras "cliente" en base a su procesador

Esta gráfica indica que el mayor número de equipos es con el procesador 80486 de 50 MHZ.

En la tabla 4, se mostrará la información correspondiente a los tipos de servidores que manejan las empresas. En este caso en particular, el tipo de servidor varía de acuerdo a las necesidades de cada empresa y debido a que

son pocas las empresas en Tampico utilizando Cliente/Servidor, existe diversidad en los equipos.

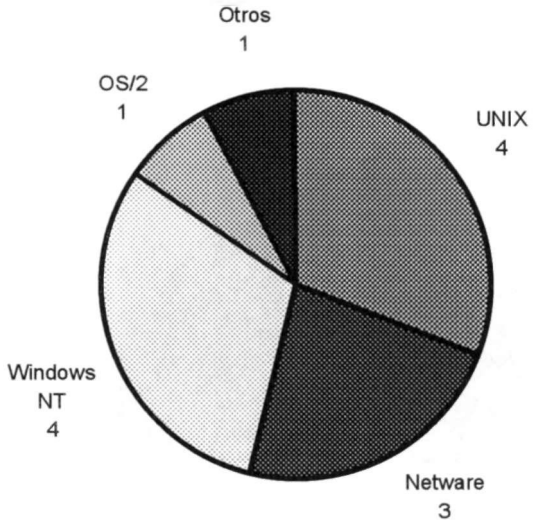
SERVIDOR	RAM	DISCO DURO	PROCESADOR
HP-NET SERVER	• 48 MB √ 32 MB ⇒ (2) 32 MB ♦ (2) 16 MB	• 2 GIGABYTES √ 2 GIGABYTES ⇒ 1 GIGABYTE ♦ 1 GIGABYTE	• 486 DX2/66 MHZ √ PENTIUM ⇒ 4d/ 66 MHZ LM ♦ 5p/ 66 MHZ LC
ACER ALTOS	64 MB	2 GIGABYTES	PENTIUM
PENTIUM	16 MB	1 GIGABYTE	INTEL
IBM RS/6000-250	32 MB	1 GIGABYTE	
GATEWAY	8 MB	0.5 GIGABYTE	486 DX/66 MHZ
ACERFRAME	32 MB	1 GIGABYTE	500
ACER	8 MB	1 GIGABYTE	486 DX/33 MHZ
RS-25 C	(3) 32 MB	300 MB	

Tabla 4. Tipos de Servidores usándose actualmente para herramientas Cliente/Servidor

Sección V. Bases de Datos y Plataformas

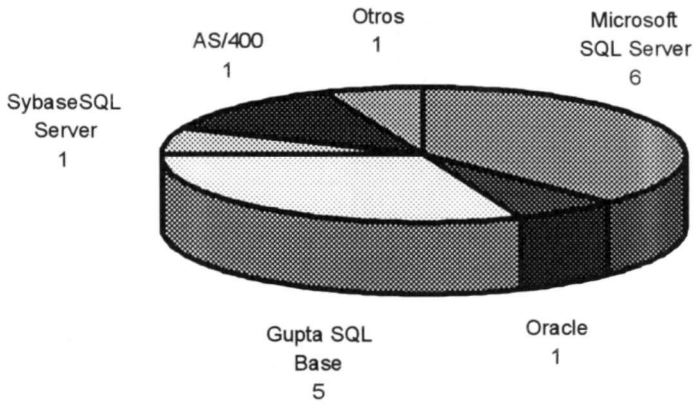
El número de personas que desarrollan proyectos sobre Cliente/Servidor en cada empresa varía, ya que se obtuvo lo siguiente:

* dos empresas tienen cuatro personas desarrollando proyectos C/S
* una empresa tiene tres personas
* una empresa tiene veinticinco
* tres empresas tienen dos personas
* tres empresas tienen ocho personas



Gráfica 24. Plataformas donde corren los Manejadores de Bases de Datos

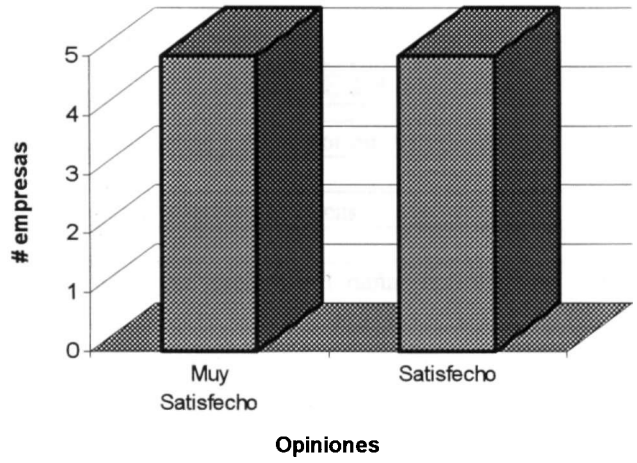
De las 10 empresas encuestadas, 4 de ellas utilizan 2 plataformas diferentes: Windows NT y UNIX. Esto da como resultado que estas dos plataformas sean las más utilizadas en Tampico.



Gráfica 25. Servidor de Base de Datos utilizado

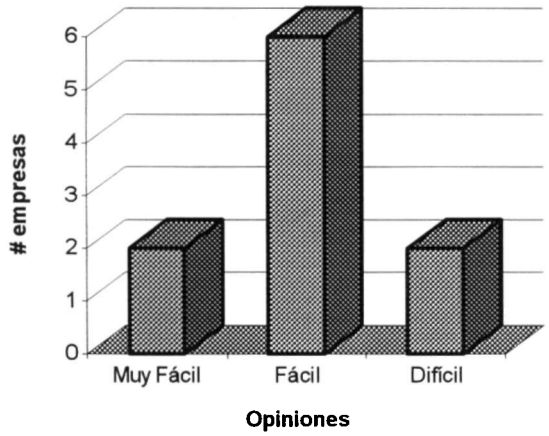
Los servidores de bases de datos más utilizados son: Microsoft SQL Server y posteriormente Gupta SQL Base, ya que de las diez empresas, seis y cinco de ellas, respectivamente, son las que más lo utilizan.

Sección VI: Requerimientos y Opiniones



Gráfica 26. Satisfacción por el desempeño del producto

La mitad de las empresas indicaron estar satisfechas o muy satisfechas, lo cual indica que la herramienta seleccionada sí les ha sido útil o al menos cumple con sus necesidades y requerimientos. El grado de satisfacción fue en base a una comparación previa con sus anteriores productos.



Gráfica 27. Facilidad de uso de la herramienta

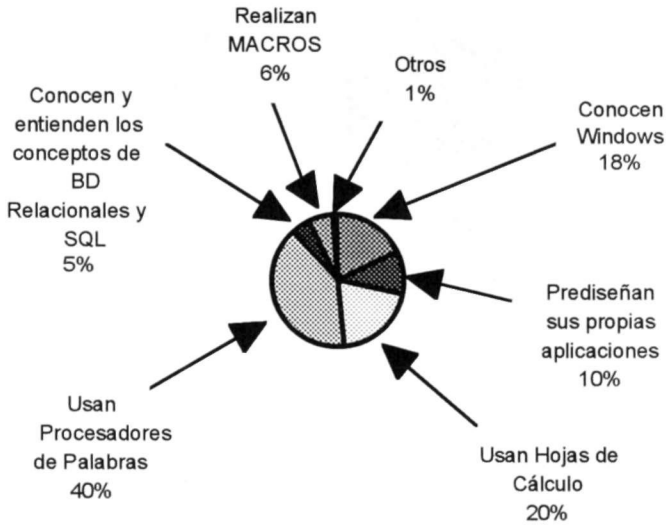
Con respecto a la facilidad de uso, seis de las diez empresas indicaron que sí era fácil de utilizar la herramienta, sin embargo dos de ellas indicaron que les era difícil el utilizarla. El grado de facilidad se indicó con respecto a los productos anteriores utilizados por las empresas encuestadas.

La tabla 5 muestra los aspectos a mejorar que desean las empresas sobre la herramienta Cliente/Servidor que utilizan.

ASPECTOS A MEJORAR DE LA HERRAMIENTA C/S
• Precios, que la tecnología sea más accesible a un mayor número de clientes
• Incremento en "performance"
• Facilidad de uso y administración, vía interfaces gráficas
• "Stored Procedures", "triggers"
• Sea mejor en cuanto a los "bugs" que presenta al dañar aplicaciones (los fuentes) cuando se están generando
• El "quick objects" crea "overhead", la versión actual.
• El reporteador de "quick object"

Tabla 5. Aspectos a mejorar de la herramienta Cliente/Servidor

Sección VII: Usuarios Finales



Gráfica 28. Promedio de habilidades que los usuarios finales tienen

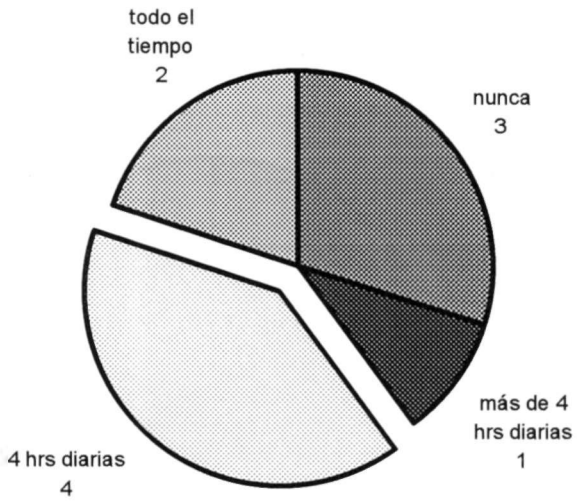
Las empresas encuestadas indicaron que los usuarios utilizan y conocen más los procesadores de palabras y en segundo término, las hojas de cálculo. En "otros" caen que algunos usuarios conocen programación orientada a objetos basada en eventos, sin embargo son menos significativos que lo antes mencionado.

La tabla 6 muestra las herramientas complementarias que se están utilizando actualmente, orientadas al usuario final.

HERRAMIENTAS COMPLEMENTARIAS		
REPORTEADORES	GRAFICADORES	CONSULTAS
• QUEST	√ Chart-FX	α SQL de Progress
• Forrest & Trees	√ GBS	α Excel
• Access	√ Harvard Graphics	α QUEST
• Reporter	√ Excel	α SQL
	√ QUEST	α Access
	√ VBGraf de Visual Basic	

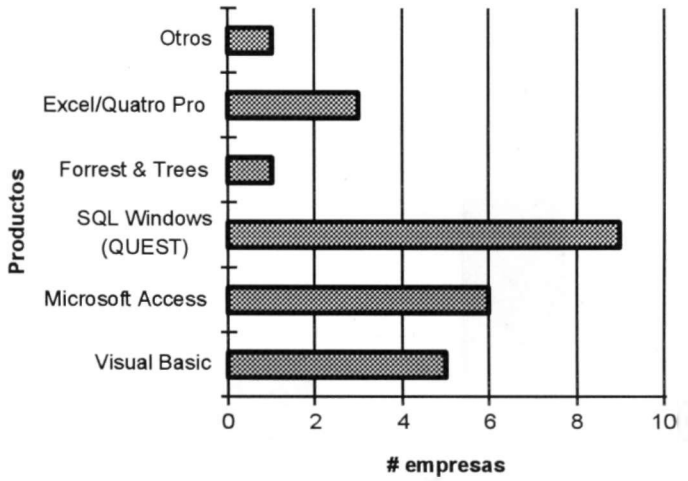
Tabla 6. Herramientas complementarias

El uso de herramientas complementarias es variado, sin embargo "QUEST" es la más utilizada en: reporteadores, graficadores y consultas.



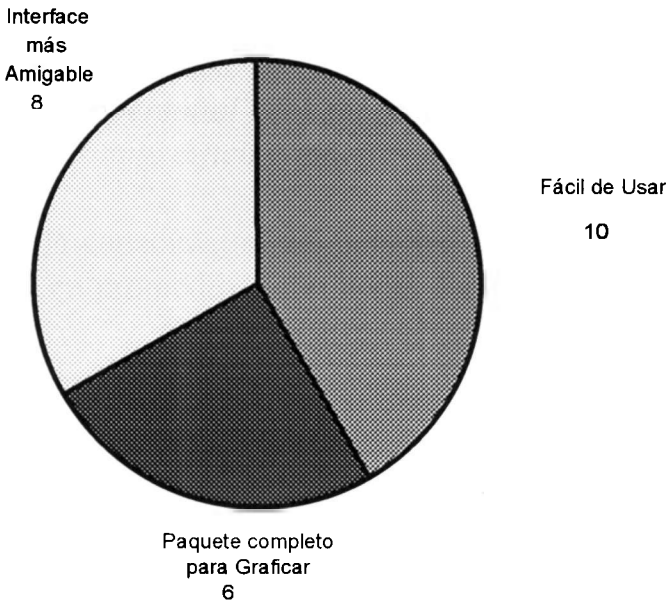
Gráfica 29. Frecuencia con que utilizan los usuarios finales los graficadores, consultas, reporteadores

Cuatro empresas de las diez, opinan que los usuarios finales sólo utilizan en promedio 4 hrs. diarias las herramientas enfocadas a ellos, sin embargo, 3 empresas opinan que nunca las usan. En este último caso, los usuarios todavía se resisten a utilizar nuevas herramientas y sólo se limitan a utilizar los sistemas computacionales que les han realizado. Aún existe muy poca iniciativa por parte de los usuarios para desarrollarse más.



Gráfica 30. Productos que se evaluaron para la selección de la herramienta enfocada al Usuario Final

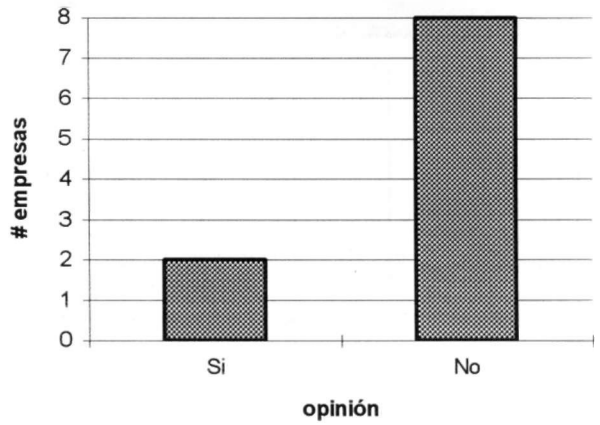
Nueve de las diez empresas evaluaron a "QUEST", lo cual da una muestra muy significativa para ser la mejor herramienta seleccionada enfocada al usuario final; seguida de Microsoft Access y Visual Basic en donde seis y cinco empresas respectivamente, las evaluaron.



Gráfica 31. Características que hicieron posible la selección de la Herramienta orientada a Usuarios Finales

De acuerdo a las características anteriores, las diez empresas encuestadas opinaron que la característica básica para seleccionar la herramienta orientada a usuarios finales es que sea "fácil de usar".

Sección VIII: Capacitación

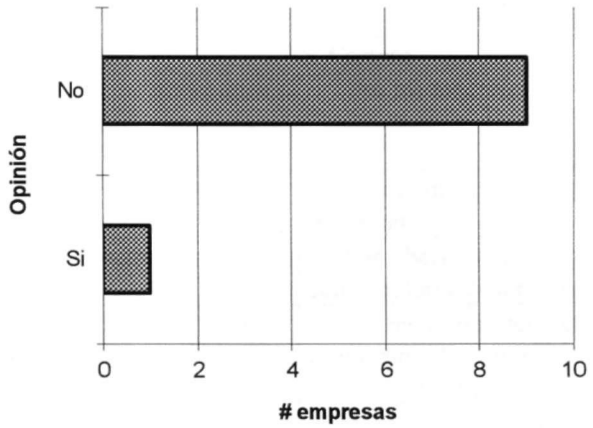


Gráfica 32. Opiniones sobre la dificultad de llevar a cabo la capacitación del personal, para aprender el uso de la Herramienta de desarrollo bajo la Tecnología Cliente/Servidor

Ocho de las diez empresas opinaron que no fue difícil el llevar a cabo la capacitación al personal, sin embargo dos de ellas opinaron que:

** debido a que existen deficiencias en los servicios y la falta de capacidad por parte del proveedor intermedios*
** debido a que todo el personal está sobrecargado de trabajo, por lo que no hay mucho tiempo y es muy difícil capacitar a un grupo de personas.*

Con respecto a la capacitación del personal de desarrollo, esta oscila entre dos semanas a tres, en promedio, ocho horas diarias de lunes a viernes en cada una de las empresas.



Gráfica 33. Opiniones sobre la dificultad para llevar a cabo la capacitación a los usuarios finales, para aprender el uso de las nuevas aplicaciones

En forma muy significativa nueve de las diez empresas, indicaron que no fue difícil la capacitación a usuarios finales sobre el uso de las nuevas aplicaciones, sólo una empresa indicó que sí fue difícil debido a los siguientes motivos:

* *No conocen conceptos de computación*
* *Falta de iniciativa para aprender más*
* *No tienen tiempo*
* *Mucha rotación de personal*

La capacitación a usuarios finales, en promedio, fue de 30 a 40 hrs.

Sección IX: Ventajas/Desventajas

En la tabla 7 se muestra una relación de opiniones de las empresas encuestadas sobre las ventajas y desventajas de Cliente/Servidor.

VENTAJAS	DESVENTAJAS
• La facilidad del uso de la interface	√ Costosa
• Facilidad para crecer y/o cambiar cualquiera de sus componentes (front-end, back-end) con un mínimo esfuerzo	√ Altos los requerimientos de "hardware"
• Alta productividad	√ Curvas de aprendizaje altas contra promesas de proveedores
• Velocidad de procesamiento y drásticamente menor tráfico en la red	√ Las herramientas que lo utilizan son altos consumidores de recursos
• Trabajar en ambiente "windows" para los usuarios finales	√ La memoria del servidor se incrementa , en promedio, alrededor de 16 mb por cada gigabyte de almacenamiento
• Seguridad en el manejo de bases de datos	√ Poco soporte técnico
• La independencia que existe entre la base de datos y los programas que las manejan (la interface)	√ Poca documentación técnica en el país
• El mantenimiento a los sistemas	√ "Performance"
• El uso de excelentes interfaces como las gráficas	√ Con el uso de generadores automáticos de aplicaciones se generan también importantes "overheads"
• Disminución en el costo de mantenimiento	
• Menor tiempo de capacitación	
• Más fácil la migración	
• Menor tiempo de desarrollo	

Tabla 7. Ventajas y Desventajas de Cliente/Servidor

Sección X: Metodologías

En esta sección se analizó el tipo de metodología que las empresas en Tampico usan actualmente, llegando al siguiente análisis:

Siete de las diez empresas utilizan el método tradicional; una no utiliza ningún tipo de metodología, ya que todo lo realizan sobre el mismo desarrollo del proyecto; dos empresas utilizan algunas fases del tradicional, y le adicionaron otras más, como el prototipeo.

Bajo este contexto, podemos ver que la mayoría de las empresas aún siguen utilizando el tradicional; esto, debido a que no han encontrado una metodología que cumpla con sus expectativas lo cual indica que están buscando la más adecuada.

Existen algunos factores que se evaluaron para conocer aún más sobre la metodología que utilizan, los cuales son los siguientes:

* *Intervención de usuarios finales en las fases de la metodología que usan*, las diez empresas indicaron que sólo intervienen en la fase de análisis o en la etapa inicial; tres indicaron que intervienen en las pruebas; y dos en prototipeo.
* *Técnicas adicionales,* una empresa utiliza herramientas CASE en Análisis y Diseño-Prototipo.
* *Problemas encontrados,* los usuarios no se involucran en el desarrollo del proyecto, escasamente intervienen en el análisis; la gente le tiene tedio a los documentos ya que evita realizarlos esto trae como consecuencia la falta de documentación donde se indique: el inicio, alcance, término del proyecto así como de las personas que intervienen, tanto a nivel desarrollador como de usuario final; falta de planeación formal del proyecto, ya que existe informalidad; falta de una metodología adecuada.

Sección XI: Comentarios

Se les pidió a cada persona que contestó la encuesta que se calificara de acuerdo a ciertas preguntas, esto sirvió para sondear su nivel cultural sobre Cliente/Servidor. De esta calificación se obtuvo un promedio por cada pregunta dando como resultado lo siguiente:

* Conocimiento de la tecnología Cliente/Servidor:	9.125
* Conocimiento de la herramienta de "software" que utilizan bajo Cliente/Servidor:	8.375
* Conocimiento sobre Bases de Datos:	8.625

En resumen, la encuesta aplicada mencionada anteriormente presenta las siguientes perspectivas a considerar:

√ La herramienta más utilizada para desarrollo de aplicaciones Cliente/Servidor es SQL Server/SQL Windows
√ La principal razón para utilizar la Arquitectura Cliente/Servidor es la Alta Productividad por parte de los desarrolladores
√ Los equipos más utilizados son 80486 a 50 MHZ
√ La plataforma más utilizada para correr los manejadores de bases de datos es Windows NT
√ El servidor de base de datos utilizado es Microsoft SQL Server
√ Las empresas opinan estar muy satisfechas del producto y su facilidad de uso
√ En cuanto al tiempo de uso de graficadores, reporteadores y consultas por parte de los usuarios finales, el promedio es de 4 hrs diarias
√ QUEST fue la mejor herramienta enfocada para el uso del usuario final, siendo su característica principal la de ser fácil de usar
√ La capacitación del personal de sistemas y de los usuarios finales no fue difícil de llevarse a cabo, tomando como marco de referencia la capacitación que tenían anteriormente
√ La metodología utilizada por parte de las empresas es la tradicional

CAPÍTULO 6
ESTRATEGIA DE IMPLANTACIÓN

6.1 Elementos a considerar

El desarrollo y soporte de aplicaciones Cliente/Servidor es un cambio intenso de los sistemas de información y de las empresas. Estas aplicaciones son complejas y costosas, y los procesos de implementación son sorpresivamente extensos cuando una nueva tecnología se le agrega. en consecuencia, el personal de sistemas de información está siendo presionado para aprender nuevos conceptos y herramientas, así como a empezar a conocer la organización. Por otro lado, los arquitectos técnicos y diseñadores deben cambiar de *diez mil* tecnologías mientras trata de mantener un grado de consistencia y estandarización en toda la empresa. Los administradores de proyectos están también presionados para liberar aplicaciones rápidamente. [GOTTESDIENER, 1994]

El hecho es, sin embargo, que las aplicaciones Cliente/Servidor son más difíciles de desarrollar que las aplicaciones tradicionales. La complejidad, el costo, habilidades e inmadurez de la tecnología, entre otros, son factores de dificultad que deben tomarse en cuenta ahora, en estas aplicaciones. Por lo tanto, es conveniente el proponer una metodología adecuada a este tipo de aplicaciones.

A continuación se propone una metodología para el desarrollo de aplicaciones Cliente/Servidor en la ciudad de Tampico Tamaulipas, esta proposición es en base a los resultados de las encuestas aplicadas mencionados en el capítulo 4, también, a las tres metodologías explicadas en el capítulo 2 y a la experiencia que se ha tenido en el desarrollo de aplicaciones. Al realizar un estudio y análisis de cada una de estas metodologías se detectaron ciertos aspectos deficientes, que se consideraron para el desarrollo de los elementos a considerar de la estrategia de implantación para el desarrollo de aplicaciones Cliente/Servidor. A continuación se mencionan estos aspectos:

Tradicional,

* Nada está hecho hasta que todas las fases están terminadas
* Al terminar una fase se olvida completamente
* En todas las fases el analista es el que tiene relación con el usuario final
* No existe la documentación formal

Estructurado,

* En algunas fases el analista es el que tiene relación con el usuario final, en otras es el diseñador
* No existe la documentación
* No considera al prototipeo

LEAD,

* No tiene retroalimentación

Es importante el denotar el tipo de términos que se utilizarán en la explicación de los elementos de la estrategia:

⇒Empresa cliente: tiene dos puntos de vista diferentes: 1) puede ser algún departamento de la empresa que desarrolla sus propias aplicaciones; o, 2) es la empresa que está comprando el producto al proveedor.

⇒Empresa desarrolladora: tiene dos puntos de vista diferentes: 1) es el departamento de sistemas encargado de desarrollar las aplicaciones propias de la empresa; o, 2) es el proveedor al cual le están comprando las aplicaciones.

Los pasos de la estrategia propuesta son los siguientes:
[GANE, 1993] [GOTTESDIENER, 1994]

0) Demostración de productos Cliente/Servidor: al existir un gran auge con la arquitectura Cliente/Servidor, es importante el concientizar a las empresas

cliente de lo que es, los requerimientos en "hardware"/"software"; haciendo una demostración de los productos que la empresa desarrolladora ofrece, de esta forma la empresa cliente se da idea de lo que se tiene en el mercado y si le es necesario o no.

1) Definir requerimientos, en base a objetivos y limitantes: esta fase del proyecto, trata de definir lo que la empresa cliente necesita tratando de llegar a uno o varios objetivos sobre el proyecto deseado, es recomendable el delimitar el proyecto en varias etapas para cumplir acertadamente con los objetivos. También, se deben de buscar las limitantes que se tengan para considerarlas en toda la trayectoria del proyecto.

   Por lo general en esta etapa, siempre son dos personas las involucradas, una por parte de la empresa desarrolladora y otra por la empresa cliente, de tal forma que se llegue a un acuerdo de la compraventa de la aplicación.

2) Recopilación de información: una vez definidos los requerimientos, es necesario recopilar información mediante los siguientes aspectos:

   * entrevistas
   * observación de actividades
   * documentación

   Las personas que se deben involucrar en esta fase son: los usuarios finales a quienes vaya orientado el proyecto, y los analistas, quienes recopilan información. En esta fase es importante que sólo los analistas intervengan, ya que es el primer contacto que se tiene con la empresa cliente o los usuarios finales, y es muy difícil el que intervengan más personas debido a la falta de confianza.

3) Análisis: etapa en la cual se modela la información obtenida, esto mediante un modelo esencial, y el modelo ambiental de cada etapa del proyecto, el cual incluye: Diagrama de Contexto, Diagrama de Flujo de Datos, y Diagrama Entidad-Relación.

   En esta etapa es conveniente que intervengan: los usuarios finales, analistas, diseñadores y desarrolladores. En esta parte, sí es importante y necesario que por parte de la empresa desarrolladora intervengan todas las personas de sistemas, ya que entre todos conocerán lo que es el proyecto,

opinarán sobre lo que se puede o no hacer, ideas sobre el desarrollo del mismo, limitantes en la construcción-producción, entre otras.

4) Prototipeo Preliminar: el prototipeo debe realizarse para conocer el *cascarón* de una de las etapas del proyecto, de tal forma que se tenga una idea clara de lo que llegará a ser el proyecto cuando esté terminado. En esta etapa es recomendable que trabajen juntos: usuarios finales, analistas, diseñadores y programadores; para diseñar un buen prototipo, ya que de aquí depende en gran parte la venta o creación del proyecto final.

5) Punto de Aceptación: una vez analizado por parte de la empresa cliente, el prototipo se deriva el aceptar seguir adelante con el proyecto o no.

6) Planeación del Proyecto: esta fase es una de las más importantes de la estrategia, debido a que se basa en documentación formal haciendo una propuesta-compromiso sobre lo que se va a hacer durante todo el proyecto, este documento deberá firmarse por ambas partes. Para darle seguimiento al proyecto es recomendable realizar:

   - definición de etapas del proyecto y sus necesidades, interviniendo: usuarios finales, analistas, diseñadores y programadores
   - definir recursos (equipo y personal) en cada etapa, interviniendo: analistas, diseñadores y programadores
   - GANTT, intervienen: usuarios finales, analistas, desarrolladores, diseñadores, y analistas

   Para la realización del documento compromiso sólo es necesario que interactúen una persona por parte de la empresa cliente y otra por la empresa desarrolladora.

7) Construcción-Producción y Pruebas: en Tampico existe una empresa desarrolladora que considera a los usuarios finales en esta etapa para que participen en conjunto con analistas, programadores, diseñadores. Esta participación tiene dos puntos de vista, los cuales son: 1) los usuarios finales empiecen a relacionarse con el desarrollo de las aplicaciones que utilizarán, esto se puede dar en algunos casos especiales en donde el tipo de usuario final se presta para conocer y entender conceptos computacionales de programación. Esto puede darse en base al tipo de "software" que se

utilice, ya que éste puede ser muy sencillo de utilizar. Este caso aplica, cuando las aplicaciones son desarrolladas por las mismas empresas; 2) las empresas cliente pueden enlazar su personal de sistemas con las empresas desarrolladoras, de tal forma que ellos mismos empiezan a adentrarse a sus propias aplicaciones y a futuro no dependerán al 100% de su proveedor. Hoy en día, este aspecto es lo más práctico y eficaz para que cada empresa siga adelante con sus propias aplicaciones evitándose gastos innecesarios. Este caso aplica, cuando las aplicaciones las compran.

Esta etapa involucra un aspecto interesante a considerar el cual es la retroalimentación.

7.1) Retroalimentación: una vez realizada la construcción del proyecto por las fases definidas anteriormente, al término de cada fase se debe considerar el dar una retroalimentación de lo que se hizo contra lo planeado (paso 6); de tal forma que, si existe algún problema se debe regresar al paso 6 para corroborar.

8) Implantación/Capacitación: una vez que se vayan terminando las etapas del proyecto se deben implantar y capacitar al personal que lo va a utilizar. Este procedimiento debe realizarse al mismo tiempo que los programadores estén realizando las siguientes etapas, de tal forma que los analistas sean los que capaciten e implanten las aplicaciones y no existan tiempos muertos.

9) Garantías: (esta fase sólo aplica cuando se compran las aplicaciones a las empresas desarrolladoras) debido a que el desarrollo de este tipo de aplicaciones es costoso, es importante ofrecer Garantías, esto implica el dar soporte al usuario o cliente (según sea el caso) de las fases realizadas. Dependiendo de la aplicación se debe ofrecer un cierto periodo de tiempo de garantía, ésta puede aplicarse a: diseño de pantallas, reportes y consultas; fallas del sistema; cierto número de horas adicionales de capacitación; entre otras. Esto, siempre y cuando se respete la documentación realizada durante la planeación del proyecto.

10) Seguimiento-Post Producción: (esta fase sólo aplica cuando se compran las aplicaciones a las empresas desarrolladoras) la empresa desarrolladora podrá ofrecer un seguimiento a futuro a la empresa cliente, de tal forma que ambas partes obtengan beneficios. El seguimiento puede darse al considerar los siguientes aspectos:

- ◊ venta de la aplicación (fuentes) del proyecto a realizar
- ◊ venta capacitaciones, del "software" a utilizar orientado al personal de sistemas
- ◊ venta "runtime" del "software"
- ◊ venta de actualizaciones del "runtime"
- ◊ venta completa de una instalación Cliente/Servidor

En resumen, este tipo de estrategia es aplicable a cualquier desarrollo de aplicaciones, sin embargo se consideraron algunos elementos de compañías desarrolladoras que están utilizando actualmente.

Otro aspecto importante, es el tratar de incluir en ciertas etapas al usuario final para lograr mejores resultados.

CONCLUSIONES

Durante el transcurso de la investigación que se obtuvo mediante el sondeo a las empresas de la ciudad de Tampico Tamaulipas, se observaron algunos aspectos interesantes sobre la arquitectura Cliente/Servidor, los cuales son:

- Las empresas están satisfechas con la migración hacia Cliente/Servidor
- El haber utilizado Cliente/Servidor por parte de las empresas, fue debido a la alta productividad que los desarrolladores producían
- El tiempo de uso de graficadores, consultas y reporteadores (bajo cliente/Servidor) por parte de los usuarios finales es en promedio, de 4 horas diarias. Lo cual significa la mitad del tiempo de trabajo diario, dando como resultado la necesidad que se tiene de utilizar este tipo de recursos

Otro aspecto interesante es, el hecho de que esta arquitectura tiene aproximadamente de ocho a diez años en el mercado, por lo que es importante el reconocer que en la ciudad de Tampico, que es tan sólo un segmento del país tiene aproximadamente de dos a tres años de utilización en algunas empresas, y que con seguridad en un periodo de un año a dos, se usará en la mayoría de las empresas.

Durante esta investigación también se obtuvieron resultados interesantes sobre el tipo de metodología que las empresas utilizan para el desarrollo de aplicaciones Cliente/Servidor, de estos resultados obtenidos se realizó un análisis para la creación de una metodología con ciertos elementos que ayudaran a las empresas a desarrollar aplicaciones Cliente/Servidor de acuerdo a sus necesidades. La tabla 8 muestra los pasos a seguir que se obtuvieron durante el desarrollo de la estrategia de implantación, para el desarrollo de las aplicaciones bajo la arquitectura Cliente/Servidor.

PASOS	NOMBRES
0	Demostración de productos Cliente/Servidor
1	Definir requerimientos, en base a objetivos y limitantes
2	Recopilación de información
3	Análisis
4	Prototipeo preliminar
5	Punto de aceptación
6	Planeación del proyecto
7	Construcción-Producción y pruebas
8	Implantación/Capacitación
9	Garantías
10	Seguimiento-post producción

Tabla 8. Pasos para la estrategia de implantación

Como conclusión final, la arquitectura Cliente/Servidor ha tenido buenos resultados en las empresas de Tampico Tam., lo cual indica que en ciudades como México D.F., Monterrey N.L., Guadalajara Jal., entre otras, la estén utilizando en un porcentaje de empresas más amplio.

Proyectos Futuros

Existen diversas áreas o sectores de interés, donde la arquitectura C/S se implante. Sin embargo, el sector EDUCACIÓN es el que mostraría un panorama más amplio y completo de la utilización de C/S.

ANEXO 1
PLATAFORMAS BAJO CLIENTE/SERVIDOR

- Ambientes PC

1) Microsoft/Sybase Structured Query Language Server 4.2

SQL Server es un DBMS (Sistema de Administración de Base de Datos), que soporta almacenamiento de procedimientos sobre el servidor de base de datos. El almacenamiento de procedimientos son rutinas de SQL que son almacenadas como parte de la base de datos; se ejecutan completamente en el servidor cuando son llamadas por los clientes, brindando un positivo impacto desempeño de la base de datos. Gran número de procedimientos son incluidos para realizar diversas funciones: administración de usuarios, monitoreo del servidor, y, creación de nuevas bases de datos, entre otros. [SALEMI, 1994]

SQL Server, también soporta un tipo especial de procedimientos almacenados llamados "triggers" (disparadores), los cuales son procedimientos que se ejecutan automáticamente cuando ciertos eventos (como INSERT, UPDATE, DELETE) se llevan a cabo en el servidor. Los "triggers" prácticamente sirven para cumplir las reglas de integridad de datos, porque automáticamente se activan.

SQL Server, da el soporte de seguridad a grupos de usuarios en el acceso de las bases de datos, esto es, creando privilegios para dar buena administración a los datos.

La versión 4.2 contiene tolerancia a fallas y resplado en tiempo real con "disk mirroring", lo cual permite rendimiento a la base de datos del servidor. El proceso de espejo, constantemente realiza una copia exacta de los archivos primarios en un disco de respaldo. Esto hace que disminuya el desempeño, pero vale la pena por la seguridad y la integridad de la información que se obtiene.

El número máximo de bases de datos en cada server son 256. Microsoft provee el ODCB (Conectividad Abierta de Base de Datos), un API (Interface de Programación de Aplicaciones) y librerías que permiten a los clientes de cualquier servidor conectarse a otro SQL Server.

El Microsoft SQL Server, requiere al menos un microprocesador 80286 corriendo OS/2 1.3 o superior, utilizar un microprocesador 80386 es más recomendable. La memoria mínima son 8 MB, sin embargo se recomienda 12 MB. El espacio mínimo en disco es de 20 MB. Puede comunicarse con: LAN Manager, Novell y, LAN Server de IBM. Y probablemente con el protocolo IPX/SPX (Internetwork Packet Exchange/Sequenced Packet Exchange). [UMAR, 1993]

Sybase SQL Server NLM (Network Loadable Module) requiere microprocesardor 80386 corriendo sobre el sistema operativo Netware 3.11. Sybase se recomienda con 12 MB en RAM y 20MB de espacio en disco.

El lenguaje de SQL Server es el TRANSACT-SQL.

SQL Server de Microsoft incluye una herramienta para administración que permite fácilmente administrar a los DBA (Administrador de Bases de Datos) otros SQL Servers, a través del uso de íconos de Windows, menús, listas, y cajas de diálogo, esta herramienta es un SQL basado en Windows 3.x.

Los proveedores de este producto son: para OS/2 su proveedor es Microsoft Corporation, para NLM su proveedor es Sybase Inc.

2) Gupta SQLBase 5.0

SQLBase, comenzó con los 3 sistemas operativos primarios: DOS, OS/2 y Netware. SQLBase es uno de los pocos RDBMS (Administradores de Sistemas de Bases de Datos Relacional) que da soporte completo al estándar ANSI nivel 2 SQL. El SQLBase soporta el RI (Integridad Referencial Declarativa). El RI es declarado como parte de la definición

de la base de datos, además de poseer "triggers" y almacenamiento de procedimientos. [SALEMI, 1994]

Las bases de datos pueden particionarse para almacenarse en múltiples discos o volúmenes. Dependiendo de la plataforma, una base de datos SQLBase puede estar en el rango de más de 500 gigabytes. Algunos estudios comparativos han demostrado que SQLBase es uno de los más rápidos SQL Servers. La razón de ello, es la compresión de datos.

SQL soporta procedimientos SQL pre-compilados, que se almacenan en el servidor para una rápida ejecución.

La versión para DOS requiere al menos de un microprocesador 80286 con 2 MB en RAM, 10 MB en disco, y PC/MS-DOS 3.1 o superior. La versión OS/2 es de 16-bits y requiere un microprocesador 80286 corriendo OS/2 1.0 o más, con 4 MB en RAM y 10 MB en disco. La versión NLM requiere un microprocesador 80386 o más, corriendo Netware 3.11, con 8 MB en RAM y 10 MB en disco. SQLBase soporta diversos protocolos, incluyendo: APPC (Comunicación Avanzada Punto-a-Punto) de IBM, Named Pipes, Net-BIOS (Network Basic Input Output System), TCP/IP, IPX/SPX. [UMAR, 1993]

Su lenguaje es SQLTalk, el cual es compatible con Ansi Level 2.

Gupta incluye dos utilerías de administración de bases de datos frontal: 1) modo caracter y 2) utilería basada en Windows 3.x. SQLTalk/Character es la interface de comando en línea que permite al DBA entrar directamente a los estatutos SQL. Con SQLTalk/Windows hace que sea más fácil el administrar y crear consultas a la base de datos que con el modo caracter.

El proveedor de este producto es Gupta Technologies Inc.

3) IBM OS/2 Database Manager 2.0

Es un sistema de clase 2 que corre sobre OS/2 1.31 o 2.0. Su característica principal es que el SQL es compatible con el DB2 para

"mainframe" y se puede conectar en forma transparente y compartir datos con DB2 a través del paquete DDCS/2 (Servicios de Conección de Bases de Datos Distribuidas/2). Incluye el soporte para el RI, como el SQLBase, RI es implementado a través de llaves foráneas y primarias que incluyen estatutos SQL que manejen cómo borrar datos-padres que existan o tengan dependencias en otras tablas. [SALEMI, 1994]

Se requiere microprocesador 80286, con 8 MB en RAM y 15 MB en disco, para correr bajo OS/2 1.31; y para el 2.0, se necesita mircroprocesador 80386 con 12 MB en RAM y 15 MB en disco.

Utiliza su NetBIOS o APPC para dar la comunicación entre el cliente y el servidor, y puede trabajar con cualquier otra red que soporte estos protocolos.

Su lenguaje nativo SQL es compatible con ANSI Level 2. El "software" incluye 3 "front-ends": CLI (Interface de comando en Línea), un Módulo de Herramientas de Bases de Datos y el Administrador de Consultas.

El proveedor de este producto es IBM Inc.

4) INGRES for OS/2

Ingres fue el desarrollo original de la optimización basada en el costo de un RDBMS, una característica es que empezó por ser el estándar en las industrias. Su optimización aún es considerada por la industria debido a que, su SQL está basado en el álgebra que asegura que las consultas sean ejecutadas rápidamente. [SALEMI, 1994]

Ingres es el único que soporta 4 tipos de métodos de almacenamiento de datos. Estos se determinan por el DBA, al momento de crear una tabla y puede ser cambiado en cualquier momento cuando la tabla se incremente.

Ingres para OS/2 tiene una limitante: los usuarios no pueden accesar bases de datos de otros tipos, incluso si accesan de su mismo tipo de base de datos deben estar en la misma red. Está limitada a 32 usuarios.

Ingres para OS/2 requiere OS/2 1.3 o superior, corriendo bajo microprocesadores 80386, con 5 MB en RAM, ocupando 10 MB en disco. Su "driver" INGRES/NET trabaja sólo con NetBIOS o con IPX/SPX.

Ingres tiene dos versiones de SQL: Ingres/SQL y Open SQL. Este "software" está limitado a "front-ends" del modo caracter, y corre en terminales tontas o en PC's.

El proveedor de este producto es Ingres Corporation.

5) ORACLE Server 6.0

Su característica principal es la portabilidad y escalabilidad, existen diversas versiones: Macintosh, "Mainframes", y variantes del UNIX. El código escrito en alguna plataforma puede ser fácilmente transportado a otro. [SALEMI, 1994]

ORACLE provee soporte para procesamiento distribuido para las bases de datos cliente/servidor, ya que tiene el protocolo de comunicación SQLNet el cual soporta: NetBIOS, IPX/SPX, Named Pipes, APPC (Advanced Program to Program Communications) y 3270 de IBM, DECNet (Digital Equipment Corporation) y TCP/IP (Transmission Control Program/Internet Protocol). Múltiples versiones de SQLNet pueden correrse sobre un server de ORACLE, y puede manejar ligas con bases de datos remotas sin la intervención de los usuarios. Esto hace posible que, diversas bases de datos remotas se conecten al mismo tiempo y pueda el usuario modificar o actualizar su base de datos desde cualquier lugar. [UMAR, 1993]

ORACLE, demanda poca memoria del servidor, la versión para OS/2 requiere 12 MB en RAM y al menos 32 MB para soportar al número máximo de usuarios. El "software" requiere 10 MB en disco. La versión NLM requiere un mínimo de 16 MB en RAM, y 64 MB para el soporte al número máximo de usuarios.

ORACLE, tiene su propio lenguaje SQL a través del PL/SQL (Lenguaje Procedural /SQL). Su "front-end" es el SQLDBA, en modo caracter, la interface es en línea y se utiliza para administrar o para realizar consultas.

El proveedor de este producto es Oracle Corporation.

Ambientes UNIX

1) INGRES para UNIX y VAX/VMS

Dependiendo de la plataforma de hardware, INGRES para UNIX puede soportar 300 o más usuarios accesando la base de datos tanto de sistemas cliente como de terminales dedicadas. [SALEMI, 1994]

INGRES soporta: diccionario de base de datos distribuido, el cual es compartido a otras bases de datos de otros servers; optimización distribuida de consultas, que optimiza los estatutos y comandos de SQL para una mayor ejecución costo-efectividad en las bases de datos de todos los sistemas; transacciones distribuidas, las cuales no se consideran hasta que todas las modificaciones individuales de las diferentes bases de datos sean realizadas completamente.

INGRES puede comunicarse con otros ambientes operativos como el de Gateway, el cual provee el mismo acceso para DEC RDB (Digital Equipment Corporation Relational Data Base) y bases de datos RMS corriendo sobre una VAX, bases de datos Hewlett-Packard AllBase corriendo sobre sistemas HP, y bases de datos DB2 o IMS corriendo sobre "mainframes" IBM.

En cuanto a los requerimientos de hardware y "software" se puede mencionar lo siguiente: la cantidad de RAM en disco que requiere INGRES varía de plataforma a plataforma por lo cual es difícil recomendar los requerimientos. VAX/VMS,e INGRES corren sobre 30 versiones de UNIX, incluyendo el Sistema V de AT&T, SunOS, AIX de IBM, ULTRIX de DEC, HP-UX de HP y SCO UNIX. INGRES también ofrece las versiones para los mercados europeos y asiáticos.

Los protocolos de comunicación entre los clientes y el servidor son provistos por INGRES/Net corriendo sobre TCP/IP para UNIX y DECNet o TPC/IP para VMS.

El lenguaje nativo SQL para INGRES es INGRES/SQL, OPEN SQL y QUEL, y EQUEL.

INGRES provee el mismo "front-end" para UNIX, VMS u OS/2. Existen algunas, herramientas para crear "front-end" para estaciones RISC (Reduced Instruction Set Computing). Con INGRES/VISION se pueden crear aplicaciones para ayudar al proceso de desarrollo sobre DOS o UNIX. Con INGRES/WINDOWS-4GL permite crear "front-ends" sobre Windows 3.x. [BERSON, 1992]

El proveedor de este producto es Ingres Corporation.

2) ORACLE Server 6.0

La escalabilidad y portabilidad de ORACLE continúa a través de sus sistemas. Existen arriba de 30 versiones tanto para VAX/VMS como para UNIX, que incluyen todas las variantes como los de SCO, Sistema V de AT&T, AIX, y SunOS. ORACLE ofrece una versión que corre bajo Data General y el sistema operativo de minis AOS-VS. Cualquier versión de ORACLE puede ser interconectada con otra versión mediante SQLNet haciéndolo líder en bases de datos distribuidas. [SALEMI, 1994]

ORACLE, permite a usuarios el acceso de datos sobre otras bases de datos como: DB2, SQL/DS, RDBMS en "mainframes" de IBM y DEC para VAX/VMS.

ORACLE corre sobre 80 plataformas, incluyendo la de DEC para VAX/VMS, VM y MVS para "mainframes" de IBM.

SQLNet de ORACLE provee comunicación cliente/servidor para todas las versiones. Soporta TPC/IP, DECNet, APPC y conexiones asíncronas. SQLNet 2.0, permite y proporciona la capacidad de implementar un TNS

(Medio transparente de Red) el cual es una comunicación común API, que permite el acceso de diversas bases de datos ORACLE en el protocolo de la red.

Tanto el SQL de ORACLE como el PL/SQL son compatibles, como el propio SQLNet, y se utilizan para crear "front-ends" y comunicarse con el "back-end" de ORACLE.

Las versiones de ORACLE incluyen una versión de SQLDBA apropiado para el sistema operativo del servidor . Cualquier versión de SQLDBA puede llamar a cualquier servidor ORACLE usando la versión apropiada de SQLNet.

Todas las herramientas de programación y "front-ends" disponibles para las versiones de PC soportan el "back-end" a través de SQLNet. El código fuente precompilado hace más fácil el mover aplicaciones de PL/SQL de una plataforma a otra.

El proveedor de este producto es Oracle Corporation.

3) SYBASE SQL

La característica más importante de SQL Server es la VSA (Arquitectura Virtual Server), en donde los sistemas de SYBASE soportan el multiprocesamiento simétrico. En 1990, las versiones de VSA de SQL server fueron disponibles para varios sistemas de multiprocesamiento, como: DEC VAX, superservers Sun 4/400 y 4/600MP SPARC, y sistemas AT/T StartServerE. [SALEMI, 1994]

Algunas pruebas demostraron que SQL Server VSA proveen de un 30% a un 50% de incremento de desempeño sobre versiones de UNIX o VMS, así como la habilidad de soportar arriba de 1000 bases de datos simultáneas.

SYBASE, puede integrar datos de otras DBMS con datos de SQL server a través de su procesamiento distribuido API Open Client/Server. Open Client provee soporte de acceso a clientes a través de diferentes

protocolos de red, como: DECNet, TCP/IP, IPX/SPX y Names Pipes. Open Server es la base para un número de SYBASE Open Gateways para RDBMS que no sean del tipo SYBASE incluyendo INFORMIX, INGRES, ORACLE, DEC Rdb y RMS, y DB2 de IBM. Open Gateway permite a cualquier cliente de SYBASE accesar a otras bases de datos como si fueran SQL SERVER.

SQL Server comparte con ORACLE portabilidad y escalabilidad, ofreciendo versiones para VAX/VMS y las versiones de UNIX, como: AIX, Sistema V de AT&T, y el sistema operativo de NeXT MACH.

SYBASE SQL SERVER corre sobre VAX/VMX y 30 versiones mayores de UNIX. Es el único que tiene una versión para NEXT y también la única que soporta sistemas de multiprocesamiento simétrico.

Todas las versiones de SQL Server usan TRANSACT-SQL como su lenguaje, que incluye el soporte para procedimientos de almacenamiento y "triggers". SYBASE aún no adopta el soporte para movimientos de "scroll" que fueron desarrollados por MICROSOFT para su versión de SQL Server. Pero SYBASE, provee un gran número de herramientas de desarrollo para UNIX que usan APT-SQL.

El conjunto de herramientas de SQL usan: TRANSACT-SQL y APT-SQL; los cuales incluyen: APT Workbench, es el ambiente de desarrollo en base a formas que dejan al programador crear consultas pintando en la pantalla; Data Workbench, es un reporteador que los programadores pueden a través de la pantalla; APT-build, es un generador de código y aplicaciones.

Se pueden crear programas "front-end" en : C, COBOL, FORTRAN.

El proveedor de este producto es Sybase Inc.

4) Gupta SQLBase

Su principal característica es su precio: $9995 dls, para un número ilimitado de usuarios, es el más económico de los servidores de bases de datos para UNIX. [SALEMI, 1994]

Requiere un Sun SPARCStation o SPARCServer corriendo sobre SunOS 4.1.1 (o superior), con al menos 4 MB en RAM, versión para 5 usuarios. El "software" requiere aproximadamente 10 MB en disco.

Estaciones de DOS y OS/2 pueden comunicarse con SQLBase vía TCP/IP, y usa el SQLTalk como las versiones basadas en PC. Pueden crearse "front-end" en C o COBOL.

El proveedor de este producto es GUPTA Technologies Inc.

5) INFORMIX

Presenta dos diferentes clases de RDBMS, que actúan como servidores de bases de datos: INFORMIX-SE e INFORMIX-ONLINE. INFORMIX-SE, es un DBMS multiusuario diseñado para pequeñas bases de datos que se pueden utilizar en un sistema ciente/servidor con "software" de comunicación INFORMIX-NET. INFORMIX-ONLINE tiene todas las características de la versión SE pero con mayor desempeño y capacidad. De los dos, la versión ONLINE es la mejor para las aplicaciones de cliente/servidor. [SALEMI, 1994]

La versión 5.0 de INFORMIX-ONLINE, está diseñada para dar un alto desempeño en sistemas de transacciones, en donde la integridad de los datos y la disponibilidad de la base de datos es necesaria.

La integridad de los datos es provista a través de un número de aspectos, incluyen: procedimientos de almacenamiento, "triggers", e integridad de entidades.
La disponibilidad de las bases de datos es por medio de "disk mirroring".

Los procedimientos de almacenamiento de INFORMIX son implementados de la misma forma que SYBASE; los procedimientos son almacenados como parte de la base de datos y pueden ser ejecutados por cualquier aplicación de "front-end" que accese a la base de datos.

Los "triggers", son definidos por el usuario, éstos pueden ser usados para proveer integridad, pero no son necesarios, porque INFORMIX contiene RI que junto con ANSI SQL Nivel 2 lo provee. En resumen, INFORMIX provee integridad de entidades.

La versión 5.0 de INFORMIX-ONLINE, está disponible para sistemas SPARC corriendo sobre: SunOS, Sistema V de AT&T, y la versión 4.1 de HP-UX de Hewlett Packard.

INFORMIX-STAR usa TCP/IP para dar comunicación entre sistemas en línea y sistemas cliente. Los clientes basados en DOS pueden usar INFORMIX-NET PC sobre TCP/IP para comunicarse con el sistema INFORMIX-STAR.

INFORMIX trae consigo su propio SQL.

INFORMIX-ONLINE, ofrece un número de utilerías de administración basadas en UNIX. El DB-Menu es un manejador de menús, que permite al DBA hacer respaldos y restauraciones en línea.

Algunos comandos en línea proveen al DBA realizar arreglos a tablas con índices dañados. Se pueden crear "front-ends" en: C, COBOL, y FORTRAN.

El proveedor de este producto es Informix "software" Inc.

- Ambientes en Sistemas de Minicomputadoras

1) RDB/VMS Versión 4.0

DEC (Digital Equipment Corporation) ha sido líder en el mercado de sistemas de minicomputadoras, y fueron creadores del manejador de base de datos relacional RDB/VMS . [SALEMI, 1994]

Lo principal de DEC ha sido VAX/VMS, que ha sido diseñado para sistemas multiusuarios y sistemas operativos de acceso en tiempo real.
RDB/VMS corre en cualquier sistema VMS, algunas versiones se incluyen con el sistema operativo. Las aplicaciones corren sobre cualquier VAX sin tener que comprar el producto completo. El sistema RDB es necesario para desarrollar aplicaciones y proveer accesos cliente/servidor a la base de datos.
RDB/VMS usa el concepto "storage maps" para almacenar datos, el cual permite que el DBA asigne espacio de varias formas.

El SERDB 4.1, junto con RDB/VMS provee un alto grado de seguridad en los datos. El RDB/ELN, brinda una alta capacidad de captura de datos en tiempo real.

RDB/VMS corre sobre computadoras DEC que utilicen el sistema operativo VMS, incluyendo: estaciones DECStation, superservidores MICROVAX y minicomputadoras VAX. El número de usuarios que soporta depende de la capacidad de la plataforma de hardware.

RDB/VMS utiliza el protocolo de comunicación DECNET para comunicarse con otros sistemas VAX corriendo VMS o ULTRIX. El paquete completo de RDB incluye: el protocolo de comunicación SQL/Services para accesar a las base de datos de sistemas DEC. Se pueden accesar PC's corriendo PC/MS DOS conectadas a la VAX sobre un DECNET LAN.

El SQL de RDB/VMS es compatible con el nivel 2 de ANSI. DEC provee su propio lenguaje relacional, llamado RDML (Lenguaje de Administración de Base de Datos Relacional). RDB utiliza optimización dinámica para optimizar los estatutos de SQL. La optimización dinámica es única para RDB pudiendo dinámicamente combinar entre diferentes técnicas de búsquedas basadas en consultas.

El proveedor de este producto es: Digital Equipment Corporation.

2) ALLBASE/SQL

Hewlett Packard es el único que provee dos diferentes interfaces para ALLBASE/SQL: una interface presenta los datos relacionales ALLBASE basándose en el modelo de base de datos para redes y, la otra tiene una interface SQL para usarse en aplicaciones cliente/servidor. [SALEMI, 1994]

ALLBASE permite que algunos usuarios realicen procesamientos de bases de datos distribuidas remotas como si fueran parte de una base de datos local. HP mercadea, TURBOIMAGE versión de base de datos para red, y TURBOCONECT dejando el acceso a usuarios (en forma transparente) a base de datos basados en imágenes.

ALLBASE corre sobre cualquier mini HP3000 o HP9000. Tanto el sistema operativo MPE/XL como HP-UX, son soportados.

HP soporta TCP/IP para comunicar clientes y ALLBASE. La comunicación entre éstos es trasparente gracias al propio protocolo de HP NS (Sistema de Red).

HPSQL es el lenguaje de ALLBASE.

El proveedor de este producto es Hewlett-Packard Company.

3) SQL/400 Versión 2.1.1

Una de las motivaciones para el desarrollo de AS/400 fue la existencia de un SAA (Arquitectura de Sistemas de Aplicación) compatible con OS2 y DB2 de IBM. SQL/400 provee una interface compatible de datos con SQL y una AS/400, con procesamiento distribuido y capacidades cliente/servidor.
SQL/400 provee la misma estructura de las base de datos de DB2 y OS/2. [SALEMI, 1994]

SQL/400 sólo corre bajo OS/400, el sistema operativo para la línea AS/400.

SQL/400 se comunica con otros sistema de IBM a través de APPC. Desafortunadamente, APPC requiere de extensa memoria y mucho RAM en sistemas PC-MS DOS para los sistemas clientes basados en DOS. TCP/IP está disponible para AS/400 y puede ser usado por AIX (otra variante de IBM que corre bajo AS/400).

SQL/400 es compatible con DB2 SQL de IBM. El lenguaje de OS/400 es el 3GL RPG/400, el soporte sobre COBOL también está disponible. Los datos en el sistema AS/400 pueden ser accesados a través de aplicaciones fuera de la interface SQL/400, esto puede implicar el violar los principios de integridad de los datos de Modelos Relacionales.

La serie AS/400 aún no ha crecido como otras, orientada a los sistemas "host" para terminales dedicadas. Los "front-end" para clientes basados en PC los soportan a través de un Query Manager que viene en el administrador de bases de datos del OS/2. El principal motivo por el que las PC basadas en MS DOS no puedan "tener front-end", es la gran cantidad de memoria que requiere.

El proveedor de este producto es IBM Inc.

- Ambientes Mainframes

1) SQL/DS

Existen 2 versiones de SQL/DS: SQL/DS 3.1 para DOS/VSE y SQL/DS 3.3 para VM. [SALEMI, 1994]

DOS/VSE es el sistema operativo para "mainframe" más expandido para el soporte a usuarios. VM, ha sido el sistema operativo preferido de IBM para pequeños "mainframes". VM ha sido preferido porque ofrece mejor capacidad en procesamiento de transacciones de tiempo real.

El sistema operativo VM corre sobre cualquier "mainframe" System/370, así como de: 4381 a ES/9000, y de 3090s a ES/9000s. Tanto VM como SQL/DS requieren pocos recursos del sistema.
SQL/DS para VM requiere el mismo protocolo de comunicación que DB2.

ORACLE, es el tercer proveedor de RDBMS que provee un "gateway" para SQL/DS. Su SQLConnect para SQL/DS soporta las 2 versiones proviendo la capacidad de cliente/servidor para cualquier "front-end" compatible con ORACLE Server.

SQL/DS para VM es 100% compatible con DB2 SQL, la versión de VSE SQL es compatible con ANSI LEVEL 1.

El proveedor de este producto es IBM Inc.

2) ORACLE

La única diferencia entre ORACLE para "mainframe" y el ORACLE de otras plataformas es el sistema operativo que cada uno soporta y la capacidad de datos que provee un "mainframe". El RDBMS es el mismo. [SALEMI, 1994]
La versión de "mainframe" para ORACLE corre bajo los sistemas operativos MVS y VM. SQLNet se utiliza para comunicarse con los clientes de otras bases de datos de ORACLE bindando la capacidad de bases de datos distribuidas.

Cada versión de ORACLE usa SQLNet para accesar al DBMS. La versión "mainframe" de SQLNet soporta: APPC, TCP/IP, DECNet, SNA y protocolos asíncronos de red. Otros protocolos se pueden soportar a través de "links" (ligas) entre diferentes bases de datos ORACLE.

Las versiones de ORACLE para "mainframe" usan el mismo SQL y PL/SQL como las otras versiones. Las aplicaciones y los "front-ends" pueden accesarse desde cualquier base de datos ORACLE por medio de su propio SQLNet.
El proveedor de este producto es Oracle Corporation.

ANEXO 2
ENCUESTA

Tampico Tam., Julio 1995.

A quien corresponda:

Soy alumna de la maestría en Administración de Sistemas de Información con especialidad en Informática del ITESM Campus Monterrey, cuyo objetivo principal para obtener el grado de Maestro se requiere el realizar una tesis en el área, la cual consiste en hacer investigación sobre tópicos actuales. La tesis que presentaré en agosto de este año es la siguiente:

"IMPLANTACION DE ARQUITECTURA CLIENTE/SERVIDOR: ¿VENTAJA COMPETITIVA EN LAS ORGANIZACIONES?"

El objetivo principal de dicha tesis es el demostrar que esta arquitectura está causando impactos positivos en las organizaciones de Tampico y el proponer una estrategia de implantación que se adecúe al medio. Para poder lograr esto, es necesario que investigue algunos puntos importantes, tales como: si existe mercado para esta arquitectura, si se utiliza, si la inversión fue adecuada, si no existe resistencia al cambio, la metodología utilizada, entre otros.

Por lo que mi propósito es el obtener su ayuda y colaboración para responder la siguiente encuesta, haciéndole incapié en el hecho de que al obtener cualquier información de su empresa ésta se utilizará en forma confidencial y solamente será tratada para efectos estadísticos de la tesis. Además, me comprometo a entregar a Ud. un documento sobre el análisis de esta investigación al término de ésta.

Dicha encuesta será llevada a cabo por medio de secciones, indicándose cada una de ellas.

Los tipos de respuestas que contiene la encuesta son las siguientes:

* abiertas: se le da opción a contestar abiertamente lo que desee
* Seleccionar una opción : ___ si ____ no
* Indicar prioridades :
  _1_ ahorro en costos
  _2_ soporte total en windows
* Porcentajes :
  5% usuarios finales 20% Director Gral.
  80% Gerente de Sistemas

De antemano le agradezco el tiempo que me ha brindado así como la colaboración que Ud. me proporcionará para la realización y término de esta investigación para llevar a cabo la tesis.

Atentamente,

Lic. Alma Gloria Flores Trejo

I. Sección: Información sobre la empresa.

1. Nombre de la empresa a encuestar :_____
2. Puesto que ocupa actualmente:_____
3. Indique el número de empleados que tiene la empresa actualmente:

___ menos de 10 ___10-20 ___21-30 ___31-50 ___51-100

II. Sección: Software de Desarrollo

4. ¿Cuál es el software de desarrollo que ha utilizado para sus aplicaciones?:

_____

_____

5. ¿Cuál es el software de desarrollo bajo la tecnología Cliente/Servidor, que ha utilizado para sus aplicaciones?:

_____

_____

6. ¿Cuál ha sido el tiempo de uso del software bajo la tecnología Cliente/Servidor?:_____

7. Indique las herramientas con las que empezó a desarrollar software para aprender la tecnología Cliente/Servidor:

___ COBOL ___ C ó C++ ___ 4GLWINDOWS ___ Visual Basic
___ XBase/Clipper/Paradox/Access ___Object View ___ PowerBuilder
___ SmallTalk ___ SQLWindows
___ Otros (menciónelos):

_____

_____

8. ¿Cuáles fueron las razones para decidirse a usar el producto que utiliza la tecnología Cliente/Servidor (indique la prioridad, donde 1 es la más alta):

____ Ahorro en costos
___ Capacidad de prototipeo rápida
___ Alta productividad de los desarrolladores
___ Facilidades para programar en equipo
___ Interfaces orientadas a objetos
___ Soporte total en Windows (GUI)
____ Soporte de múltiples bases de datos
___ Vanguardia Tecnológica
___ Otras (menciónelas):

_____

_____

9. Muestre el rango (en porcentaje) de la toma de decisiones del siguiente tipo de personas para la adquisición de la herramienta Cliente/Servidor:

___ Desarrolladores
___ Líderes de Proyecto
___ Director de Sistemas de Información
___ Usuarios Finales
___ Administrador de Bases de Datos
___ Administradores de Sistemas de Información
___ Otras (menciónelas):

_____

_____

III. Sección: Proveedores.

10. ¿Quiénes son sus proveedores para el desarrollo de las aplicaciones?:

_____

_____

11. ¿Cuáles fueron los criterios de evaluación para la selección de estos proveedores?:

_____

_____

IV. Sección: Hardware.

La siguiente matriz muestra una relación de HW, en donde Ud. tendrá que indicar el número de equipos que estén utilizando actualmente para el desarrollo y uso de las aplicaciones bajo Cliente/Servidor:

Estaciones Cliente:

RAM

PROCESADORES	4	8	12	16	más
486-33					
486-50					
486-66					
Pentium					

Otro tipo:

_____

_____

Servidores:

	RAM	DISCO DURO	PROCESADOR
HP-NET SERVER			
ACER ALTOS			
PENTIUM			

Otro tipo:

V. Sección Bases de Datos y Plataformas.

12. En promedio, ¿Cuánta gente de su empresa está desarrollando proyectos sobre la herramienta Cliente/Servidor?: _____

13. ¿Bajo qué plataformas corren los manejadores de bases de datos:

___ Netware ___ OS/2 ___ DOS ___ Sun Os ___ UNIX
___ Windows NT ___ Otros (menciónelos):

14. ¿Están usando algún servidor de bases de datos, si es así indique cuales:

___ Oracle ___ Sybase SQLServer ___ Microsoft SQLServer
___ Informix ___ DB2 ___ AS/400
___ Gupta SQL Server ___ Otros (menciónelos):

VI. Seccion. Requerimientos y Opiniones.

15. ¿Está Ud. satisfecho con el desempeño del producto?

___ Muy satisfecho ___ Poco satisfecho
___ Satisfecho ___ Insatisfecho

16. ¿Es fácil de usar la herramienta?

___ Muy Fácil ___ Difícil
___ Fácil ___ Muy Difícil

17. ¿Qué le gustaría que mejorara de la herramienta (stored procedures, trigger's, etc), explique:

VII. Sección: Usuarios Finales.

18. ¿Cuál es el promedio de habilidades que los usuarios finales tienen en cuanto a?:

___ Conocen Windows
___ Prediseñan sus propias aplicaciones
___ Usan hojas de cálculo
___ Usan procesadores de palabras
___ Realizan "Macros"
___ Conocen y entienden los conceptos de bases de datos relacionales y SQL
___ Otros (menciónelos):

_____

_____

19. Mencione las herramientas complementarias que están utilizando, orientadas a usuario final:

Reporteadores :

_____

_____

Graficadores:

_____

_____

Consultas ("Queries") :

_____

_____

Otras (menciónelas):

_____

_____

20. ¿Con qué frecuencia utilizan los usuarios finales los graficadores, "queries", reporteadores?:

___ nunca
___ más de 4 horas diarias promedio
___ 4 horas diarias promedio
___ todo el tiempo

21. ¿Qué productos se evaluaron para la selección de la herramienta enfocada al usuario final?:

___ Visual Basic
___ "Queries" para Oracle/Informix
___ Microsoft Access
___ SQL Windows (Quest)
___ Otros (menciónelos):
___ Forrest & Trees
___ Business Object
___ Excel/Quatro Pro

_____

_____

22. ¿Qué características hicieron posible el hecho de haber seleccionado la herramienta que utilizan los usuarios finales de otras?:

___ fácil de usar
___ paquete completo para graficar, uso de "queries", reportes
___ Interface más amigable
___ Otras (menciónelas):

_____

_____

VIII. Sección: Capacitación

23. ¿Fue difícil llevar a cabo la capacitación del personal, tomando como marco de referencia anteriores capacitaciones, para aprender el uso de la herramienta de desarrollo bajo la tecnología Cliente/Servidor?:

___ Si ___ No (pase a la preg. 25)

24. Indique los motivos:

___ Resistencia al cambio
___ Falta de cultura informática
___ La mayoría del personal son técnicos no profesionistas
___ Falta de iniciativa por parte de las personas
___ No quieren aprender otras herramientas
___ Actualmente se encuentran desmotivados
___ Otros (menciónelos):

_____

_____

25. En promedio, ¿Cuánto tiempo se llevó el capacitar al personal?:

_____

_____

26. ¿Fué difícil llevar a cabo la capacitación a los usuarios finales, tomando como marco de referencia anteriores capacitaciones, para aprender el uso de las nuevas aplicaciones?: ___ Si ___ No (pase a la preg. 28)

27. Indique los motivos:

___ Resistencia al cambio
___ No conocen conceptos de computación
___ Inseguridad, requieren al personal de sistemas
___ Falta de iniciativa para aprender más
___ No desean aprender otras herramientas
___ Desmotivación
___ Otros (menciónelos):

_____

28. ¿Cuánto tiempo se llevó la capacitación a usuarios finales?:

IX. Sección: Comentarios.

29. A cualquier nivel (ej. herramienta de desarrollo, aplicaciones, etc.) , ¿Qué ventajas han visualizado de cliente/servidor?:

30. A cualquier nivel (ej. herramienta de desarrollo, aplicaciones, etc.) , ¿Qué desventajas han visualizado de cliente/servidor?:

X. Seccion: Metodologias.

Tomando como base el ciclo tradicional de desarrollo de un sistema computacional:

ANÁLISIS DISEÑO PROGRAMACIÓN PRUEBAS MTTO.

31. ¿Cuál es la metodología que Uds. utilizan para el desarrollo de aplicaciones Cliente/Servidor?:

32. ¿Intervienen los usuarios en el desarrollo de sus aplicaciones? ¿En qué momento?:

33. ¿Utilizan el prototipeo? ¿Cómo? ¿En qué fase de la metodología?:

34. ¿Utilizan alguna técnica adicional en la metodología, como herramientas CASE? ¿Cuáles son y en qué fase las involucran?:

35. ¿Cuáles han sido los problemas que se han enfrentado o encontrado durante el el desarrollo del proyecto?

XI Seccion: Comentarios

36. ¿Cómo se califica Ud. en cada uno de los siguientes puntos?: (del 1-10)

___ Conocimiento de la Tecnología Cliente/Servidor
___ Conocimiento de la herramienta de software que utilizan bajo Cliente/Servidor
___ Conocimiento sobre Bases de Datos (relacionales, jerárquicas, redes)

37. Comentarios Adicionales:

REFERENCIAS

[BERSON, 1992]
BERSON, Alex. CLIENT/SERVER ARCHITECTURE. ed McGraw-Hill, U.S.A., 1992.

[BURCH, 1991]
BURCH, John G.; Strater Felix R. SISTEMAS DE INFORMACIÓN. ed Noriega LIMUSA, México, 1991.

[CLIENT, 1994]
CLIENT Server Designs. THE LEAD MODEL: CREATING SUCCESSFUL CLIENT/SERVER APPLICATIONS. U.S.A., 1994.

[COMPUTER WORLD, Num. 387, 1993]
COMPUTERWORLD, Año 14, Num. 387, México D.F., 1993.

[COMPUTERWORLD, Num. 391, 1994]
COMPUTERWORLD, Año 14, Num. 391, México D.F., 1994.

[DIECK, 1994]
DIECK, Assad Ma. Elena. BENEFICIOS DE LA TECNOLOGÍA DE INFORMACIÓN ENFOQUE TÉCNICO Y PRÁCTICO EN LA CIUDAD DE MONTERREY. México, 1994.

[GANE, 1993]
GANE, Chris. "A Model-Driven Methodology for Client/Server System Design". EN: CASE TRENDS, BUSINESS AND ADVANCED TECHNOLOGY SOLUTIONS FOR SOFTWARE PROFESSIONALS. Volúmen 5, Número 6, Págs. 23-28, 1993.

[GOTTESDIENER, 1994]
GOTTESDIENER, Ellen. "What's New In Client/Server Development Methods". EN: APPLICATION DEVELOPMENT TRENDS, BUSINESS AND TECHNOLOGY SOLUTIONS FOR SOFTWARE DEVELOPMENT. Volúmen 1, Número 12, pág. 50-61, 1994.

[KEEN, 1991]
KEEN, Peter G.W., SHAPING THE FUTURE: BUSINESS DESIGN THROUGH INFORMATION TECHNOLOGY. Harvard Business School., Press. U.S.A., 1991.

[KENDALL, 1991]
KENDALL, Kenneth E.; Kendall Julie E. ANÁLISIS Y DISEÑO DE SISTEMAS. ed. Prentice Hall, México, 1991.

[LUFTMAN, LEWIS, ALDACH, 1993]
LUFTMAN, J.N.; Lewis, P.R.; Oldach, S.H., "Transforming the enterprise: The alignment of business and information technology strategies". EN: IBM SYSTEMS JOURNAL. Volúmen 32. Número 1, pág. 198-220, 1993.

[RED, Num. 25,1993]
Revista RED, Año III, Num. 25, México D.F., 1993.

[RED, Num. 47,1994]
Revista RED, Año IV, Num. 47, México D.F., 1994.

[RENAUD, 1993]
RENAUD E. Paul. INTRODUCTION TO CLIENT SERVER SYSTEMS. ed Wiley Professional Computing, U.S.A., 1993.

[SALEMI, 1994]
SALEMI Joe. PC MAGAZINE GUIDE TO CLIENT/SERVER DATABASES. ed. Ziff-Davis Press, U.S.A., 1994.

[SENN, 1987]
SENN A. James. SISTEMAS DE INFORMACIÓN PARA LA ADMINISTRACIÓN. ed. Iberoamérica, México, 1987.

[SMITH, 1992]
SMITH Patrick. CLIENT/SERVER COMPUTING. ed Prentice Hall Computer Publishing, U.S.A., 1992.

[TRAVIS, 1993]
TRAVIS Dewire Dawna. CLIENT/SERVER COMPUTING. ed McGraw-Hill, U.S.A., 1993.

[VARHOL, 1993]
VARHOL D. Peter. CLIENT/SERVER TECHNOLOGY. ed. Computer Technology Research Corp., U.S.A., 1993.

[UMAR, 1993]
UMAR Amjad. DISTRIBUTED COMPUTING. ed. Prentice Hall, U.S.A., 1993

[UNIX, 1993]
UNIX Review, 1993.

[YOURDON, 1993]
YOURDON Edward. ANÁLISIS ESTRUCTURADO MODERNO. ed. Prentice Hall, México, 1993.