



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Escuela Nacional de Estudios Profesionales
Campus Aragón

DESARROLLO DE APLICACIONES
CLIENTE/SERVIDOR PARA LAN BAJO
SISTEMAS OPERATIVOS DE MICROSOFT DE
32 BITS

T E S I S

Que para obtener el título de
INGENIERO EN COMPUTACION
p r e s e n t a

LAZARO ESTRADA MENDOZA



ASESOR DE TESIS: ING. JUAN GASTALDI PEREZ

San Juan de Aragón, Estado de México

2000



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

**ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ARAGÓN**

**JEFATURA DE INGENIERÍA EN
COMPUTACIÓN**

OFICIO: ENAR/JACO/105/00.

ASUNTO: Asignación de Jurado.

LIC. ALBERTO IBARRA ROSAS
Secretario Académico
Presente.

RECIBIDO
MAR 20 2000
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN

Por este conducto me permito presentar a usted, nombre de los profesores que sugiero integren el Sínoo del Examen Profesional del alumno LAZARO ESTRADA MENDOZA, que presenta el tema de tesis: "DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR PARA LAN BAJO SISTEMAS OPERATIVOS DE MICROSOFT DE 32 BITS".

- PRESIDENTE:** ING. JUAN GASTALDI PÉREZ
- VOCAL:** ING. SILVIA VEGA MUYTOY
- SECRETARIO:** ING. RAÚL LEÓN Y BERBER
- SUPLENTE:** ING. MARCELO PÉREZ MEDEL
- SUPLENTE:** ING. GABRIEL ALDERÉTE CERÓN

Quiero subrayar que el director de tesis es el Ing. Juan Gastaldi Pérez, el cual está incluido con base en lo que reza el reglamento de Exámenes Profesionales de esta Escuela.

Sin otro en particular, me es grato enviarle un cordial saludo.

RECIBIDO
MAR 20 2000 10 55

A T E N T A M E N T E
"POR MI RAZA HABLARÁ EL ESPÍRITU"
San Juan de Aragón, Edo. de México, Marzo 20 del 2000.
EL JEFE DE ÁREA

ING. JESÚS DÍAZ BARRIGA ARCEO

c.c.p. Lic. Ma. Teresa Luna Sánchez.- Jefa del Departamento de Servicios Escolares.
Ing. Juan Gastaldi Pérez.- Director de Tesis.

JDBA/gga.

Recibido
J.G.P.

Agradecimientos

Este trabajo esta dedicado a mis padres. Por su amor paciencia orientación y sacrificios para dar a sus hijos lo mejor de sí. Por haber hecho todo lo posible por mi educación en mi niñez cuando uno no tiene aún el suficiente uso de razón para comprender el verdadero valor de la educación académica. Aunque las circunstancias en un principio no les fueron del todo favorables se esforzaron en guiar y animar mis primeros pasos de mi educación ... Se que con este trabajo también se ven alcanzados algunos de sus sueños.

Llevaré siempre en mi mente como legado la frase que bien pronuncian ellos: "La educación es el más preciado tesoro y lo único que podemos heredarte, aquilátalo y utilízalo para servir a tus semejantes". ... A ambos les quedo profundamente agradecido.

Agradezco también a todos aquellos profesores de mi noble Universidad que contribuyeron en mi educación, me reservo el mencionar sus nombres por temor de omitir a alguno de ellos.

Lázaro Estrada Mendoza.

TEMARIO

- **AGRADECIMIENTOS**
- **CONVERSIONES**
- **LIMITACIONES DE ESTA TESIS**
- **OBJETIVO DE ESTA TESIS**

DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR PARA LAN BAJO SISTEMAS OPERATIVOS DE MICROSOFT DE 32 BITS

- 1. Componentes de los Sistemas C/S.**
- 2. Arquitectura Cliente/Servidor.**
- 3. Creación de la plataforma para Aplicaciones C/S.**
- 4. Metodología.**
- 5. Herramientas para el desarrollo de Aplicaciones C/S.**
- 6. Requisitos de Rendimiento y Seguridad de una Aplicación C/S.**
- 7. Control de los Costos en la ejecución de Aplicaciones C/S.**

CONCLUSIONES

APÉNDICE A

Ejemplo de una aplicación en Visual Basic Ver 5.0.

Documentos.

Código fuente de la aplicación.

GLOSARIO

BIBLIOGRAFÍA

DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR PARA LAN BAJO SISTEMAS OPERATIVOS DE MICROSOFT DE 32 BITS

- AGRADECIMIENTOS
- CONVENSIONES
- LIMITACIONES DE ESTA TESIS
- OBJETIVO DE ESTA TESIS

1. Componentes de los Sistemas Cliente/Servidor.

1.1	Introducción	1
1.2	Evolución de los esquemas de Cómputo Cliente/Servidor.	2
1.2.1	Primer Etapa	2
1.2.2	Segunda Etapa	3
1.2.3	Tercera Etapa	3
1.2.4	Cuarta Etapa	4
1.2.5	Quinta Etapa	5
1.2.5.1	Entorno Centralizado Cliente/Servidor	6
1.2.5.2	Entorno Cliente/Servidor distribuido	6
1.2.6	Impulsos del Modelo Cliente/Servidor	7
1.2.6.1	Monitores TP	8
1.2.6.2	Groupware	9
1.2.6.3	Objetos Distribuidos.	9
1.3	Que es una Aplicación.	10
1.4	Que son las GUI.	10
1.4.1	Network User Interface	11
1.5	Concepto de Front-End.	12
1.6	Concepto de Back-End.	12

2. Arquitectura Cliente/Servidor.

2.1	Introducción	15
2.2	Que es el modelo Cliente/Servidor.	16
2.3	Cuál es la función primordial de la arquitectura Cliente/Servidor.	17
2.3.1	¿Por qué las organizaciones migran hacia una Arquitectura Cliente/Servidor?	17
2.3.1.1	Reducción en los costos de procesamiento.	18
2.3.1.2	Reducción en los costos de desarrollo de aplicaciones.	18
2.4	Concepto de Cliente.	18
2.4.1	Tipos de plataformas Cliente	19
2.4.2	Redirector	19
2.5	Concepto de Servidor.	20
2.5.1	Tipos de Servidor	21
2.5.2	Características de hardware de la computadora servidor.	22
2.6	Concepto de Red.	23
2.6.1	Tipos de Arquitecturas del Modelo Cliente/Servidor	24
2.6.2	Arquitecturas del Modelo Cliente/Servidor basadas en Software.	24
2.6.3	Arquitecturas Cliente/Servidor basadas en hardware	25
2.6.3.1	Servidor de archivos	25
2.6.3.2	Peer to Peer (Punto a Punto)	25

2.7 Tipos principales de Arquitectura Cliente/Servidor en el mercado.	26
2.7.1 Acceso Remoto a Datos (RDA)	26
2.7.2 Servidor de Base de Datos o Base de Datos Distribuida (DBS).....	27
2.7.3 Servidor de Aplicaciones o Aplicaciones Distribuidas (APS).....	27
2.8 Ventajas del modelo Cliente/Servidor en General.	28
2.9 Desventajas del modelo Cliente/Servidor en General.	29

3. Creación de la Plataforma para Aplicaciones Cliente/Servidor.

3.1 Introducción.	30
3.2 Diferencia entre aplicaciones de 16 bits y de 32 bits.	31
3.2.1 Ventajas de diseñar aplicaciones de 32 bits.....	31
3.3.1 Ethernet proporciona resultados suficientes	35
3.3.2 Cálculo del número de usuarios soportados.....	36
3.4 Un poco de historia de Windows NT.	37
3.4.2 ¿Por qué se elige Windows para este trabajo?.....	40
3.4.3 Características principales de Windows NT.....	41
3.4.4 Comparación de Microsoft Windows NT Server con Novell Netware 5.....	48
3.5 Requerimientos de hardware Intel para Sistemas Operativos actuales MS de 32 bits.....	50
3.5.1 Instalando Windows NT Server y Windows NT Workstation.....	50
3.5.2 Requerimientos de hardware para instalar aplicaciones Cliente/Servidor.....	52
3.6 Programación en Windows.	53
3.7 Filosofía de diseño de Aplicaciones en Windows.	53
3.7.1 Convenciones acerca del ratón.....	54
3.7.2 Convenciones acerca del teclado en Windows.....	54
3.7.3 Convenciones acerca de menú en Windows.....	55
3.7.4 Que son las DDE y OLE.....	56
3.7.5 Que son las DLL.....	58
3.7.6 Que son las APIs.....	59
3.7.6.1 Ventajas de utilizar APIs.....	60
3.7.6.2 Desventajas de utilizar APIs.....	60
3.7.7 Registro de Windows.....	60
3.8 Proceso de comunicación de una Aplicación Cliente/Servidor.	64
3.8.1 Fallas en la Comunicación Cliente/Servidor.....	66
3.8.2 Comunicación Cliente/Servidor dentro de una aplicación.....	67
3.8.2.1 Comunicación entre un cliente y un servidor en proceso.....	67
3.8.2.3 COM, GUID y el registro del sistema.....	68
3.9 Bases de datos utilizadas como fuentes de información.....	69
3.10 Motores de Bases de Datos.	69
3.10.1 Concepto de OLTP.....	70
3.10.2 Concepto de RPC.....	70
3.10.3 Procedimientos Almacenados.....	70
3.11 Conectividad en Bases de datos abiertas (ODBC).	71
3.11.1 Cómo trabaja ODBC.....	72
3.11.2 WOSA (Windows Open Service Architecture).....	75
3.12 Concepto de Middleware.	75
3.12.1 Como funciona un software de middleware.....	76
3.12.2 Ejemplo de un software de middleware.....	77
3.12.3 Futuro del middleware.....	77
3.13 Adecuación de la plataforma de hardware.	77

4. Metodología.

4.1 Introducción.....	79
4.2 Desarrollo de la Metodología para Aplicaciones Cliente/Servidor.....	80
4.3 Decida cómo los Datos estarán Distribuidos.....	82
4.4 Ciclo de vida de las aplicaciones Cliente/Servidor.....	82
4.4.1 Documentos requeridos para los proyectos de desarrollo Cliente/Servidor.....	88
4.4.2 Técnicas de análisis y diseño aplicables al diseño Cliente/Servidor.....	89
4.4.2.1 Diagramación.....	89
4.4.2.2 Codificación.....	90
4.4.2.3 Técnicas de Programación.....	90
4.4.2.4 Análisis mediante componentes.....	93
4.4.2.5 Herramientas requeridas.....	99
4.5 Diferencias entre aplicaciones controladas por Procedimientos y.....	100
4.6 Ambientes de desarrollo de Aplicaciones Cliente/Servidor.....	101
4.7 Desarrollo de Aplicaciones en ambiente MS Windows.....	102
4.7.1 Cómo se desarrolla una Aplicación Cliente/Servidor en un ambiente MS Windows.....	103
4.7.1.1 Creación de la interfaz de usuario.....	104
4.7.1.2 Estilos de interfaz de usuario.....	105
4.7.1.3 Desarrollo Interactivo.....	105
4.8 Desarrollo de una Aplicación Cliente/Servidor en VB.....	106

5. Herramientas para el Desarrollo de Aplicaciones Cliente/Servidor.

5.1 Introducción.....	109
5.2 Antecedentes.....	110
5.3 Categoría de Herramientas de Desarrollo.....	110
5.4 Herramientas "Front-End" (Cliente).....	111
5.5 Eligiendo el Correcto Front-End.....	112
5.6 Ambientes de Desarrollo.....	112
5.7 Herramientas CASE.....	112
5.8 Grúpa SQL Windows.....	113
5.9 C++.....	114
5.10 Lotus Notes.....	114
5.11 Microsoft SQL Server.....	115
5.12 VisualAge.....	115
5.13 Delphi Client/Server Suite.....	116
5.14 PowerBuilder Enterprise.....	117
5.15 Java.....	117
5.16 Visual Basic.....	118
5.17 Rendimiento General de Algunas Herramientas.....	119

6. Requisitos de Rendimiento y Seguridad de una Aplicación Cliente/Servidor.

6.1 Introducción.....	121
6.2 Requisitos de hardware para una Aplicación Cliente/Servidor.....	122
6.3 Requisitos técnicos que debe reunir una Aplicación Cliente/Servidor.....	124
6.4 Cinco niveles de clasificación de las organizaciones tomando en cuenta el grado en el cual han asimilado los procesos de análisis y Diseño de Aplicaciones.....	125
6.5 Confiabilidad de Windows NT.....	126
6.6 Medidas de rendimiento y seguridad de las Aplicaciones.....	127
6.6.1 El Rendimiento.....	128
6.6.2 La Seguridad.....	133
6.7 Seguridad en los datos que desplazan las Aplicaciones.....	135
6.8 Seguridad en las Aplicaciones Cliente/Servidor.....	136
6.8.1 Protección contra Virus.....	137

7. Control de los Costos en la ejecución de Aplicaciones Cliente/Servidor

7.1 Introducción.....	138
7.2 Presupuestos.....	139
7.3 Costos de hardware en Cliente/Servidor.....	140
7.4 Fórmula para calcular los costos de implantación de Aplicaciones.....	142
7.5 Control de Costos de Administración de Aplicaciones Cliente/Servidor.....	143
7.6 Control de los Costos de Ejecución de Aplicaciones Cliente/Servidor.....	144
7.6.1 Costo de la red.....	146
7.7 Relación Costo-Benéfico de las Aplicaciones Cliente/Servidor.....	147
7.8 El Futuro de las Aplicaciones Cliente/Servidor.....	148

CONCLUSIONES.

APÉNDICE A.

Ejemplo de una aplicación en Visual Basic Ver 5.0.

Documentos.

Código fuente de la aplicación.

GLOSARIO.

BIBLIOGRAFÍA.

CONVENCIONES

Este trabajo toma en cuenta que el lector está familiarizado con la programación en algún lenguaje de alto nivel bajo ambiente Windows. Un conocimiento que vaya de elemental a medio, acerca de Windows NT a la vez que ciertas nociones de tecnología de redes y de ingeniería de software, podrán ayudar al lector a comprender mejor las ideas que aquí se manejan.

En la actualidad dentro de las organizaciones existe una gran diversidad de versiones y tipos de sistemas operativos denominados Windows; mi trabajo se ha enfocado principalmente aquellos sistemas operativos de Microsoft que son reconocidos por que son considerados verdaderos sistemas operativos de 32 bits como lo es Windows NT es sus diferentes modalidades. A lo largo de este trabajo se aborda la versión 4.0 de Windows NT Server y Windows NT Workstation. En este mismo sentido hago la advertencia que la modalidad Windows NT Terminal Server no es tocada en ningún instante.

Durante el transcurso del desarrollo de este trabajo se habla ya de la muy próxima llegada de la siguiente evolución del sistema operativo Windows NT, cuando me refiera a este nuevo producto como Windows 2000 o como Windows NT 5.0 entiéndase que será indistintamente.

En algunas partes de este trabajo hago mención de "**Windows 9x**" se entenderá que hago referencia a Windows 98 o Windows 95 de forma indistinta, en cualquiera de sus versiones. Cuando se cite Windows me referiré en general al sistema operativo como tal abarcando Windows 95, Windows 98, Windows NT workstation ver. 4.0 y Windows NT Server ver. 4.0.

Se utilizó en este trabajo para pruebas, ejemplos y desarrollo de la aplicación práctica la herramienta MS Visual Basic Enterprise Edition Ver. 5.0.

A lo largo de este trabajo encontrará la palabra **organización** que es un ente que agrupa a un número determinado de recursos materiales y humanos en un espacio físico y que cuentan con un sistema de jerarquización, esta puede ser una compañía, una empresa, un conjunto de oficinas etc.

Debido a que este estudio gira en torno a las redes de área local (LAN), la palabra "red" simplemente, dentro de este trabajo equivale muchas veces a LAN. De igual manera ocurre con el término "manejador" que es utilizado como equivalente de DBMS según el contexto de la frase y el término "aplicación" que se refiere a aplicación Cliente/Servidor.

Pasando a la parte de presentación. Utilizo las palabras en negritas para las definiciones, para destacar términos y para diferenciar entre instrucciones u opciones de algún menú del texto del trabajo.

El uso de letra tipo bastardita es cuando desee acentuar el título de un párrafo o haga mención de ciertas tecnologías o sistemas que involucran el tema, para destacar puntos de sus definiciones

Para diferenciar los temas de los de los subtemas: Dentro de cada capítulo, los temas están en un recuadro con fondo gris y los subtemas son escritos con tipo de letra en negritas y ligeramente mayores.

A lo largo de este trabajo puede ser encontrada "C/S" que son las abreviaturas de Cliente/Servidor.

Se ha intentado en lo posible respetar nuestro idioma, aunque muchos de los términos aquí empleados le son ajenos puesto que son términos tecnológicos. Por lo que decidí emplear los términos más comunes sin ninguna traducción o interpretación como si fueran de origen del idioma español, por ejemplo los casos de software y hardware. En las ocasiones que no fue posible, por cuestiones de claridad, se emplearon los términos en su idioma original.

Limitaciones de esta Tesis

Este trabajo contempla el desarrollo de aplicaciones Cliente/Servidor en arquitecturas de hardware basadas en microprocesadores de Intel.

Este trabajo solo esta abocando la versión 4.0 de Windows NT de sistema operativo y a Visual Basic Versión 5.0 como herramienta de desarrollo para crear aplicaciones de 32 bits.

Este trabajo fue pensado en redes LAN alámbricas exclusivamente, se entiende el término alámbricas como todas aquellas que utilizan como enlace físico alambre de cobre para comunicarse. En ningún instante se contempla las redes conectadas por ondas de radiofrecuencias o por infrarrojo.

Este documento tampoco debe ser considerado como un manual de programación puesto que no pretende dar tips sobre instrucciones de alguno de los lenguajes de programación para diseño Cliente/Servidor que se tocan en su momento.

DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR PARA LAN BAJO SISTEMAS OPERATIVOS DE MICROSOFT DE 32 BITS.

Objetivo de esta Tesis

Este trabajo pretende ser una guía para auxiliar a todos aquellos profesionales de sistemas de información, que sin ser especialistas en esta área, están o estarán involucrados en el desarrollo de aplicaciones Cliente/Servidor en organizaciones relativamente pequeñas donde sus máquinas PCs están conectadas a través de una red LAN y cuya plataforma de sistema operativo es Microsoft Windows NT o que están en la transición de Windows 9x hacia Windows NT. Esta guía también puede ser aplicable a aquellos ambientes donde se pretende utilizar una máquina con gran capacidad de procesamiento, de memoria y de disco duro como servidor en el cual residirá el motor de base de datos y al cual se le conectarán PCs con Windows 9x o Windows NT Workstation versión 4.0 como clientes. Ya que las aplicaciones destinadas a trabajar en Windows NT generalmente funcionarán sin problemas bajo sistemas operativos menores como Windows 9x por lo que es posible aplicar esta guía a ambientes heterogéneos Windows.

El uso de los entornos de cómputo Cliente/Servidor son cada vez más frecuentes, y las demandas de estos por cada vez más y mejores aplicaciones también lo será y continuarán seguramente siéndolo en el futuro. El abaratamiento constante del hardware contribuye favorablemente a impulsar este fenómeno.

Se describirán como han evolucionado los entornos Cliente/Servidor, cómo se distinguen unos de otros, cuales son las partes constitutivas de estos.

Cuales son las técnicas de desarrollo Cliente/Servidor que es recomendable aplicar; cual es su justificación teórica y práctica. Y cuales son los elementos constitutivos de la metodología.

Recomendaré los requisitos de infraestructura necesarios para poder implantar las aplicaciones Cliente/Servidor.

Hablaré de cual es la mejor opción en tipos de redes LAN para implantar este tipo de entornos dentro de las organizaciones.

Abordare el tema de los costos que generará el implantar esta arquitectura con sistemas operativos de Microsoft.

Por último en este trabajo trataré los aspectos de rendimiento y seguridad que las aplicaciones Cliente/Servidor deben de brindar, cómo se ajustan estas medidas a las medidas ya existentes dentro de la organización o a las medidas tradicionales de seguridad y de rendimiento.

Componentes de los Sistemas Cliente/Servidor

1.1 Introducción

Definir los elementos que constituyen los entornos Cliente/Servidor surgió una vez que las computadoras ya no estaban aisladas, es decir después de que llegaron a ser necesidad básica para la operación de las organizaciones, las redes de cómputo. Gracias a las redes de cómputo las organizaciones podían explotar la posibilidad de compartir recursos de cómputo a todos o a determinados usuarios de la organización. Entonces surge otra necesidad, la de una organización o clasificación abstracta para simplificar su estudio y análisis. El primer paso para poder organizar, es definir los elementos que constituyen un ambiente tanto conceptual como funcionalmente.

Este capítulo tiene como finalidad dar a conocer y describir los elementos que conforman un entorno Cliente/Servidor, cuál fue su origen y cuál ha sido su evolución hasta el momento actual; pretende también que el lector obtenga una visión general sobre los diferentes modelos de entornos considerados Cliente/Servidor cuál ha sido la causa fundamental que los ha impulsado; e ilustraré los nombres técnicos que reciben cuando se encuentran trabajando en conjunto. También hablaré de cómo las interfaces gráficas de usuario son parte constitutiva fundamental de una aplicación Cliente/Servidor formal. Y expondré un ejemplo de una organización utilizando un entorno Cliente/Servidor.

1.2 Evolución de los esquemas de Cómputo Cliente/Servidor

Dependiendo de la plataforma de software instalada. Un entorno Cliente/Servidor es un tipo de cómputo distribuido y cómputo cooperativo a la vez, donde se establece un tipo de comunicación de igual a igual, conocido técnicamente como peer to peer o punto a punto, como es el caso de las redes Windows 95 simples donde la aplicación que se encuentra ejecutándose en el equipo servidor puede ser a su vez cliente de otra aplicación ejecutándose en otro servidor. Es posible distinguir las siguientes grandes etapas de evolución en este tipo de entornos.

1.2.1 Primer Etapa

Multiusuario

Los equipos **multiusuario** empezaron como un anfitrión con terminales tontas conectadas a ellos, aquí los equipos anfitriones contaban gran capacidad de cómputo. Y las terminales actuaban como terminales de entrada/salida sin capacidad de cómputo, dejando al anfitrión toda la carga de proceso y control de periféricos. Cada servicio tenía un grupo aislado de usuarios. Y el personal de sistemas se encargaba de consolidar o integrar la información cuando era necesario.

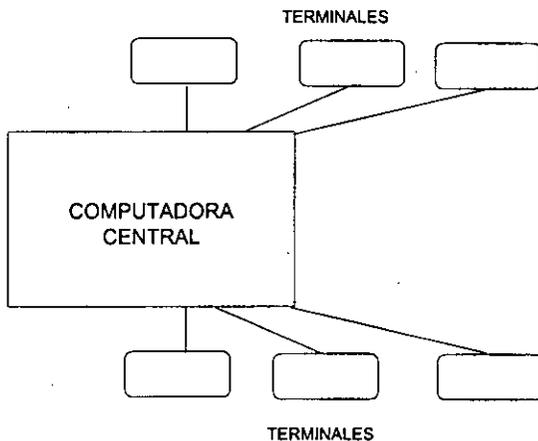


Fig. 1.1. Cómputo central.

1.2.2 Segunda Etapa

Maestro/esclavo

Los equipos **maestro/esclavo** conformaron una etapa posterior en la que la computadora maestra coordinaba el trabajo de los equipos esclavos, y estos a su vez se encargaban de coordinar los equipos periféricos. La ventaja es que descargaban el trabajo de entrada/salida de los equipos que realizaban la función de esclavos, de esta forma el equipo maestro podía dedicarse al procesamiento completamente.

Esta etapa tiene lugar con la aparición de computadoras un tanto más pequeñas y baratas que las convencionales en ese entonces. Pronto esto derivó en un esquema en el cual cada servicio empleaba su propia computadora a la cual se le conectaban directamente los usuarios correspondientes. Esta estructura que data de hace 20 años, tiene dos inconvenientes, el primero es que a medida que crece el número de personas que requieren acceso a la información en cada sistema, hay que emplear computadoras cada vez más poderosas en sus sistemas de entrada/salida; aunado a esto existe actualmente, igual como en el pasado, la estructura de ventas de los sistemas de información que están basados en el concepto de licencias para un número determinado de usuarios simultáneos y, al aumentar el número de usuarios necesariamente, el costo total de las licencias necesarias para emplear los programas, aumentará. El segundo es que estas computadoras no se comunican entre sí, por lo tanto no pueden compartir información, ni tampoco los usuarios de una máquina pueden acceder a las demás.

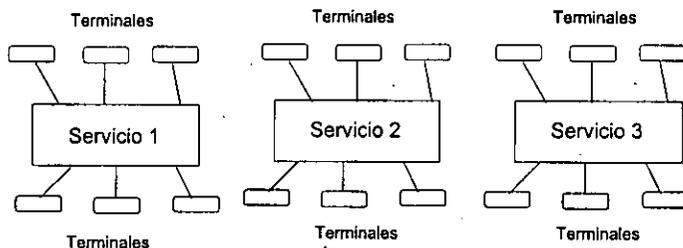


Fig. 1.2 Maestro/esclavo

1.2.3 Tercera Etapa

Conectándose libremente

Con su aparición las primeras computadoras personales permitieron que una fracción apreciable del trabajo de cómputo se llevara a cabo desde el escritorio del usuario, como son los cálculos o aspectos de presentación. En algunos casos el usuario obtiene la información de alguna computadora de servicio, empleando algún programa que emula algún tipo de terminal. Y en otros casos se les transfiere la información mediante medios magnéticos.

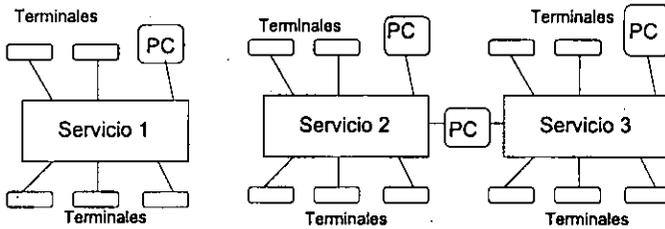


Fig. 1.3. Cómputo distribuido auxiliado por PC

1.2.4 Cuarta Etapa

Cómputo a través de redes

La comunicación de las computadoras en una red, permite planear un esquema totalmente distinto y mucho más productivo, en donde la información reside en una o varias computadoras para posibilitar a todos los usuarios que estén conectados a esa red el obtener información del resto de las demás computadoras, a la vez, que, también las diversas aplicaciones intercambian información de sus procesos. En este esquema la computadora personal de cada empleado le permite no solo realizar las funciones aisladas típicas como son manejar su procesador de textos, sus hojas de cálculo, diseñar bosquejos etc., sino también conectarse a otras computadoras dentro y fuera de la organización para obtener o distribuir información.

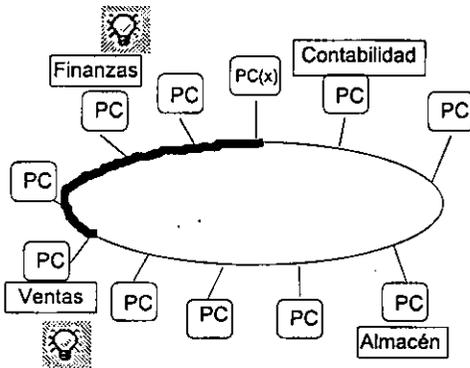


Fig. 1.4. Cómputo en red con PC. Muestra la conexión de una computadora PC(x) a una computadora PC de ventas y a una PC de finanzas.

1.2.5 Quinta Etapa

Cliente/Servidor

Esta es una etapa inmediata posterior a la del cómputo a través de redes. Esta etapa surge cuando sobre una red, cada uno de los servicios públicos que se ofrecen en las diversas máquinas se modifican, permitiendo que los programas reciban preguntas a través de la misma red. Y cada computadora que requiere obtener información de estos servicios se habilita para formular preguntas. Actualmente el no emplear este esquema sería destinar a las computadoras de los usuarios (equipos cliente) a solamente a cumplir funciones de terminales remotas, desperdiciando en gran medida su poder de cómputo individual.

Aquí cada usuario (computadora cliente) prepara una demanda de información a cualquiera de las computadoras que proporcionan información (computadora servidor). Esta información se transmite a través de la red es recibida y procesada por el servidor quien responde la demanda del cliente que realizó la solicitud. Los clientes y los servidores pueden estar conectados a una red local o a una red de área amplia.

En la etapa de **Cliente/Servidor**, el procesamiento de la información se divide entre el servidor y el cliente aprovechando mejor la capacidad de cómputo de los equipos de forma independiente.

Así cada usuario tiene la libertad de obtener información en un momento dado de fuentes locales o distantes ya procesadas o semiprocesadas, en este esquema no solo los clientes intercambian información sino que los servidores también pueden realizar esta acción.

Dentro de esta etapa podemos distinguir dos clasificaciones: *el entorno centralizado Cliente/Servidor* y *el entorno Cliente/Servidor distribuido*.

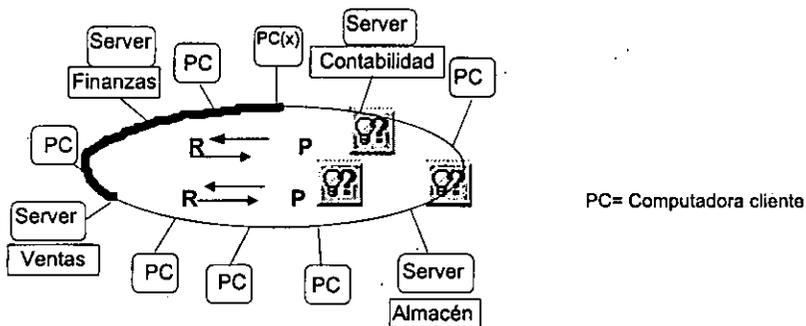


Fig. 1.5. Cómputo en Red Cliente/Servidor. Muestra como viajan las peticiones y las respuestas en un entorno Cliente/Servidor de los clientes a los servidores.

1.2.5.1 Entorno Centralizado Cliente/Servidor

Es aquel en el cual una aplicación Cliente/Servidor reside en una máquina central muy potente y a la cual se tiene acceso a través de terminales que son máquinas con pobre poder de cómputo. La terminal envía una petición de información a la maquina central esta obtiene el resultado y lo regresa a la terminal.

La petición de datos viaja en porciones del tamaño de un mensaje¹ a través de la red, desde el servidor hasta el cliente que requisito la información, esta actividad tiene lugar entre el sistema operativo de red, la tarjeta adaptadora de red y el cable. No hay mucha coordinación entre el cliente y el servidor para determinar que datos son necesarios obtener y/o enviar.

Inevitablemente en este esquema, la trasferencia de información entre los clientes y el servidor provocan gran tráfico sobre la red y hacen más lentas las peticiones de otros clientes, ya que ante una petición de información de una base de datos, el sistema mueve todos los datos y la negociación de los procesos a través de la red hasta el cliente y si esta negociación es amplia provocará lentitud en el tráfico de la red.

1.2.5.2 Entorno Cliente/Servidor distribuido

También conocido como simplemente Cliente/Servidor. Para que un entorno se considere Cliente/Servidor distribuido es necesario que se compartan las tareas de proceso de datos en el equipo cliente y en el equipo servidor, este último debe ser más potente o de igual capacidad que el equipo cliente pero no es recomendable que sea de menor capacidad.

Este entorno tiene gran aceptación en todo nivel jerárquico dentro de la organización sobre todo en aquellas organizaciones donde un número alto de personas requieren tener acceso constante a múltiples fuentes de datos.

Un entorno Cliente/Servidor es la forma más eficiente de conseguir acceso a bases de datos y administración de aplicaciones como las siguientes:

- Hojas de cálculo.
- Contabilidad.
- Comunicaciones.
- Administración de documentos.
- Almacenamiento centralizado de archivos.
- Administración de bases de datos (es la aplicación más utilizada en un entorno Cliente/Servidor).

¹ El tamaño de un mensaje se refiere al tamaño de un paquete de datos que es transmitido a través de la red por un protocolo de comunicación.

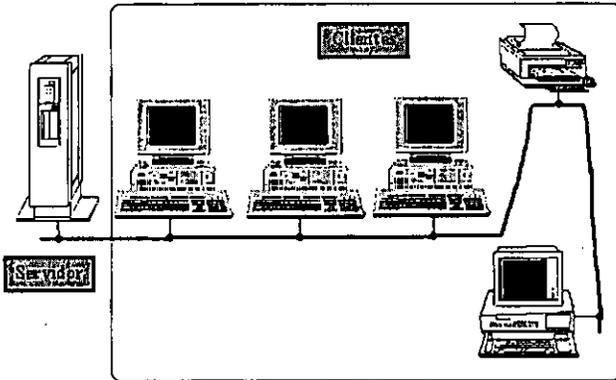


Fig. 1.6 Muestra una red Cliente/Servidor sencilla

Para finalizar mencionare que el avance tecnológico hace que cada vez se tenga más poder de cómputo en una PC de escritorio, en consecuencia hace que el entorno Cliente/Servidor sea cada vez más natural y popular.

1.2.6 Impulsos del Modelo Cliente/Servidor

El modelo Cliente/Servidor ha ido ganando terreno rápidamente gracias a desarrollos que se han dado en Internet, Sistemas Operativos, bases de datos y lenguajes de programación. A su vez gran impulso han dado principalmente, entre otros, los aspectos siguientes:

- El sistema operativo *Windows NT* que hace más sutil y transparente la migración hacia la arquitectura Cliente/Servidor.
- Los *Intranets* y el propio *Internet*, porque es la mejor manera de acceder a la información depositada en un servidor, ya sea próximo o distante, por su característica de navegador.

También debemos citar que este impulso se ve complementado por lo siguiente:

- *Data Warehouse* que fue diseñada para crear bases de datos, donde se almacena información analizada y sintetizada destinada para quienes toman decisiones.
- *Java*, aunque es un lenguaje de propósito general, tiene mucha orientación para desarrollos de Internet, que hace posible realizar aplicaciones independientes del sistema operativo.
- *Visual Basic* como lenguaje también de propósito general, Pero con una cualidad que lo hace sobresalir de los demás debido a que puede proporcionar acceso directo a las APIs de Windows.

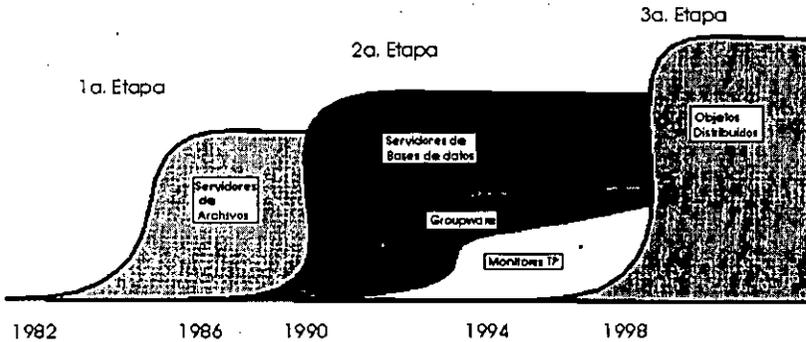


Fig. 1.7 Etapas de Cliente/Servidor. Las bases de datos, los monitores TP y el groupware han comenzado a desplazar a los servidores de archivos, esto ocurre en la era de Ethernet. Sin embargo. Poco a poco se introduce la era de redes de alta velocidad. La era de Ethernet es donde las aplicaciones son enfocadas a procesar archivos para Cliente/Servidor, después viene una ola de aplicaciones enfocadas en bases de datos, los monitores TP y el groupware generaron olas pequeñas y los objetos distribuidos son la próxima gran ola. (Fuente: revista Byte México, junio 1995 artículo: Computación Intergaláctica c/s).

Tecnología de servidores que contribuyen al impulso Cliente/Servidor en los 90s

1.2.6.1 Monitores TP

Los monitores de TP (también conocidos como monitores transaccionales), administran los procesos y orquestan los programas dividiendo las aplicaciones complejas en partes de código llamadas transacciones, estas realmente no son muy usadas en las redes Ethernet, sino que se espera que tengan un gran auge en las próximas tecnologías de redes, como lo son las redes de alta velocidad, con máquinas dedicadas a estas transacciones exclusivamente y repartidas a lo largo de las redes. Las transacciones se han convertido en una filosofía de diseño de aplicaciones que realmente cumplen eficientemente su función dentro de los ambientes distribuidos Cliente/Servidor donde se utiliza un monitor de transacciones. Una transacción puede ser administrada desde su origen, es decir, la transacción viaja desde un cliente pasa a través de uno o varios servidores y regresa los resultados al cliente que realizó la transacción, una vez terminada la transacción todas las partes que intervinieron muestran los resultados al cliente, si se realizó, si tuvo o no tuvo éxito su transacción en cada uno de los servidores que utilizó y por último reúne los resultados. Cuando se utiliza un monitor TP todos los programas participantes deben respetar las reglas de la disciplina transaccional, ya que de otra forma un solo programa defectuoso puede afectar a todo el proceso.

Los procesos de recuperación de datos por medios de transacciones son el ideal dentro de los entornos Cliente/Servidor. Sin embargo. Solo son eso ya que la mayoría de las redes (sobre todo las LAN) aun no están preparadas para soportarlas.

Los monitores de transacciones definen cuándo comienza una transacción, cuándo termina y cuáles serán las unidades de recuperación en caso de que ocurra una falla. Actualmente sólo el modelo de transacciones planas es el más utilizado dentro de la actual generación de monitores TP. En una transacción plana no hay forma de suspender una transacción. Es decir. Se realiza toda la transacción o no se realiza en lo absoluto. Una transacción empieza con un **"begin_transaction"** y termina con un **"commit_transaction"**, y no existe forma de realizar solo partes de la transacción.

Los monitores de transacciones fueron inventados para ejecutar aplicaciones que atiendan a miles de estaciones cliente, no importando los servidores y clientes que se encuentren en su paso. Son capaces de administrar las transacciones, dirigirlos a través de los sistemas, nivelar su ejecución y reiniciarlas tras un fallo, también son capaces de administrar recursos transaccionales en un solo servidor por medio de servidores múltiples, además de que pueden cooperar con otros monitores de transacciones también pueden encausar y manejar una gran cantidad de estaciones cliente ayudando al sistema operativo y a los administradores de recursos en el servidor.

Realizar una aplicación en un solo servidor para una red LAN pequeña utilizando monitores de TP tal vez sea demasiado sobrado. Esta es una de las razones por las cuales no han tenido el éxito deseado los monitores transaccionales, además de que los distribuidores no han llegado aún al mercado de software para empresas pequeñas, donde aún no se ven las verdaderas ventajas de los monitores transaccionales.

1.2.6.2 Groupware

El groupware esta formado por cinco tecnologías que son aplicables cuando se trabaja en grupo:

- Administración de documentos de multimedia.
- Flujo de trabajo.
- Correo electrónico.
- Conferencias.
- Programación correo de agenda.

Groupware es una tecnología de macrocomputadoras aplicada a las redes LAN de PCs, en otras palabras es tecnología downsizing o minimizada. Se trata de un modelo nuevo para el cómputo Cliente/Servidor, es una forma de juntar y organizar datos no estructurados en forma de documentos, como son: texto, imágenes, faxes, correo y conferencias en línea, permitiendo a los usuarios visualizar esos documentos, almacenarlos, replicarlos y dirigirlos a cualquier parte de la red. Una forma de poder comprender más la función que desempeña un groupware es la siguiente un documento para un groupware es lo mismo que una tabla para un SQL. El groupware es óptimo para la administración de bases de datos de documentos y hace efectivo el uso del correo electrónico.

Lotus Notes es considerado el principal producto de groupware actualmente, es debido al uso efectivo del correo electrónico y por ser capaz de administrar bases de documentos en forma de Cliente/Servidor. Lotus Notes no es recomendable cuando se necesite concurrencia o actualizaciones inmediatas.

1.2.6.3 Objetos Distribuidos.

La tecnología de objetos distribuidos proporcionará mayor flexibilidad en sistemas Cliente/Servidor, debido a que encapsula datos y lógica comercial en objetos que pueden navegar en cualquier sitio de la red y ejecutarse en plataformas distintas, esta tecnología esta más adelantada que los demás enfoques Cliente/Servidor.

Una vez que la tecnología de objetos distribuidos se extienda, puede absorber a todos los demás tipos de cómputo Cliente/Servidor, incluyendo monitores de TP, bases de datos SQL y groupware.

Una ventaja es que los objetos distribuidos pueden ayudarnos a dividir grandes aplicaciones monolíticas en componentes maleables que existan en el software abierto de objetos.

1.3 Que es una Aplicación

El usuario es el destino final de toda aplicación. Por lo que inicio este apartado con una pregunta interesante: ¿Que es lo que un usuario² entiende por la palabra "aplicación"? Debido a que es un término que muchas veces provoca discusión. Aplicación es un término que tiene diferentes acepciones, pero la definición aceptada de *aplicación* en el ámbito del diseño de software, es aquella que dice lo siguiente: **"Una aplicación es un programa de computadora que resuelve una necesidad concreta y específica"**.

Otra definición que más adelante tomará forma y sentido desde un punto de vista meramente técnico será aquella que dice que una "aplicación es un conjunto de uno o más componentes que incluye al menos un proceso y que puede contar también con una interfaz gráfica de usuario", utiliza diversos componentes externos, que puede admitir múltiples usuarios concurrentes, distribuir el trabajo entre equipos diferentes y utilizar uno o más orígenes de datos externos.

Por lo tanto, una aplicación también puede ser descrita en cuanto a componentes u objetos cada uno de los cuales ofrece uno o varios servicios o hace uso de ellos o ambas cosas a la vez.

Por último. "Aplicación" también se utiliza para designar al "software de aplicación". Es decir, aquellos programas destinados para un grupo de personas o empresas que dan solución a determinados ramos, por ejemplo procesadores de texto, hojas de calculo, etc.

Muchas veces los usuarios y aún las personas muy relacionadas con el medio utilizan aplicación como sinónimo de sistema y viceversa aunque en el estricto punto de la definición un sistema es la interrelación de varias aplicaciones. No es lo mismo referirse a varias aplicaciones trabajando en conjunto que a una sola de ellos.

1.4 Que son las GUI

GUI son las siglas de "Interface Graphical User" o interface gráfica de usuario. Es la parte de la aplicación destinada a presentar y recabar los datos ingresados por el usuario. Generalmente una aplicación es juzgada por su GUI, ya que esta presenta siempre la primera impresión para el usuario y de esta muchas ocasiones depende la rapidez con que aprende a utilizar y dominar la aplicación y como consecuencia obtener el mejor provecho de la misma.

El GUI es en términos llanos, aquello que despliega sobre la pantalla del monitor una aplicación en ejecución, con todas sus características como son colores, ventanas, botones y menús que el usuario puede utilizar o disfrutar visualmente mediante los cuales puede auxiliarse para formular y escribir sus

² Aquí, me refiero a usuarios con un nivel de conocimiento que va de medio a avanzado en cultura computacional.

peticiones. Las GUIs totalmente gráficas son las más vistosas, alegres y amigables para el usuario y tienen la ventaja de que pueden casi mostrar cualquier cosa sobre la pantalla sin tener la limitación de ningún formato o código de caracteres, como sucede con las interfaces de terminales basadas en texto monocromático.

1.4.1 Network User Interface³

Las Network User Interface (NUI) es una evolución de las GUIs. Puede verse como la combinación de GUIs donde interviene fuertemente las conexiones a redes. La justificación de estas nuevas interfaces gráficas de usuario se basa en el hecho de la tremenda penetración de internet y a que una cantidad de servicios crecientes se encontrarán disponibles a través de las redes dentro de las organizaciones. Se espera que las "NUIs" mejoren no solamente la forma en que los usuarios interactúan con las computadoras sino también con las redes. En lugar de manipular representaciones directas de unidades de disco y directorios los usuarios navegan en los recursos locales y remotos con hiperligas mediante herramientas parecidas a los navegadores. Las "NUIs" hacen más fácil que los departamentos de sistemas personalicen y administren escritorios en toda la organización. Además de que todas las "NUIs" pueden ser almacenadas en el disco del servidor para que el estado del escritorio de cada PC (esto también es conocido como definición de perfiles de escritorio, el cual es parte de los "perfiles de usuario") sea siempre el mismo no importando desde que máquina cliente acceda al servidor.

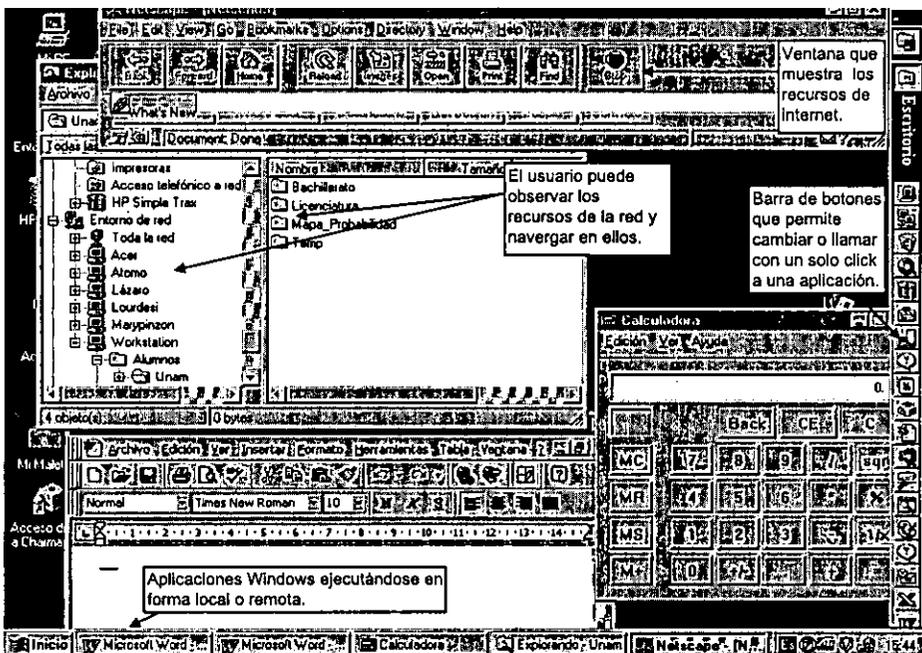


Fig. 1.8. Ejemplo, simulación de un NUI.

³ Definidas así según artículo de portada de la revista Byte México año 9 No. 114

1.5 Concepto del Front-End

Un front-end puede definirse como aquella parte de la aplicación que es capaz de desencadenar que el usuario genere una petición en el cliente, para luego el cliente continúe con la ejecución de la aplicación que realiza lo siguiente:

1. Presenta una interface de usuario.
2. Da formato a la petición del usuario.
3. Presenta los datos que recibe desde el servidor.

El equipo cliente a través de su GUI acepta instrucciones del usuario, las prepara para el servidor y después envía una petición de información específica por la red hacia el servidor, este procesa la petición y envía la información-respuesta de regreso al cliente. El cliente inserta esta información en su interfaz para presentarla al usuario.

De acuerdo a la combinación Cliente/Servidor más conocida como lo veremos en el capítulo cuatro. En el servidor no reside la interfaz de usuario, esta reside en el cliente, puesto que es el responsable de dar presentación y forma a los datos que recibe del servidor para que estos sean mejor entendidos por el usuario. Esto lo realiza a través del uso de una o varias, interfaces gráficas de usuario y herramientas de escritura de informes.

1.6 Concepto del Back-End

El término Back-End se aplica a aquel servidor que es el responsable de albergar al DBMS. En palabras simples. Es el elemento que recibe las peticiones de los clientes, las procesa y envía de regreso el resultado al cliente que generó la solicitud.

Ejemplo de un entorno Cliente/Servidor dentro de una organización:

En una empresa de fabricación y distribución, toda la información de clientes y de productos se almacena en una base de datos, esta información puede recuperarse y presentarse de diversas formas dependiendo del área de la empresa que la solicite.

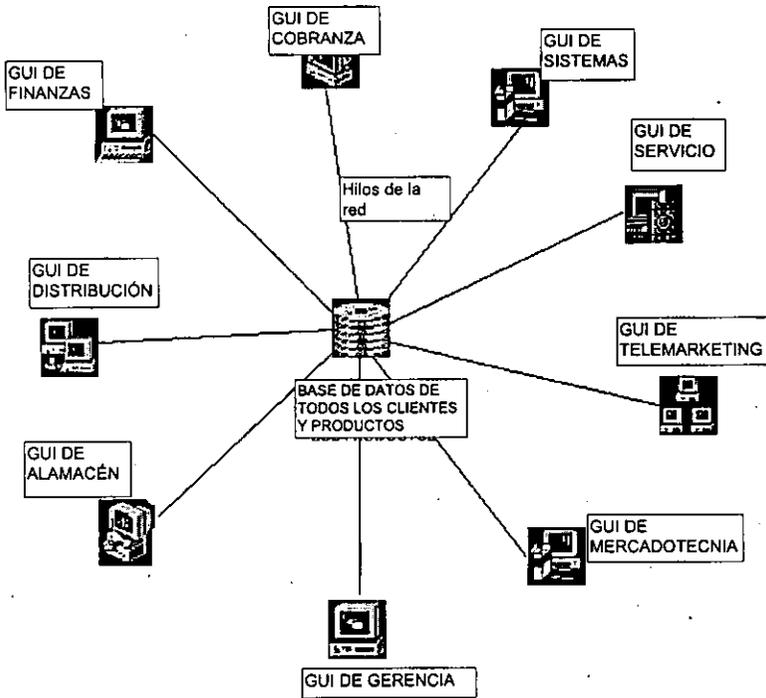


Fig. 1.9. Ejemplo de un ambiente Cliente/Servidor.

- En mercadotecnia pueden enviar información promocional a los clientes en una área determinada de la ciudad.
- El almacén puede registrar las entradas de nuevos productos así, como los que son dados de baja
- El área de distribución y pedidos puede encontrar y enviar los productos que se encuentren en existencia en el almacén.
- El área de servicio al consultar la base de datos, puede saber que clientes requieren de servicio.
- Soporte técnico sabe que clientes recurren a soporte técnico con más frecuencia y sobre que aspecto requieren más soporte.
- Telemarketing con regularidad llama a los clientes para conocer y almacenar necesidades de sus clientes.
- Finanzas puede saber el estado actual de los recursos financieros de la empresa.
- Cobranza mediante la consulta a la base de datos puede dejar de aceptar pedidos de aquellos clientes atrasados en sus pagos.

Debido a las diferencias radicales entre cada área de la empresa necesitan una interfaz gráfica de usuario prediseñada que se ajuste a las necesidades particulares de cada área. No obstante, que todas las áreas de la empresa tienen en común la misma base de datos.

1.7 Front-End y Back-End Trabajando Juntos

En el entorno Cliente/Servidor hay dos componentes lógicos principales:

1. La aplicación, que a menudo se conoce como cliente o "Front-End".
2. El servidor donde reside la base de datos es conocido como servidor o "Back-End"

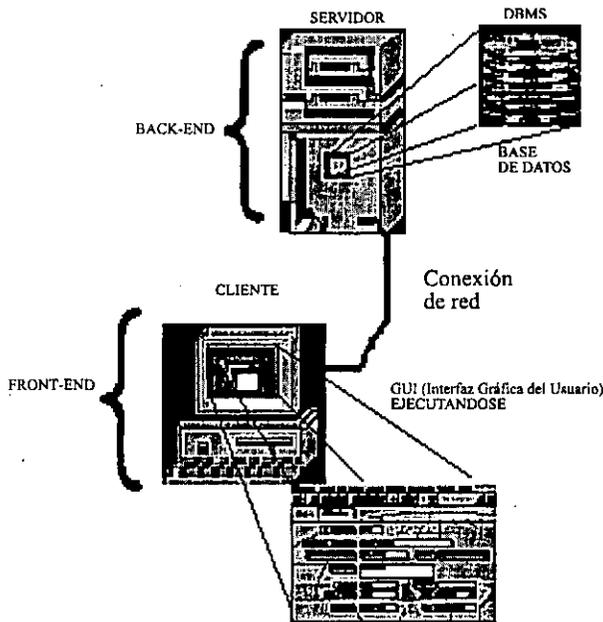


Fig. 1.10. Front-End y Back-End

Arquitectura Cliente/Servidor

2.1 Introducción

Cuando nos enfrentamos a desarrollar un proyecto que involucre aplicaciones Cliente/Servidor debemos de conocer en primer instancia cuál es la infraestructura con la que contamos actualmente y cuáles serán los objetivos que deseamos alcanzar y cuales serán los beneficios de implantar este entorno dentro de una organización.

En este se capítulo describirá los elementos que constituyen los entornos Cliente/Servidor, como están definidos, cuales son los requisitos de cada uno de estos para que se consideren constitutivos de un entorno Cliente/Servidor. Abordaré también las preguntas más frecuentes que surgen sobre los beneficios y desventajas de adoptar los entornos Cliente/Servidor dentro de las organizaciones. Los diferentes tipos de servidores que pueden ser implementados en estos entornos de cómputo. Así como también hablare del por que las organizaciones son orilladas a adoptar estos entornos en harás de

mantenerse competitivas en su mercado.

2.2 Que es el modelo Cliente/Servidor

Una arquitectura es un conjunto de definiciones, reglas y términos que se emplean para construir un producto. Algunas arquitecturas son empleadas por varios productos (estas también son conocidas como arquitecturas abiertas), otras son empleadas solamente por sus creadores (son conocidas también como propietarias).

Un importante punto en el diseño de una aplicación Cliente/Servidor es determinar la arquitectura de la aplicación y definir la infraestructura destacada. En la práctica, la arquitectura es el arreglo de la plataforma cliente, la plataforma servidor y la comunicación de la red más el entorno de funcionalidad y datos (presentación, aplicación y componentes de bases de datos) que existe entre las plataformas del cliente y servidor. La infraestructura es el hardware y el software requerido para apoyar y soportar esa arquitectura (por ejemplo: rutinas de soporte, sistema operativo, base de datos, controladores de tarjetas de red y de enlace).

Hay varias formas en las cuales se puede definir la arquitectura de una aplicación Cliente/Servidor, las diferencias entre esas variedades esta en como el GUI, la aplicación y los componentes del sistema de base de datos son distribuidos entre las plataformas cliente y servidor.

Debemos destacar que en la practica es difícil crear una definición única de lo que es un modelo o un entorno Cliente/Servidor sobre una red LAN. Debido a la diversidad de combinaciones que se dan dentro de las organizaciones. Por ejemplo:

- Se dice que "varias terminales conectadas a un mainframe es modelo Cliente/Servidor".
- Se dice que "varias computadoras personales conectadas a un servidor también es Cliente/Servidor".
- Se dice también que "Terminales y computadoras personales conectadas a servidores es Cliente/Servidor también".

Una definición simple se da partiendo desde un punto de vista genérico y por lo tanto expresa la definición más rudimentaria y común. Una definición posterior resulta de la síntesis de las diversas combinaciones encontradas dentro de las organizaciones.

"Un modelo Cliente/Servidor es aquel ambiente de cómputo donde el cliente es un modulo de hardware o software el cual hace solicitudes y el servidor es un modulo de hardware o software el cual da respuesta a estas solicitudes".

2.3 Cuáles la función primordial de la arquitectura Cliente/Servidor.

En organizaciones de todo tipo a través del tiempo se han adquirido equipos de todos tipos, tamaños, modelos y capacidad de procesamiento esto se debió principalmente a que las organizaciones necesitaban facilitar su trabajo y mejorar sus comunicaciones, posteriormente se requirió de poder compartir la información de una manera más fácil y rápida. Y en otros casos algunas organizaciones adquirieron y siguen adquiriendo software para desarrollar diferentes aplicaciones a fin de mejorar la productividad de sus propias organizaciones. Estos dos fenómenos han hecho que las organizaciones tengan un número impresionante de equipos de hardware, un número impresionante de software de aplicación junto con una gran cantidad aplicaciones desarrolladas, provocando un consecuente aumento en el TCO de las organizaciones en hardware y software.

Es principalmente por esta razón que las organizaciones empezaron a poner sus expectativas en soluciones que mejorarán el acceso a la información pero que a la vez mantuvieran la cualidad de ser altamente tolerantes a los cambios dentro de la organización, que contribuyan a reducir los costos de operación y a aumentar la funcionalidad de las aplicaciones existentes y futuras. La solución actual que cubre todas estas expectativas es el entorno de cómputo Cliente/Servidor.

El desarrollo de la tendencia Cliente/servidor persigue que los usuarios tengan más accesibles los datos. Para esto se desarrolla software que permite que un usuario tenga acceso a la información indistintamente del tipo de computadora que utilice; también permite que cualquier computadora actúe como cliente o como servidor en un momento dado. En un modelo Cliente/Servidor el servidor almacena y distribuye tanto información como aplicaciones. De esta forma. Las organizaciones tienen la posibilidad de obtener el control de las aplicaciones para tener un mejor tiempo de la respuesta de sus transacciones, con el modelo Cliente/Servidor también se persigue una descentralización de las aplicaciones de la organización para ser implantadas más cerca de los usuarios demandantes. Así también los departamentos de sistemas de las grandes compañías donde están específicamente bien delineadas las funciones de cada sección se ven transformadas sus funciones, pasan de ser solamente administrador de recursos a ser proveedor de servicios de cómputo.

2.3.1 ¿Por qué las organizaciones migran hacia una Arquitectura Cliente/Servidor?

Existen dos causas fundamentales

1. Por la necesidad de disminuir sus costos.
2. Por la necesidad de obtener competitividad en el mercado.

Al migrar a una plataforma Cliente/Servidor se obtienen los siguientes beneficios:

- Reducción en los costos de procesamiento.
- Reducción en los costos de desarrollo de aplicaciones

2.3.1.1 Reducción en los costos de procesamiento.

Mediante la adquisición de equipos de escritorio de gran capacidad de procesamiento, en lugar de comprar un equipo mainframe, se abaten los costos en un rango de que va de un 10% a un 20%⁴ obteniendo la misma capacidad de procesamiento.

2.3.1.2 Reducción en los costos de desarrollo de aplicaciones.

Mediante la utilización de sistemas operativos de 32 bits más poderosos tales como Windows NT o UNIX, que proporcionan capacidades de alto desempeño entre las más destacadas destinadas a aplicaciones Cliente/Servidor se encuentran el desarrollo gráfico y las capacidades multitarea.

También se abaten gastos a la vez que se genera una mayor productividad; desarrollando y liberando sus aplicaciones en menor tiempo, a través del uso de tecnología estándar las organizaciones se mantienen competitivas a la vez que se colocan en un punto donde es posible una migración posterior aunado a que pueden adoptar tecnologías mucho más rápido y a menores costos.

2.4 Concepto de Cliente

Para realizar una solicitud en un sistema autónomo (local), el usuario escribe un comando para que el equipo realice una tarea, la petición viaja por el bus local de la computadora hasta el CPU, el CPU interpreta la petición a través de su sistema operativo y realiza la petición. No ocurre así cuando se trabaja en un entorno de red Cliente/Servidor, ya que aquí cuando un usuario inicia una petición para usar un recurso que se encuentre en un servidor en alguna parte de la red, la petición tiene que reenviarse fuera del bus local de la computadora hasta el servidor donde se encuentra el recurso.

El cliente es el elemento que interactúa con el usuario, por lo que el cliente debe de poseer capacidad de procesamiento y es preferible que cuente con una interfaz gráfica fácil de usar.

El cliente es el equipo solicitante dentro de una red de comunicaciones, este recibe y utiliza la información corporativa que necesita en el momento en que la solicita.

El cliente es la combinación de software y hardware que invoca los servicios de uno o de varios servidores, e incluso de otro cliente.

⁴ Según estudios de grupo Scanda, empresa en México dedicada a consultoría y a la implementación de soluciones Cliente/Servidor.

2.4.1 Tipos de plataformas Cliente

En la práctica las plataformas cliente generalmente cae dentro de una de las siguientes categorías:

Terminal Cliente [P]

Una terminal cliente contiene solamente un componente de presentación, por ejemplo un X-Windows workstation. Este puede ser solamente un dispositivo de despliegue (por ejemplo un scoreboard o la pantalla de salida de los vuelos en el aeropuerto) o un dispositivo solamente de entrada (por ejemplo un lector de pases de seguridad o un sensor de temperatura).

Plataforma Cliente Personal [PA] o [P]

Una plataforma personal es una PC o una workstation con generalmente un GUI, y posiblemente una aplicación o aplicaciones usadas por uno solo usuario (humano). Este contendrá un componente de presentación que es generalmente un componente de la aplicación.

Controlador de Máquina [PA]

Este tipo de cliente contiene alguna forma externa de interface (por ejemplo sensores y actuadores) agregan esto a algunas aplicaciones para funcionalmente controlar o monitorear la interface. Por ejemplo una estación externa de un sistema de control. Este contendrá un componente de presentación (la interface) y un componente de aplicación.

2.4.2 Redirector

Este elemento es el encargado de reenviar las peticiones de una máquina cliente, el redirector depende del software de red, es conocido por su función dentro del cliente como interprete de comandos para la red. El software que se utilice en una red para administrar sus recursos de E/S debe de ser un sistema operativo para red, por lo que el redirector será una pequeña sección del sistema operativo de la red que tiene la responsabilidad de desempeñar las siguientes tareas:

- Intercepta las peticiones en el equipo cliente.
- Valida si las peticiones del equipo cliente son peticiones locales o para la red.

En Windows NT, el servidor procesa las conexiones requeridas por los redirectores clientes y les ofrece acceso a los recursos que solicitan. En otras palabras, el servidor atiende o satisface la petición realizada por el cliente. Como se ilustra enseguida.

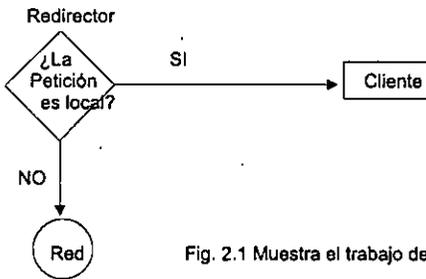


Fig. 2.1 Muestra el trabajo del redirector.

El redirector se encarga también de realizar un rastreo de los nombres de las máquinas asociadas con los recursos. En el caso de Windows NT es necesario tener acceso a un directorio compartido para que pueda hacer uso de los recursos de la red, esta operación de compartir la puede realizar sencillamente utilizando el "Explorador de Windows NT" que es un accesorio parte del sistema operativo de red.

2.5 Concepto de Servidor

El servidor es la unión de software y hardware que responde a las peticiones de los clientes. La popularidad de los microprocesadores Intel hace que la mayoría de los servidores de las redes LAN estén basados en microprocesadores Intel.

En un entorno Cliente/Servidor el servidor esta dedicado al almacenamiento y administración de datos, en el servidor es donde tiene lugar la mayor parte de la actividad de la base de datos, el servidor en este tipo de entornos es conocido como "back-end". Este recibe las peticiones estructuradas de los clientes, procesa las peticiones y regresa la información solicitada al cliente.

El software del servidor reacciona a las consultas del cliente ejecutando búsquedas y solo se concreta a la realización de la búsqueda, un proceso posterior implica la ordenación, la extracción de los datos solicitados y el envío de los datos al usuario.

Por otra parte el software del servidor también administra los datos de la base de datos cuyas acciones comprenden:

- Actualizaciones.
- Eliminaciones.
- Inserciones.
- Protecciones. (Respaldos, seguridad etc.)

Un servidor en este tipo de entornos deberá de ser el equipo de más potencia y capacidad que el de los equipos clientes, este equipo desempeñará las tareas siguientes:

- Múltiples tareas simultáneas.
- Esquemas de seguridad.

- Tareas de administración de red.

Es recomendable que en una red donde se pretenda incorporar un entorno cliente/servidor el servidor sea dedicado para controlar las funciones de postproceso.

El servidor también almacena y distribuye tanto información como aplicaciones en algunos casos donde las organizaciones son en extremo reducidas.

Muchos de los servidores para redes LAN basados en procesadores Intel vienen con Microsoft Windows 9x preinstalado, esto significa que con este sistema operativo solo son máquinas personales. Y pueden constituir una red LAN pequeña con pocas opciones de seguridad y una pobre administración. Habría que instalar un sistema operativo de red más robusto o un sistema operativo para trabajo en grupo o requisitar servidores con Windows NT Server preinstalado, a fin de que sean aprovechados como servidores.

2.5.1 Tipos de Servidor

- Servidores de Bases de Datos (La mayoría).
- Servidores de Archivos.
- Servidores de Datos.
- Servidores de Cómputo.
- Servidores de Aplicaciones.
- Servidores de Comunicaciones.
- Servidores de Correo.
- Servidores de Impresión.

Servidor de archivos

Es aquel que permite compartir archivos en un grupo de trabajo. Cuando un cliente hace solicitud de un archivo el servidor envía una copia completa al cliente, incluyendo los índices del archivo. En este proceso se bloquea el archivo hasta que el cliente que hizo la solicitud libera el archivo.

Servidor de datos

Los servidores de datos no efectúan ningún tipo de procesamiento de la aplicación solo realizan validación de datos y se encargan de proporcionar datos a los **servidores de cómputo**, los servidores de cómputo son los que ejecutan la aplicación. Por tal razón los servidores de datos y los servidores de cómputo se utilizan siempre en conjunto su ventaja es que se pueden tener dos servidores por separado. En uno de ellos los datos exclusivamente y en el más potente las aplicaciones.

Servidor de Aplicaciones

Un servidor de aplicaciones es aquel que realiza tanto el manejo de los datos, como el procesamiento de las aplicaciones, en otras palabras es un servidor de cómputo y un servidor de datos en la misma computadora.

Servidor de bases de datos

Es el servidor más común, el servidor de bases de datos es la computadora en la cual se ejecuta el manejador de bases de datos y donde se efectúan las funciones de modificación, actualización, consulta, verificación de integridad, referencia y administración de recursos.

Servidores de comunicaciones

Es una computadora con software específico para comunicaciones y su función es la de controlar el acceso, seguridad enrutamiento y puenteo del tráfico de la red, hacia afuera de la organización o dentro de esta hacia otras partes de la red.

Servidores de correo

Es la computadora configurada de tal forma que es capaz de administrar mensajes en cuentas de usuario, este es asíncrono por naturaleza.

Servidores de impresión

Como su nombre lo indica son equipos especializados en recibir y procesar solicitudes de impresión de documentos. Son los responsables de administrar todas las solicitudes de impresión. Generalmente estos servidores son declarados con las mismas herramientas que trae consigo el sistema operativo, por lo que no se requiere de realizar inversiones posteriores en la adquisición de software para implementar esta función.

2.5.2 Características de hardware de la computadora servidor.

El servidor es una máquina con características sobresalientes a las demás computadoras de nuestra red, este tipo de computadoras están fabricadas pensando que tendrán un trabajo continuo durante las 24 hrs. del día los 365 días de año, estas computadoras cuentan con la capacidad modular más avanzada de todos los equipos de cómputo, además cuentan con los sistemas más sofisticados de seguridad y de redundancia de componentes de hardware junto con sistemas avanzados de predicción de fallas. Por lo que sus partes electrónicas serán de otras construcciones comparadas con las partes de una PC normal de escritorio.

Contar con la característica multiprocesador es otra de las cualidades principales también de un servidor. Esta característica consiste en la posibilidad de aceptar más de un microprocesador en la tarjeta principal de la computadora, los procesadores se agregarán en función de la demanda de procesamiento por parte de las aplicaciones.

Los elementos principales a contemplar en un servidor para que pueda funcionar óptimamente y sea capaz de albergar adecuadamente el sistema operativo de red (para este caso MS Windows NT Server 4.0) son los siguientes:

- Espacio suficiente disponible en disco, para albergar al SO.
- Tipo de procesador y capacidad de este, compatibles con MS Windows NT Server Ver. 4.0.⁵

⁵ Puede checar en la lista de compatibilidad de hardware (HCL) que marcas de servidores son compatibles con Windows NT, en la dirección <http://www.microsoft.com/hcl/default.asp?RLD=34>, esta se actualiza regularmente.

- Memoria RAM
- Tipo de sistemas de archivos.
- Clientes.

En la siguiente tabla se muestran un ejemplo de requisitos de hardware para albergar Windows NT Server ver 4.0

Categoría	Requisitos para microprocesadores Intel
Procesador	x86 de 32 bits o superior (486/33MHz o superior).
Monitor	VGA o de mayor resolución.
Espacio en Disco Duro	Uno o más discos duros, con un mínimo de espacio de 125MB en la partición que vaya a contener los archivos de Windows NT Server.
Otros Controladores	Unidad de disco Flexible de 3.5" de alta densidad, Unidad SCSI o IDE de CD-ROM.
Memoria	16 MB mínimo.
Componentes adicionales	Mouse u otro dispositivo apuntador.
Otros componentes	Una o más tarjetas adaptadoras de red.

2.6 Concepto de Red

Es el elemento físico por el cual se transmiten los requerimientos del cliente al servidor y viceversa. También la red mediante el SO proporciona mecanismos de seguridad y monitoreo. Estos mecanismos son transparentes para el usuario, es gracias a la red que el cliente obtiene los datos y la información como si se generarán localmente.

La red es el hardware y el software de comunicaciones que enlaza a los clientes con los servidores, dependiendo de su cobertura estas se clasificarán en redes de área local (LAN), redes de área metropolitana (MAN), y redes de área amplia (WAN). Como es sabido este estudio se enfoca en el aspecto de red LAN. Estos tres tipos de redes se diferencian por la distancia que pueden cubrir, la tecnología de comunicaciones, los canales de comunicación que pueden usar y la velocidad de operación. Para las redes LAN se puede utilizar la tecnología Ethernet con velocidades de 10Mbps o 100Mbps, Token Ring con velocidades de 4 o 16 Mbps, FDDI con velocidades de 100Mbps, CDDI con velocidades de 100 Mbps y ATM con velocidades de 44,736 Mbps.

En una red se pueden emplear diversos protocolos de comunicación. Por ejemplo, en una red Ethernet se pueden emplear los protocolos TCP/IP, NetBios, NetBeui o IPX/SPX. Para una red LAN pequeña puede resultar suficiente y adecuado utilizar IPX/SPX o Netbeui, Mientras que para una red LAN compleja con muchos servidores y subredes puede resultar más adecuado utilizar TPC/IP.

Un requisito indispensable es que una red debe de ser un medio eficiente y confiable para que la comunicación entre los clientes y los servidores no sufra cuellos de botella. La arquitectura Cliente/Servidor ha influido en la tecnología de redes de tal forma que los principales fabricantes de equipos de red ofrecen productos especiales para entornos Cliente/Servidor como son

concentradores conmutados (switching hubs) o redes virtuales (virtual networks) que es la nueva tendencia en productos de redes.

2.6.1 Tipos de Arquitecturas del Modelo Cliente/Servidor

Las dos configuraciones Cliente/Servidor más importantes son:

- Donde los datos pueden estar situados en un único servidor.
- Donde los datos pueden distribuirse en varios servidores de bases de datos, esto dependerá de la ubicación de los usuarios y de la naturaleza de los datos.

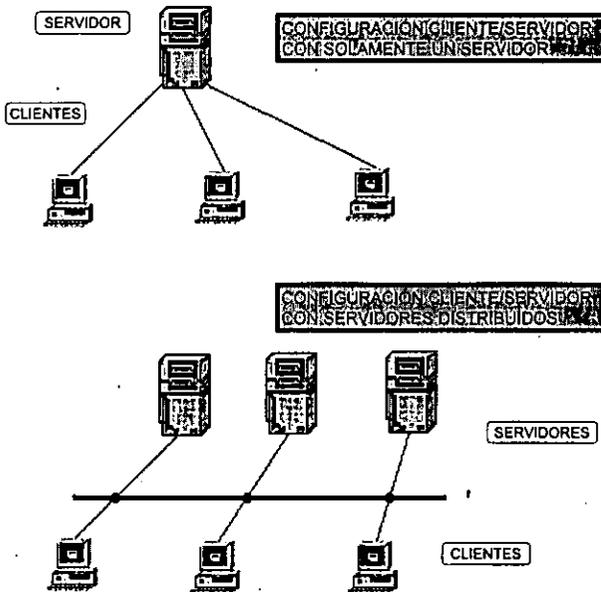


Fig. 2.2. Los datos pueden estar concentrados o distribuidos o diferidos

Existen dos tipos de arquitecturas del modelo Cliente/Servidor

1. Arquitecturas Cliente/Servidor basadas en hardware.
2. Arquitecturas Cliente/Servidor basadas en software.

2.6.2 Arquitecturas del Modelo Cliente/Servidor basadas en Software.

Es un conjunto de programas que constituyen una aplicación independiente del hardware que describe el lugar de los componentes de esa aplicación y como se comunican entre sí, la aplicación se divide en tres partes que son la **Interfaz gráfica de usuario (GUI)** que es la forma de comunicación de la aplicación con el usuario, la **lógica de la aplicación**, que es la parte que realiza los cálculos y operaciones y una **base de datos**, que es donde se almacenan los datos.

2.6.3 Arquitecturas Cliente/Servidor basadas en hardware

Tipos de arquitecturas

- Servidor de archivos
- Peer to Peer (Punto a Punto)

2.6.3.1 Servidor de archivos

Es en donde tenemos solo una máquina muy poderosa con gran capacidad de procesamiento y de cómputo donde se tienen concentrados todas las aplicaciones, los GUI, y la base de datos en un solo equipo y todos los clientes van a ser terminales no gráficas o modo carácter cuya única tarea es desplegar información al usuario.

Ventajas

- Se posee la información centralizada en un solo equipo.
- Fácil de administrar.

Desventajas

- Se debe adquirir un equipo altamente costoso (capaz de contar con CPU simétricos, gran memoria, gran capacidad en disco)
- Escalabilidad limitada.
- Bajo rendimiento conforme se incrementa el número de usuarios.
- Alto tiempo de respuesta con plena carga de usuarios.
- Rápida disminución de espacio en disco.

2.6.3.2 Peer to Peer (Punto a Punto)

Es un modelo combinado ya que depende de un SO ligero para la comunicación peer to peer. Este tipo de arquitecturas comercialmente fueron vendidas como basadas en hardware ya que lo único que necesitaban adicionalmente para trabajar Cliente/Servidor punto a punto eran tarjetas de red.

Es un modelo en donde cualquier computadora en la red puede funcionar como cliente o como servidor o como servidor y cliente a la vez.

Ventajas

- Mayor rendimiento en los equipos.
- Menor costo en los equipos. (No se requiere de equipos tan poderosos)
- Administración descentralizada (donde cada usuario dueño de su equipo de cómputo se encarga de administrar los recursos de su equipo)
- Las computadoras en este tipo de entornos se constituyen de manera natural en una red fácil de mantener y administrar.

Desventajas

- No solicita permiso para conectarse como cliente a un servidor.
- Poca seguridad.

2.7 Tipos principales de Arquitectura Cliente/Servidor en el mercado.

Son de tres tipos las configuraciones Cliente/Servidor que generalmente se encuentran más en el mercado y a las que más recurren las organizaciones para interconectar sus equipos:

- Acceso Remoto a Datos (RDA)
- Servidor de Base de Datos o Base de Datos Distribuida (DBS)
- Servidor de Aplicaciones o Aplicaciones Distribuidas (APS)

2.7.1 Acceso Remoto a Datos (RDA)

En este tipo se puede definir las siguientes partes constitutivas, la parte GUI y la parte de logística de la aplicación, estas dos partes residen en la parte cliente y la base de datos reside en la parte del servidor. En la parte cliente están todas las pantallas o el GUI de la aplicación que es en donde el usuario elige los procesos, por ejemplo, si se desea hacer una consulta de alguna información, el GUI le pasa esa solicitud a la aplicación cliente, esta realiza las operaciones previas necesarias para posteriormente hacer la solicitud a la base de datos del servidor a través de un SQL o cualquier otro motor de base de datos este se encarga de buscar los registros que coinciden con esta búsqueda, en el momento que encuentra esos datos se los pasa a la aplicación del cliente y esta se encarga de realizar los procesos, cálculos etc. y si necesita datos adicionales los solicitará al motor de base de datos y así sucesivamente hasta que el proceso se termina, cuando la lógica de la aplicación ha terminado su labor pasa el resultado al GUI y lo despliega al usuario.

Ventajas.

Las ventajas que podemos apreciar son las siguientes:

Se trata del modelo tradicional en el mercado y hay un gran número de herramientas gráficas que soportan este tipo de modelo, también podemos acceder a diferentes bases de datos vía remota, además de su facilidad de uso.

Desventajas.

Se genera tráfico en la red ya que la aplicación y la base de datos realizan muchas solicitudes.

La escalabilidad es limitada, por que si queremos migrar el sistema operativo o instalar una nueva aplicación; tendremos que recompilar toda la aplicación. Además. El tiempo de respuesta es lento debido al tráfico de la red.

2.7.2 Servidor de Base de Datos o Base de Datos Distribuida (DBS)⁶

Tenemos el GUI en la parte cliente y en la parte servidor esta la aplicación además de la base de datos. Aquí si el cliente quiere hacer una consulta selecciona la pantalla de consulta hace la petición al servidor y la aplicación cuenta con "store procedures" (procedimientos almacenados) los cuales los genera un procedimiento o rutina de la aplicación. Estos procedimientos almacenados hacen una petición a la base de datos, esta procesa y genera la información de respuesta para el cliente.

Ventajas

- Reduce el tráfico en la red debido a que todos los procedimientos y los cálculos se realizan en el servidor.
- Mayor rendimiento ya que disminuye el tráfico en la red.
- Mejor tiempo de respuesta.

Desventajas

- Estamos encadenados a un solo tipo de base de datos.
- No es posible trabajar con diferentes tipos de bases de datos.
- Se tiene gran carga en la base de datos. Por que en el momento de estar haciendo el "store procedures" y los RPC's se activan más procesos que están entrando a la base de datos.

2.7.3 Servidor de Aplicaciones o Aplicaciones Distribuidas (APS)

En donde el GUI esta en la parte cliente, la aplicación como un servidor y la base de datos también como un servidor este modelo actualmente es el que se utiliza más donde tenemos varias terminales y varias PC's accediendo a diferentes aplicaciones, por ejemplo: nomina u otras bases de datos, aquí el GUI funciona como un "front-end" plenamente y la base de datos como un "Back-end", también plenamente y la parte de la lógica de la aplicación es la responsable de la comunicación entre el "Front-end" y el "Back-end", la aplicación también tiene que manejar el nivel de conductividad entre ellas.

Ventajas

- Descarga el trabajo de la base de datos
- Diminuye el tráfico en la red.
- Tenemos portabilidad en la base de datos.
- Fortalece la computación distribuida.

⁶ Es un típico entorno distribuido Cliente/Servidor (apartado 1.2.5.2 de este trabajo)

2.8 Ventajas del modelo Cliente/Servidor en General

El constante avance tecnológico lleva a que cada día dispongamos de equipos con mayor capacidad de procesamiento y de almacenamiento. Hoy en día se cuenta con un equipo que en muchos casos rinde entre un 40% a 100% respecto al equipo que está funcionando como servidor de la red de la organización⁷. Es el modelo Cliente/Servidor quien aprovecha ampliamente esta ventaja individual de cada equipo. Con el modelo Cliente/Servidor se integran los sistemas de información corporativa y se fusionan los sistemas existentes y nuevos. También la inversión hecha en equipo de cómputo se ve mejor aprovechada. Se pueden integrar diferentes tipos de equipos y sistemas operativos en un ambiente único de procesamiento, de esta forma se empiezan a borrar las diferencias entre hardware y software debido a que se propicia el trabajo en forma transparente con sistemas abiertos, y al trabajar con sistemas abiertos es posible instalar productos de diferentes proveedores sin interferir en el desempeño de las demás aplicaciones ni de los otros equipos de la red.

Ventajas de trabajar en entornos Cliente/Servidor

- ◆ Las aplicaciones bien diseñadas dentro de un entorno Cliente/Servidor son relativamente económicas y ofrecen la capacidad de los grandes sistemas al tiempo que son fáciles de personalizar para desempeñar acciones específicas.
- ◆ En los entornos Cliente/Servidor se disminuye el tráfico en la red debido a que el servidor solo se concreta a enviar los resultados de las consultas por la red.
- ◆ Se asigna la tarea de la búsqueda de archivos a un equipo que tiene más capacidad que el cliente y puede tratar las peticiones, en una red de varios servidores donde el proceso se distribuirá más homogéneamente que en un sistema tradicional basado en un único servidor.
- ◆ Una red en una arquitectura Cliente/Servidor distribuida es ahorradora de memoria RAM en el equipo cliente, porque toda la lógica de E/S de datos y de archivos se encuentran en la aplicación del servidor.
- ◆ Los servidores en los entornos Cliente/Servidor centralizado son capaces de almacenar grandes cantidades de datos, lo cual permite tener más espacio en los equipos cliente para otras aplicaciones. Debido a que los servicios de archivos de datos están centralizados en el servidor son más fáciles de respaldar, proteger y mantener.
- ◆ Cliente/Servidor permite el acceso de un cliente a diferentes servidores en forma simultánea.
- ◆ Implementar un entorno Cliente/Servidor distribuido reducirá el tráfico sobre la red, debido a que solo por la red viajarán los requerimientos y las respuestas.
- ◆ Cuando se utiliza Cliente/Servidor es posible operar bajo sistemas abiertos, lo que significa capacidad para cambiar las plataformas con un mínimo de problemas y de riesgos.

⁷ Según estudios de grupo Scanda, empresa en México dedicada a consultoría y a la implementación de soluciones Cliente/Servidor.

- ◆ Sin duda uno de los mayores beneficios de la arquitectura Cliente/Servidor en general es la habilidad de poder controlar y distribuir datos en una organización. Esta capacidad permite diseñar aplicaciones de acuerdo a la forma en que la organización trabaja.

2.9 Desventajas del modelo Cliente/Servidor en General.

En las desventajas encontramos que las aplicaciones Cliente/Servidor desde el punto de vista de diseño y desarrollo llegan a ser de las aplicaciones más complejas respecto de las aplicaciones tradicionales (monousuario), también las aplicaciones Cliente/Servidor exigen más de la infraestructura de red. Se consideran que la seguridad inicial o natural de las aplicaciones es más débil en este tipo de entornos por lo cual se deben de invertir mucho más recursos en poder reducir o eliminar la vulnerabilidad a la seguridad. Debido a que la seguridad esta sujeta al diseño de la aplicación y a las medidas que se le incorporen al entorno para hacerlo seguro.

Creación de la plataforma para Aplicaciones Cliente/Servidor

3.1 Introducción

Hablar de la creación de la plataforma Cliente/Servidor es referirse al lugar físico, a la lógica y a las reglas que rigen dentro de un entorno de cómputo para que este pueda ser apto para soportar aplicaciones Cliente/Servidor. El lugar físico se refiere al hardware básico necesario donde se implementará la lógica de las aplicaciones. La plataforma para las aplicaciones también es conocida como la infraestructura Cliente/Servidor.

3.2 Diferencia entre aplicaciones de 16 bits y de 32 bits.

El que se desarrolle una aplicación para un Sistema Operativo Microsoft de 32 bits lleva como ventaja que esta aplicación pueda ser diseñada tanto a 16 o a 32 bits. Debido a que los microprocesadores de 32 bits tienen la capacidad de trabajar en la modalidad virtual o comúnmente llamado modo protegido en el que se pueden acceder a todas las características y funciones que ofrece un microprocesador de 32 bits.

Pero el problema se encuentra en poder alternar entre el modo de funcionamiento real y el modo de funcionamiento protegido ya que son incompatibles entre sí y sólo uno puede estar activado a la vez.

Las aplicaciones de 16 bits

Una aplicación de 16 bits utiliza para en su ejecución una palabra de tamaño de 16 bits este tamaño también es utilizado para su desplazamiento interno y externo es decir memoria, subsistema de disco, procesador y subsistema de red. Este tipo de aplicaciones son más lentas e inseguras y menos robustas, obviamente demandan también de menos memoria, esta característica esta también relacionada con el tipo de microprocesador, un microprocesador cuya arquitectura interna es de 16 bits jamás podrá ejecutar aplicaciones de 32 bits. Un microprocesador es considerado de 32 bits a partir de la familia Intel 80386. Por otra parte la cualidad de multitarea que aprovecha Windows NT también esta relacionada con el microprocesador.

Las aplicaciones de 32 bits

Una aplicación de 32 bits ocupa una tamaño de palabra de 32 espacios, el doble de una de 16 bits, el crecimiento hacia el doble de espacio se explica por la relación con los octetos (ocho bits, que es la unidad básica de representación de un carácter). Una aplicación diseñada para trabajar a 32 bits será más rápida, más segura, más robusta y demandará de mayor espacio en los recursos, (memoria, subsistema de disco, procesador y subsistema de red). La aplicación siempre dependerá para su pleno funcionamiento de los límites y ventajas que ofrezca el sistema operativo sobre el cual se este desempeñando. Por esta razón aunque desde el microprocesador Intel 80386 empieza la familia de microprocesadores de 32 bits, la versión actual de Windows NT (4.0) no funciona sobre este microprocesador debido a requerimientos adicionales, por lo que es necesario para instalar Windows NT versión 4.0, un microprocesador más evolucionado.

3.2.1 Ventajas de diseñar aplicaciones de 32 bits.

Las aplicaciones de 32 bits en ambientes Windows NT cuentan con la capacidad de generar automáticamente un espacio de memoria reservado para su desempeño, en este espacio de memoria se ejecutan todos los procesos en forma de threaders que la aplicación desencadene. Esta ventaja es propia de los sistemas operativos Windows NT la cual es aprovechada por las aplicaciones de 32 bits exclusivamente. Esto hace que cuando se desempeñen aplicaciones de 32 bits sobre sistemas operativos Windows NT, no interfieran con otras aplicaciones en

funcionamiento en tal forma que si llegara a ocurrir un problema con una aplicación y esta se descontrolara y dejara de funcionar no afecta a las demás aplicaciones que en ese momento se encuentren corriendo sobre la máquina ya que cada una trabaja en espacios separados de memoria RAM. Recordemos que esta separación no la realiza la aplicación por el solo hecho de ser diseñada a 32 bits sino que la realiza el SO. Por tanto al diseñar y utilizar aplicaciones de 32 bits contribuimos a aumentar la seguridad en el desempeño de todas las aplicaciones Cliente/Servidor de nuestra red.

3.3 Requisitos presentes y futuros que debe de cumplir una red LAN para poder implantar Cliente/Servidor.

Como ya hemos visto el entorno Cliente/Servidor constituye el cómputo basado en redes. Esto significa que los diseñadores de la red deben de colaborar estrechamente con quienes crean las aplicaciones Cliente/Servidor porque de otra forma ambos fracasarán. Los diseños de aplicaciones Cliente/Servidor optimizadas utilizarán las redes diez² veces más eficientemente que las configuraciones prefabricadas. Las entidades de un diagrama de flujo de datos (bases de datos, usuarios y otros recursos) están distribuidas en una red; lo que interconecta a las entidades son los flujos de tráfico. Los diseñadores de la red deben de tener en mente estas corrientes de transacciones al diseñar la infraestructura de la red.

Cada vez es más difícil planear una red; los diseñadores de redes se están percatando de que las aplicaciones del mundo real y los sistemas de redes se combinan para dificultar su planeación, en la ecuación de diseño de la red debe incluirse factores como los recursos centralizados, los nuevos tipos de aplicaciones y los nuevos paradigmas de uso, el flujo cambiante de tráfico, un mayor número de usuarios conectados, y otras plataformas de cómputo más poderosas. Además de que la gran cantidad de opciones tecnológicas y de arquitectura podrían influir en el problema del diseño de la red hasta hacerlo parecer casi irresoluble.

Cuando se planea la construcción de una red, los desarrolladores de aplicaciones y los diseñadores de la red deben de trabajar en forma conjunta los aspectos siguientes:

- **Romper con las estructuras organizacionales.**
- **Establecer un presupuesto para la red.**
- **Incorporar nuevas aplicaciones si es necesario.**
- **Estudiar las alternativas de topologías de aplicaciones.**
- **Realizar pruebas razonables sobre la red.**
- **Evaluar conjuntos de protocolos.**
- **Identificar a los usuarios y determinar si su posición es estática o dinámica dentro la red.**
- **Identificar los ciclos de uso de las aplicaciones.**
- **Rebasar los requerimientos de la producción actual de la red.**
- **Analizar las compensaciones.**

² Según estimación de grupo Northeast Consulting Resource.

Romper con las estructuras organizacionales

La mayor parte de las organizaciones de IT (Tecnología de Informática) constan de un grupo de operación de la red y otro de desarrollo de aplicaciones, generalmente con un jefe común, lo cual tiene sus inconvenientes debido a que no siempre esto consigue buenos resultados. Tanto los planificadores de la red como los desarrolladores de aplicaciones deben de crear grupos multidisciplinarios de trabajo que colaboren con los usuarios, para conocer y detectar las necesidades para así poder sentar bases de planificación y desarrollo reales a la vez que razonables.

Establecer un presupuesto para la red

Se debe de contemplar el costo que representará incrementar o implantar nuevos nodos en la red y que técnicamente se pueda soportar el aumento del tráfico en la red, cuando aumentan los nodos en la red esto redundará en tiempos de respuesta reducidos ó amplios dependiendo también de la tecnología de conexión de red que estemos usando. Lo ideal es que sean reducidos los tiempos de respuesta de las aplicaciones Cliente/Servidor distribuidas.

Incorporar nuevas aplicaciones si es necesario

Esto tiene lugar en las primeras fases del diseño de la red donde es posible establecer plazos adecuados para el rediseño de la red si es necesario. Los diseñadores de la red deben de poder identificar las necesidades relacionadas con nuevas aplicaciones en las etapas de formación, la planeación adecuada y la comunicación plena dentro de la organización acaba con las sorpresas que pudieran tener los diseñadores de la red al planear para un número menor de aplicaciones de las que los gerentes o directores de las organizaciones tienen en mente.

Estudiar las alternativas de topologías de aplicaciones

La evaluación de las topologías de aplicaciones distribuidas y centralizadas se debe hacer antes de que estas se codifiquen, además. Estudiar el impacto que tiene al acercar los servidores al cumulo de usuarios que más demanden del uso de la aplicación y efectuar las actualizaciones de la base de datos maestra, también estudiar el efecto de estas topologías en la red, así como el tiempo de respuesta para el usuario.

Realizar pruebas razonables sobre la red

Desarrollar un prototipo de red con flujos aproximados de tráfico de la aplicación es un punto de referencia importante para que los desarrolladores de aplicaciones construyan las GUI a fin de mantener en aviso del tiempo que tardara su petición y cuando el GUI cambiará.

Los prototipos de red evidencian cuellos de botella potenciales que pueda presentar la red y demuestran el desempeño de redes alternativas. Permitir que los usuarios ayuden a determinar el tiempo de respuesta y colaboren con los administradores para cuidar el desempeño de la red ó justificar los gastos en infraestructura.

Evaluar a los conjuntos de protocolos

Es importante conocer protocolos de comunicación alternos para poder determinar los requerimientos de afinación. Realice un esfuerzo por comprender cómo la optimización de las operaciones de la red para una aplicación (o un conjunto de aplicaciones) podría perjudicar el desempeño de otras.

Identificar a los usuarios y determinar si su posición es estática o dinámica en la red

Es importante poder determinar la ubicación de los servidores respecto de los usuarios ya que de esto dependerá también la ubicación de la información a través de determinar si los usuarios cambiarán de lugar periódicamente o estarán contenidos en una sola red o subred y como se conectarán a esta.

Identificar los ciclos de uso de las aplicaciones

Planificar en función de las horas pico es un punto muy importante, por que de esta forma se pueden analizar las tendencias de uso semanal, mensual, trimestral, anual, etc. Trate de prever como un incremento en las actividades de la organización afectará a la red y determine que ancho de banda extra necesitará para que la organización siga funcionando.

Rebasar los requerimientos de la producción actual de la red

Es importante poder realizar pruebas para determinar si se podrá soportar la carga de información en la red en el futuro si se adquieren nuevas aplicaciones. Analice los límites y los cuellos de botella. Trate de aprender como influirán en ella los cambios en la población de usuarios.

Analizar las compensaciones

Con frecuencia la centralización de los datos significa agregar capacidad al centro de la red, mientras que distribuir los datos podría requerir de un incremento de las redes locales de los grupos de trabajo y en el poder de estos mediante la instalación de servidores de acoplamiento más rápidos se mejoran los tiempos de respuesta.

En la figura 3.1 se muestra el tiempo de respuesta de la red necesario para llevar a cabo una transacción, consulta/respuesta extensa en una aplicación de prueba⁹. Si bien el tiempo de respuesta es importante, dado que esta aplicación utiliza procesos de manufactura, la red sólo necesita soportar a 20 usuarios concurrentes. Los tiempos de respuesta (en el eje Y) indican únicamente el tiempo de red. Para determinar el verdadero tiempo de respuesta para esta consulta/respuesta, debe agregarse el tiempo de procesamiento del servidor. Por ejemplo 30 segundos de procesamiento general en el servidor de esta manera la transacción llevará aproximadamente 50 segundos en realizarse sobre una red de tecnología FDDI (19 de tiempo de red +30 de tiempo de procesamiento del servidor) respecto de 40 segundos en una red de tecnología ATM.

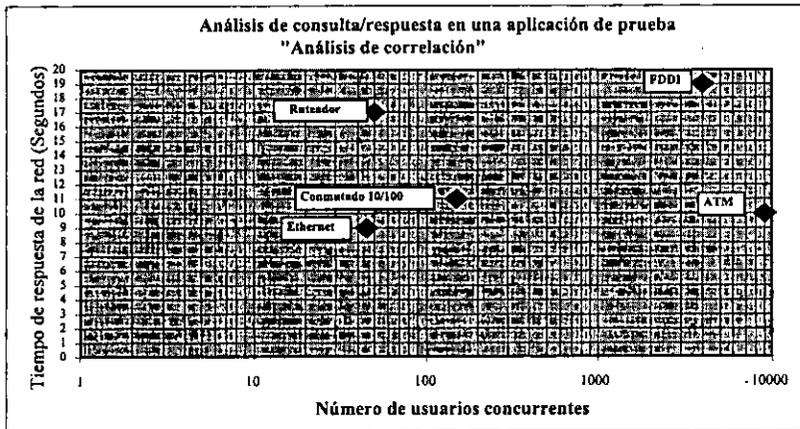


Fig. 3.1 Muestra la capacidad de respuesta de cinco redes de tecnología diferente corriendo una aplicación Cliente/Servidor de prueba.

3.3.1 Ethernet proporciona resultados suficientes

Como se aprecia en la gráfica anterior la red local Ethernet proporciona el mejor tiempo de respuesta de toda las alternativas de redes con pocos usuarios, dado que nada se interpone en las comunicaciones Cliente/Servidor, excepto la conexión CSMA/DA que es el método de acceso al cable de toda red Ethernet por lo que es normal y natural encontrar colisiones. Ethernet por sí sola soporta más de 40 usuarios (siendo la aplicación de prueba la única aplicación que se ejecutan la red). Además de que Ethernet es una red barata, respecto de las demás tecnologías.

Las demoras en la red se miden en centésimos o milésimos de segundo, las diferencias entre diferentes tecnologías de red son considerablemente importantes para el diseño. Una diferencia de 1/100 de segundo en la demora puede provocar problemas a los usuarios si ocurre mil veces en el curso de una tarea de una aplicación. En las redes de aplicaciones Cliente/Servidor modernas se presentan un número muy grande de demoras aparentemente inconmensurable, pero sus efectos se combinan. En última instancia, las pequeñas diferencias

⁹ Para más información recurrir al boletín Resumen Ejecutivo con el tema "¿Que ancho de banda se necesita? diseño de redes en un mundo complejo, Distribuido por Northeast Consulting Resource Inc. Por Christopher Serjak.

de rendimiento de las tecnologías de red (puentes ruteadores, conmutadores, tipos de red local) tiene como resultado significativas diferencias de tiempo de respuesta de las aplicaciones y en el número de usuarios que una arquitectura de red específica puede soportar confiablemente.

3.3.2 Cálculo del número de usuarios soportados

En el ámbito del diseño de redes se deben de aplicar cuatro pasos para poder calcular el número de usuarios que soportará una red.

1. Calcular el número de usuarios que podrán utilizar una aplicación concurrentemente.
2. Determinar qué tanto tráfico genera un sólo grupo de trabajo.
3. Determinar cuántos grupos de trabajo pueden utilizar los recursos de la red al mismo tiempo.
4. Multiplicar el número de grupos de trabajo sobre la red, por el número de usuarios en cada grupo de trabajo para determinar la población total de usuarios.

Otro aspecto importante a considerar que proporciona un mejor tiempo de respuesta, son las arquitecturas conmutadas (tanto el paquete conmutado 10/100 como ATM) son mejores que las arquitecturas ruteadas¹⁰ o FDDI de acceso compartido tradicionales. De hecho, la conmutación proporciona un tiempo de respuesta 50% mejor, debido a los mecanismos de conmutación especializados que se están fabricando. Casi todos los fabricantes están de acuerdo en que los conmutadores de paquete proporcionarán mejores características de latencia/demora que la gran mayoría de los puentes y ruteadores disponibles en el mercado.

Nota:

En la práctica no hay reglas fijas y fáciles de aplicar para poder elegir una arquitectura de red en lugar de otra. Los requisitos de las aplicaciones son tan numerosos y variados tanto como el número de aplicaciones con que cuenta una empresa moderna. La mayor velocidad en las redes se traduce en mejores tiempos de respuesta de las aplicaciones Cliente/Servidor para los usuarios.

Por otra parte la ecuación actual de diseño de redes contiene demasiadas variables como para que los planificadores confíen en su intuición. Hay pocas respuestas para problemas complejos de diseño de redes/aplicaciones Cliente/Servidor y hay demasiados factores en juego, pero la cooperación y la comprensión, aunadas a una planeación cuidadosa, tanto de los desarrolladores de aplicaciones como de los diseñadores de redes, son fundamentales para el buen desempeño de las aplicaciones Cliente/Servidor.

¹⁰ Ruteadas es referirse a aquellos entornos donde los protocolos de comunicación pueden ser aceptados por un ruteador.

3.4 Un poco de historia de Windows NT

Advertencia

Cabe mencionar que no se pretende profundizar en la explicación del funcionamiento total y puntual del sistema operativo Windows NT, se abordarán temas generales del SO y solo se tratará de profundizar en aquellos temas que estén relacionados con el desarrollo, instalación y uso de aplicaciones¹¹.

En 1993 Microsoft introdujo en el mercado Windows NT (Nueva Tecnología). Microsoft promovió este sistema operativo como un sistema totalmente innovador el cual incluía todas las características que esperaban los usuarios en un SO, para estaciones de trabajo, servidores pequeños y medianos. Windows NT recién liberado en 1993 con una API nueva (win32)¹² con nuevas herramientas de administración de usuarios y de sistemas, Windows NT encuentra sus raíces de su arquitectura medular y de su implementación a mediados de la década de los setenta.

La mayoría de los principales desarrolladores de Windows NT, incluyendo al arquitecto de VMS¹³ vienen de Digital que influyeron de manera definitiva en el desarrollo de Windows NT.

Se considera a David N. Cluter como el arquitecto principal de Windows NT. Trabajo al salir del colegio para Dupont ahí realizó simulaciones en máquinas de Digital como parte de su trabajo, empezó a concentrarse en el desarrollo de sistemas operativos más que de aplicaciones. En 1971 se integro a Digital para desarrollar sistemas operativos para la familia PDP-11 y RSX-11M fue el primer sistema operativo donde incorporo importantes conceptos y principios de diseño que más tarde surgieron en Windows NT. RSX-11M es un sistema operativo de las máquinas PDP-11 que Digital diseño para el control industrial y de manufactura.

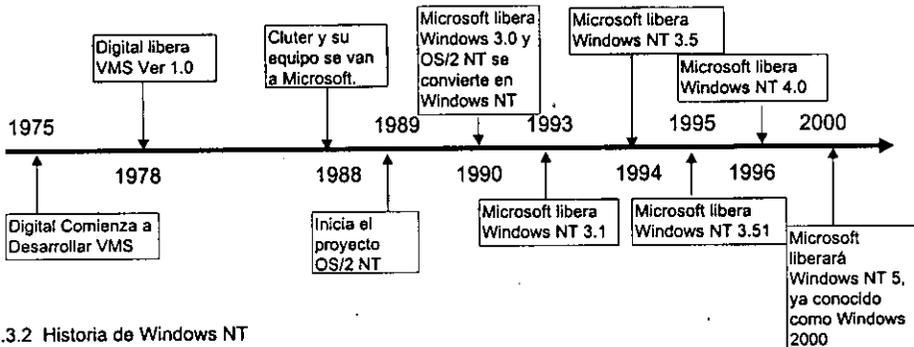


Fig.3.2 Historia de Windows NT

¹¹ Si el lector desea conocer más acerca de aspectos relacionados con Windows NT puede recurrir a los manuales de Microsoft del compendio System Engineer de Microsoft Press.

¹² La API win32 es una evolución de la API de 16 bits de Windows 3.0 además de ser compatible con esta. La compatibilidad de la API de 16 bits con la API de 32. Fue un objetivo claro de Microsoft para poder correr aplicaciones no modificadas de Windows 3.0 en Windows NT además de soportar porciones de las APIs de DOS, OS/2 y POSIX. - El tema de las APIs se tocará mas adelante.

¹³ VMS Virtual Memory System sistema operativo UNIX desarrollado por Digital destinado a las computadoras VAX, este sistema operativo aun se encuentra en uso en varias organizaciones.

En 1975 Digital descubrió que sus competidores estaban desarrollando procesadores de 32 bits y por lo tanto esa tecnología alejaría a los compradores de la arquitectura de 16 bits de PDP, entonces Digital se dirigió al desarrollo de arquitecturas de 32 bits llamadas VAX, Cluter fue requerido para desarrollar el sistema operativo para las VAXs, llamado VMS. En 1988 Cutler dejó Digital y ese mismo año se incorporó a Microsoft junto con 20 empleados del proyecto Prims de Digital, para continuar en Microsoft con este proyecto llamado ahora OS/2 NT, Microsoft deseaba que este SO compitiera con el OS/2 de IBM y conservar a la OS/2 API como su interfaz principal, pero el éxito de Windows 3.0 en 1990 hizo cambiar los planes y la relación con IBM, y cambio el nombre de OS/2 NT a Windows NT y diseño la Win32 API como interfaz.

3.4.1 Constitución de Windows NT

Un reconocimiento de Windows NT principia con un entendimiento de la arquitectura del sistema, mencionaremos a continuación los elementos clave tanto de Windows NT Server como de Windows NT workstation.

Windows NT utiliza dos modos para mantener su eficiencia e integridad, *el modo usuario y el modo kernel*

Modo Usuario.

Aquí se ejecutan las aplicaciones y los subsistemas que soportan estas. La modalidad de usuario tiene las siguientes limitantes:

- No tiene acceso directo al hardware.
- Esta limitado a una asignación direccionada de espacio.
- Puede requerir utilizar espacio de disco duro como memoria virtual.

Los accesos al hardware son otorgados por componentes del modo kernel. De esta manera se tiene protección contra mal funcionamiento de aplicaciones o contra accesos no autorizados al hardware. El modo de usuario ejecuta procesos en baja prioridad de esta forma tienen menores accesos a los ciclos del CPU que los procesos que corren en modo kernel.

Modo Kernel

En el modo kernel corre el Windows NT ejecutivo también conocido como **los componentes ejecutivos de Windows NT**, para proteger los servicios ejecutivos de las aplicaciones del modo usuario. El modo de kernel proporciona acceso a toda la memoria de la computadora y al hardware. Sin embargo. La memoria donde se aloja el sistema operativo esta protegida, las aplicaciones no pueden acceder directamente a la memoria donde se aloja el sistema operativo.

Los componentes ejecutivos de Windows NT

Estos componentes son los que soportan los requerimientos de las aplicaciones del modo de usuario. El kernel consiste de tres subsistemas ejecutivos principales, que son responsables de manejar los recursos.

- Servicios Ejecutivos.
- El microkernel.
- La capa de abstracción de hardware. (HAL)

Servicios Ejecutivos.- Consiste de manejadores y de dispositivos controladores. Los manejadores son varios módulos que manipulan los dispositivos de I/O, objetos de seguridad, procesos, comunicación de interprocesos, memoria virtual, y manejadores de ventanas y gráficos. Los dispositivos controladores son módulos que controlan el acceso al hardware.

Microkernel.- Es un componente que proporciona los servicios básicos del sistema operativo, tales como manipuleo de interrupciones u organización de tareas.

La capa de abstracción de hardware.- Es la parte que aísla las mayores diferencias de interface del hardware que puede soportar Windows NT haciendo a este más portable.

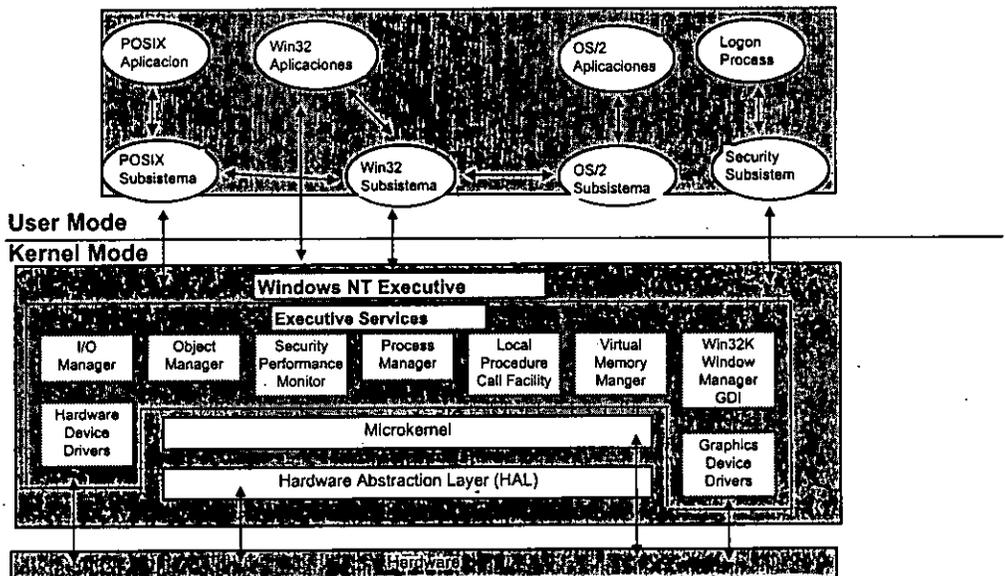


Fig. 3.3 Estructura interna de Windows NT.

3.4.2 ¿Por qué se elige Windows para este trabajo?

Desarrollar aplicaciones en Windows para usuarios finales representa dedicar una buena parte del tiempo de desarrollo a diseñar las GUIs totalmente gráficas a las que la mayor parte de los usuarios de todas las organizaciones están ya acostumbrados, esto debido fundamentalmente a la gran penetración que ha tenido Windows en el mercado. Si bien esto ha producido que los desarrolladores de aplicaciones sigan normas de desarrollo tendientes a ambientes de Windows por desempeño y comodidad con presencia cada vez mayor de herramientas de desarrollo para aplicaciones MS Windows. Se han producido también tendencias que responden meramente a aspectos puramente mercantiles. No obstante no es imprescindible desarrollar bajo ambiente Windows, ya que existen otras alternativas.

Para poder principiar a desarrollar aplicaciones Cliente/Servidor, es necesario conocer las características principales del sistema operativo bajo el cual se desempeñarán, sus alcances y sus limitantes, para sobre la base de eso poder pronosticar con exactitud de igual forma los alcances y limitaciones de las aplicaciones Cliente/Servidor que se desarrollen.

La tarea más significativa de un SO (Sistema Operativo), es administrar memoria eficientemente y controlar la manera en que las aplicaciones usan recursos del sistema. Un sistema operativo que hace administración de memoria y programación de tareas, permite tanto intercambio de contexto como tareas múltiples. El intercambio de contexto suspende la actividad de una aplicación cuando esta es empujada hacia un segundo plano, y otra es traída hacia el frente, solo la aplicación en la ventana del frente permanece activa. El intercambio de contexto es suficiente para muchos ambientes de trabajo y proporciona grandes ganancias en productividad por que tenemos salidas y aperturas de diferentes aplicaciones en ambientes de tareas individuales.

Otra característica es la de tarea múltiple, es la cualidad de poder correr operaciones múltiples concurrentemente, esto es aplicable en ambientes de usuarios múltiples, con tareas pesadas como consultas a bases de datos, compilación de programas y complejo procesamiento de gráficos. La característica de tarea múltiple es de tipo preferente, esto es, que en cualquier momento el usuario o el mismo sistema operativo puede interrumpir una tarea determinada, si esta causando problemas por haberse quedado colgada o trabada. Así interrumpiendo esas tareas el tiempo de microprocesador se reparte solo entre aquellas que están funcionando correctamente.

Windows NT es un sistema operativo netamente de trabajo a 32 bits, es un sistema operativo enfocado a ambientes Cliente/Servidor. Es el sistema operativo más popular de Microsoft, y por lo tanto de gran penetración como ya mencione. Se espera que este sistema operativo será utilizado por un 90% de las computadoras de escritorio en la próxima década.

Ambos tanto Windows NT Server como Windows NT Workstation son sistemas operativos baratos comparados con otros sistemas operativos formales de 32 bits, (como por ejemplo UNIX u OS/2 en sus diferentes modalidades) y su trayectoria de confiabilidad ha sido comprobada, así como su rendimiento y bajos costos de administración.

Windows NT permite ejecutar aplicaciones de 32 bits más eficientemente que cualquier otro sistema operativo de Microsoft, y las ejecuciones de las aplicaciones resultan ser más rápidas y seguras. Por esta razón es preferible usar aplicaciones de 32 bits en cuando se tenga como plataforma operativa a Windows NT. Aunque las aplicaciones de 16 bits pueden también ser ejecutadas estas deberán ser adaptadas manualmente para ser ejecutadas en su propio espacio de memoria para que no interfieran con otras aplicaciones de 16 bits en el momento de

la ejecución. Esta característica es conocida como "sistema de protección de memoria" y actúa de manera automática cuando se ejecutan aplicaciones de 32 bits.

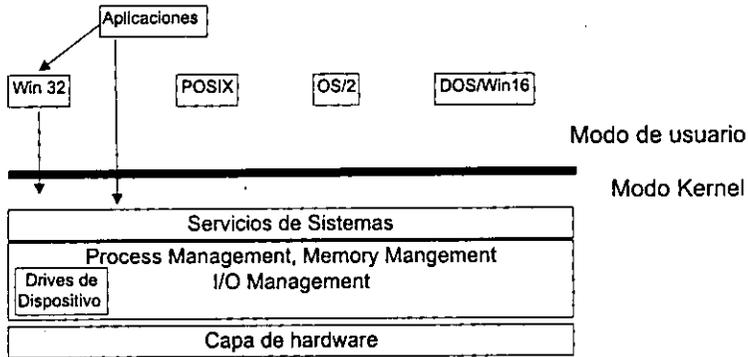


Fig. 3.4 Arquitectura Básica de Windows NT.

3.4.3 Características principales de Windows NT

Windows NT Workstation Versión 4.0

Windows NT, a diferencia de otros sistemas operativos cuenta con capacidades de red ya incluidas dentro de sí mismo. Todavía hasta hace muy poco tiempo, el software de la red de los equipos PC se agregaba a los sistemas operativos existentes, así cada equipo contaba con dos sistemas operativos el autónomo y el sistema operativo de red ambos sistemas operativos tenían que estar instalados en el mismo equipo

Windows NT Workstation también puede ser clasificado como un sistema operativo de escritorio aunque tiene características limitadas respecto de Windows NT Server. Windows NT Workstation ver. 4.0 puede ser utilizado en organizaciones pequeñas (de no más de 10^{14} personas) como un servidor.

- Windows NT Workstation 4.0 continúa proporcionando el mejor rendimiento total de aplicaciones que cualquier otro miembro de la familia de escritorio Windows.
- Microsoft Windows 98 y 95 es superado en hardware principal soportado por Windows NT Workstation 4.0.
- Proporciona la más alta confiabilidad que cualquier otro miembro de la familia de escritorio Windows.

¹⁴ Según recomendaciones de Microsoft.

- Ha sido probada en aproximadamente 9,000 marcas de PC y 6,500 periféricos y dispositivos de PC.
- Ejecuta las aplicaciones empresariales líderes disponibles en el mercado.
- Ejecuta varias aplicaciones de 16 bits en forma más rápida y confiable que cualquier otra versión de Windows.
- Ofrece características clave de movilización para laptops incluyendo la administración de energía y soporte para tarjeta de PC, así como soluciones hot-docking a través de accesorios de terceras partes.
- Ofrece los costos más bajos totales de propiedad (TCO) que cualquier sistema operativo de escritorio de la familia Windows.
- Es un cliente de red excelente de fines múltiples para el Windows NT Server.

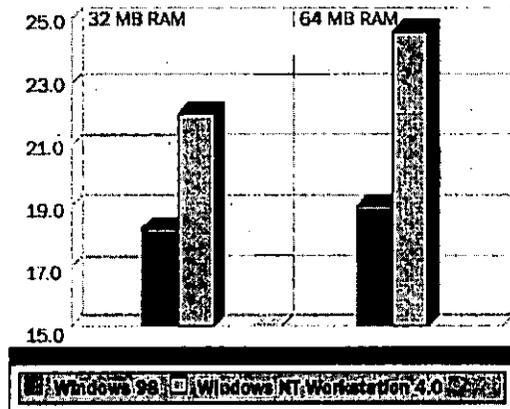


Fig. 3.5. Comparativo Winstone 98 Desempeño General Conforme aumenta el RAM el nivel de desempeño del sistema operativo de Windows NT Workstation 4.0 mejora en comparación con el sistema operativo de Windows 98. De hecho, un estudio independiente realizado recientemente por NSTL, donde se utilizó el ampliamente aceptado punto de referencia Winstone 98, versión 1.0, de Ziff-Davis¹⁵. Windows NT Workstation 4.0 se desempeñó 26% más rápido en promedio que el Windows 98 en la prueba general empresarial Winstone, en cualquier configuración de memoria.

Windows NT Server Versión 4.0

Microsoft Windows NT Server 4.0 es un sistema operativo para redes muy poderoso. Su facilidad de uso, flexibilidad, servicios avanzados de Internet/intranet y comunicaciones, satisfacen hasta las necesidades más avanzadas de cómputo para negocios. Además, proporciona hoy, la mejor plataforma de redes para el crecimiento a futuro.

¹⁵ El lector puede profundizar más recurriendo al CD de Service Pack de Windows NT 4.0 en Español de Microsoft.

Introducción¹⁶

En el complejo mundo del cómputo para los negocios, las empresas de hoy requieren un sistema operativo para redes que proporcione facilidad de uso, flexibilidad y servicios de comunicación integrados, ofreciendo, al mismo tiempo, el mejor rendimiento. Windows NT Está diseñado para trabajar con los sistemas que hoy se tienen y la tecnología futura. Además, las capacidades de comunicación optimizadas y funcionalidad Internet/Intranet de Windows NT Server 4.0, le ayudan a mejorar la comunicación tanto interna como externa, proporcionándole mayores capacidades de compartir información.

Cuenta con un rendimiento y arquitectura superior que Windows NT Server 3.51, Windows NT Server 4.0 mejora la facilidad de uso, instalación y administración, integrando la interfaz de usuario de Windows 95. Los administradores ahora pueden tener la misma interfaz de usuario en todas sus plataformas Windows de 32 bits¹⁷, teniendo como resultando menores requerimientos de entrenamiento y facilidad de migración de usuarios dentro de la familia Windows de sistemas operativos.

Microsoft ha añadido varios Asistentes para Administración en Windows NT Server 4.0. Estos Asistentes, diseñados principalmente para los administradores de redes menos experimentados, proporcionan una guía completa e interactiva para realizar las tareas administrativas más comunes.

En el área de flexibilidad, Windows NT Server 4.0 integra todas las características de escalabilidad, portabilidad y seguridad, sin sacrificar la velocidad o el tiempo de respuesta. Las mejoras en velocidad, rendimiento en compartición de impresoras y archivos, procesamiento de aplicaciones, Internet y acceso remoto, hacen de Windows NT Server 4.0 de las plataformas más poderosas y completas que existen.

Para los desarrolladores y "Web masters", la conectividad es la clave para producir aplicaciones que puedan integrarse tanto a través de redes locales como de Internet.

Windows NT Server 4.0 es considerado por Microsoft como el sistema operativo para redes que proporciona la máxima facilidad de uso y el mayor rendimiento aun en las necesidades de negocios más exigentes. Así también Microsoft promueve las siguientes características y cualidades.

- Fácil instalación, administración y uso

La administración cotidiana de servidores de red, se facilita con herramientas como el Administrador de Tareas y el Monitor de Redes. El Administrador de Tareas monitorea las aplicaciones e indicadores de rendimiento de Windows NT Server 4.0, proporcionando información detallada de cada aplicación y proceso que se está ejecutando en el sistema. Con esta información, los administradores pueden tomar acciones inmediatas para mejorar la confiabilidad y rendimiento del sistema.

¹⁶ Según propaganda publicitaria de Microsoft Service Pack para Windows NT 4.0, 1999.

¹⁷ Los sistemas operativos Windows 98 y Windows 95 no son considerados verdaderos sistemas operativos de trabajo a 32 bits, No obstante pueden ejecutar aplicaciones de 32 bits.

Otra herramienta muy poderosa para diagnóstico, es el Monitor de Redes, que examina el tráfico que entra y sale del servidor, incluso a nivel paquetes, y lo captura para su análisis posterior. De esta forma, es posible detectar y corregir fallas reales o potenciales en la red.

- Arquitectura flexible de servidor de red

Windows NT Server 4.0 proporciona integración transparente en una sola plataforma para su correo electrónico, servidor de archivos, bases de datos y comunicaciones. Trabaja con los sistemas que se tengan en la organización tales como: NetWare, UNIX, y mainframes o minicomputadoras IBM. Además, Windows NT Server 4.0 soporta más de 5,000 plataformas de hardware, compatible con todos los protocolos de red actuales, incluyendo: TCP/IP, IPX/SPX, NetBEUI, AppleTalk, DLC, HTTP, SNA, PPP, y PPTP. Soporta a los siguientes clientes: Windows 3.x, Windows 95, Windows NT Workstation, IBM OS/2, y Macintosh. El Servicio de Directorio de Windows NT (NTDS) puede soportar más de 25,000 usuarios por dominio y, literalmente, cientos de miles de usuarios por empresa. No importa que tan centralizada o distribuida sea la organización. NTDS permite crear un directorio que se ajuste exactamente a su organización y le permita administrar todos sus recursos, servicios y aplicaciones.

Windows NT Server es el único sistema operativo para redes con un servidor Web integrado: Microsoft Internet Information Server (IIS). El que IIS esté incluido en Windows NT Server 4.0 significa que la instalación y administración del servidor Web es tan sólo otra parte del sistema operativo. Además, con IIS, es posible administrar un sitio Web de forma remota, utilizando un explorador Web.

FrontPage, se incluye con Windows NT Server 4.0. FrontPage está diseñado para ambientes individuales y de grupo, permite diseñar sitios Web muy profesionales tanto a usuarios novatos como a desarrolladores experimentados.

El Microsoft Index Server, un componente gratuito disponible en el Web, indiza automáticamente el texto y las propiedades de los archivos, incluso HTML, que se encuentren en su servidor Internet, Intranet o de archivos. Los servicios de búsqueda de documentos le permiten expandir la funcionalidad de un servidor Web utilizándolo como herramienta para indizar contenido y búsqueda de archivos HTML o Microsoft Office.

Los desarrolladores de aplicaciones pueden tomar ventaja de la infraestructura Cliente/Servidor única conocida como Distributed Component Object Model (DCOM), para construir aplicaciones de alto rendimiento, seguras y distribuidas a través de la Internet. DCOM extiende la arquitectura COM permitiendo a los componentes interactuar a través de las redes mejorando la seguridad total y rendimiento del sistema operativo de red.

- Redes privadas virtuales con PPTP

Esto es utilizar la Internet como su propia red privada virtual para acceso remoto de bajo costo. Con el protocolo Point-to-Point Tunneling (PPTP) los usuarios remotos pueden conectarse a la red a través de cualquier proveedor local de acceso a Internet y establecer un túnel seguro para acceder su propia red, tal y como si estuvieran en su escritorio. PPTP proporciona este nivel de seguridad gracias a que soporta encapsulado de protocolos y conexiones RAS encriptadas.

- *Servicios Integrados de Acceso Remoto*

Windows NT Server 4.0 ofrece acceso remoto transparente a través del *Remote Access Service* (RAS). RAS proporciona acceso remoto accesible, seguro y fácil de usar con la mejor flexibilidad y escalabilidad. Además, RAS puede crecer modularmente desde uno hasta 256 usuarios simultáneos por servidor y varios servidores pueden combinarse para proporcionar acceso a una gran empresa o servicio de redes públicas. RAS está mejorado en Windows NT Server 4.0, con muchas características nuevas: Enlace Múltiple PPP que combina dos o más líneas no dedicadas para mejorar las velocidades de transmisión. Mejoras en auto-conexión y enlace a la red hacen a RAS más rápido y fácil de usar. Además, RAS soporta recuperación automática para copiado de archivos que le permite completar una transferencia de archivos más rápidamente en caso de pérdida de la conexión.

Resumen de Características y Ventajas

- *Interfaz Windows 95*

La interfaz Windows 95 está integrada con Windows NT Server 4.0, y a Windows NT Workstation 4.0, haciendo al servidor más fácil de usar y consistente con las otras plataformas de Windows 32-bits.

- *Asistentes para Administración*

Los Asistentes para Administración agrupan las herramientas más comunes de administración del servidor en un sólo lugar y lo guían por los pasos necesarios para realizar cada tarea.

- *Administrador de Tareas*

Proporciona información detallada sobre cada aplicación y proceso corriendo en el sistema. También muestra gráficamente el estado del servidor en uso de recursos.

- *Monitor de Red'*

Examina el tráfico de red que entra y sale del servidor, hasta nivel paquete, captura esa información para análisis posterior.

- *Microsoft Internet Information Server (IIS).*

IIS está integrado con Windows NT Server 4.0. Es el más rápido servidor Web sobre Windows NT Server. Además, incluye: Servidor Gopher, Servidor FTP Administrador de Servicios Internet y Conector Internet para Bases de Datos.

- *Microsoft Internet Explorer*

Integra los estándares HTML existentes con mejoras como vídeo en línea, gráficas de fondo, soporte a Secure Sockets Layer (SSL) y soporte a compras a través de Internet.

- *Microsoft FrontPage*

Permite tanto a no programadores y desarrolladores de aplicaciones no experimentados crear y administrar sitios Web de calidad profesional.

- *Microsoft Index Server*

Ayuda a los usuarios a encontrar información en servidores distribuidos en su Intranet corporativa.

- *Distributed Component Object Model (DCOM)*

Permite a las aplicaciones compartir componentes a través de redes incluyendo Internet.

- *Servicio de Acceso Remoto Multilink Channel Aggregation*

Permite a los clientes que accedan remotamente a Windows NT Server 4.0 combinar todas las líneas disponibles para incrementar el ancho de banda.

- *Point-to-Point Tunneling Protocol (PPTP)*

PPTP permiten a los usuarios extender la seguridad de las redes privadas a través de la Internet.

- *Multi-Protocol Router (MPR)*

Elimina la necesidad de utilizar ruteadores dedicados en redes pequeñas y medianas utilizando Windows NT Server 4.0 como una solución de bajo costo para ruteo entre redes. Proporciona ruteo de IPX/SPX, TCP/IP, y AppleTalk.

- *Telephony Application Programming Interface (TAPI) y Unimodem*

Proporciona las tecnologías requeridas por las aplicaciones de fax, el subsistema de mensajería de Windows (Cliente Microsoft Exchange), MSN, el servicio de información en línea del Microsoft Network y el Microsoft Internet Explorer.

- *Integración del Servidor Domain Name System (DNS) con el Windows Internet Name Service (WINS)*

Permite el acceso a recursos en la red o sobre Internet utilizando nombres DNS. Las características DNS incluyen: Una utilidad gráfica de administración Inter-operabilidad con el protocolo de notificación

- *Escalabilidad*

Soporta hasta 5,000 clientes de bases de datos concurrentes y bases de datos de 100 GB o más. Proporciona la mejor escalabilidad multiprocesador incluso con equipos de más de 4 procesadores. Soporta actualmente más de 2,000 aplicaciones comerciales con 4,000 más esperadas en los próximos 12 meses.

- *"Boot" remoto de Windows 95*

Permite iniciar remotamente sistemas basados en Windows 95 desde el servidor de la red.

- *Configuración de Políticas para Estaciones de Trabajo*

Controla configuraciones y ambientes en las estaciones de trabajo, Estaciones comunes proporcionando un ambiente en la organización.

- *Cryptography APIs*

Permite a los desarrolladores crear soluciones de encriptación propias.

Requerimientos

Intel y compatibles:

- Procesador 486/33 MHz o superior.
- 125MB de espacio en disco duro disponible.
- 16MB de RAM, 32MB recomendable.

Sistemas RISC:

- Procesador RISC compatible con Windows NT Server versión 4.0
- 160MB de espacio en disco duro disponible 16MB de memoria (RAM)
- Unidad CD-ROM.
- Adaptador de vídeo VGA, Super VGA o compatible con Windows NT Server 4.0.

Windows NT Server 4.0 soporta, directo de caja, hasta cuatro microprocesadores. Versiones de Windows NT Server 4.0 con soporte hasta para 32 procesadores están disponibles a través de los fabricantes del hardware.

Opciones de Red

Microsoft Windows NT Server 4.0 trabaja con:

- Microsoft LAN Manager
- Sistemas Macintosh
- DEC PATHWORKS
- Banyan VINES
- IBM LAN Server
- Redes IBM SNA
- La Internet
- Redes NFS
- Novell NetWare
- RAS a través de ISDN, X.25, y líneas telefónicas estándares
- Redes TCP/IP

Clientes soportados:

Windows 3. x
Windows for Workgroups
Windows 95
Windows NT Workstation
Apple Macintosh
MS-DOS
OS/2
UNIX**

**Requiere software de cliente de Visigenic Software, San Mateo, California.

3.4.4 Comparación de Microsoft Windows NT Server con Novell Netware 5¹⁸

Encontré esta comparación interesante que me pareció importante citar sobre todo por la representatividad que tiene Novell en el mercado de las redes LAN. Presento por razones de espacio y de profundidad que no abarcan este trabajo, solo una parte de esa comparación. Esta porción se vería complementada por una comparación desde el punto de vista de Novell. Sin embargo. Da pie para que el desarrollador tenga algunos puntos de referencia más sólidos y un porqué de la razón de implementar o no Windows NT como plataforma operativa para aplicaciones Cliente/Servidor dentro de una organización.

"Con la liberación de Novell del NetWare Versión 5.0, Microsoft ha tenido que comparar frecuentemente el Microsoft Windows NT Server con el Novell NetWare Versión 5.0. Microsoft considera que los clientes ya no buscan un sistema operativo para un servidor que sea único o con un propósito limitado, requieren de un sistema operativo de servidor que pueda tener un papel multipropósito". Por las razones que se especifican dentro de este documento, Microsoft considera que Windows NT Server satisfará los requerimientos de los clientes de hoy para un sistema operativo multipropósito, y lo continuará siéndolo en el futuro.

La siguiente tabla es una porción de la comparación detallada entre Windows NT Server y NetWare 5 que se encuentra en el CD Service Pack 4 de Microsoft para Windows NT, 1999.

Windows NT Server 4.0	Windows NT Server 5.0	Novell NetWare 5.0
Microsoft ha proporcionado un TCP/IP nativo con Windows NT Server desde la versión 3.1. Se han realizado muchas mejoras a lo largo de las sucesivas liberaciones.	Windows NT Server 5.0 ha actualizado constantemente TCP/IP para conectarse en red dentro de ambientes LAN y WAN ancho de banda amplio.	Novell está tratando de actualizar al ofrecer un TCP/IP nativo en NetWare 5. Microsoft ha estado enviando el TCP/IP nativo con Windows NT Server desde la versión 3.1.
No disponible	Para grupos de trabajo pequeños, Windows NT Server 5.0 configura automáticamente una red completa y la conecta a Internet a través de una interfaz de enrutador de marcación al hacer clic en una caja. El servidor define automáticamente las direcciones IP, establece DHCP y un proxy DNS, configura un NAT y configura la conexión por predeterminación. Los administradores necesitan saber el número telefónico, nombre del usuario y contraseña para la conexión ISP. No se requiere experiencia en IP.	Sin funciones comparables.
Disponible a través de terceros.	Proporciona soporte nativo a redes ATM y Gigabit.	El soporte ATM únicamente está disponible como add-on al Enrutador de multiprotocolos de NetWare que se adquiere por separado. Las conexiones ATM sólo están basadas en el enrutador; no hay ninguna opción para conectar el

¹⁸ Sitio esta breve comparación ya que Novell es uno de los principales fabricantes de herramientas de software para redes LAN y principal competidor de Microsoft en redes en dichas redes, para que el lector tenga una visión más completa puede encontrar una comparación exhaustiva en el CD Service Pack 4 de Microsoft 1999.

Windows NT Server 4.0	Windows NT Server 5.0	Novell NetWare 5.0
<p>Microsoft trabajó como parte del esfuerzo de proveedores múltiples para crear un Protocolo de Afinación punto a punto (PPTP) para proporcionar a los clientes una solución VPN segura y fácil de usar en el Windows NT Server 4.0. Como parte de Windows NT 4.0, el PPTP soporta conexiones VPN de cliente a servidor y de servidor a servidor</p>	<p>Windows NT Server 5.0 construye sobre el soporte PPTP y agrega los protocolos más utilizados incluyendo IPSec y L2TP para permitir soluciones VPN seguras. Los clientes pueden elegir entre estos protocolos según se requiera para los diferentes requerimientos de negocios. La negociación de protocolos y su integración dentro de una interfaz común esconde las distinciones al usuario final.</p>	<p>servidor NetWare directamente a ATM. Soporta Gigabit Ethernet</p> <p>Novell únicamente ofrece una VPN compleja, propietaria, maestro/esclavo de servidor a servidor. Se requiere de mucha experiencia en NetWare para configurar el servicio VPN. Las organizaciones deben adquirir el BorderManager, con un costo adicional de \$995 para el servidor y \$35 por usuario, para lograr este servicio VPN sólo de servidor a servidor.</p>
<p>Microsoft comprende que el acceso remoto a las computadoras corporativas es una de las principales prioridades de los clientes. Los Servicios de Acceso Remoto están incluidos en el Windows NT Server a partir de la versión 3.1 ofrece a los usuarios finales conexiones de marcación directa a las redes corporativas. Microsoft también reconoce que la marcación directa y la VPN resuelven un problema común al conectar a los usuarios remotos a las redes. Los usuarios y los administradores no deberían ser forzados a utilizar dos herramientas por separado para la marcación directa y para el acceso VPN. De este modo, Windows NT 4.0 los integra dentro de un servicio común y una interfaz de usuario común. Se soportan hasta 256 conexiones concurrentes.</p>	<p>Windows NT Server 5.0 continúa la integración de la marcación directa y VPN, y aumenta lo anterior con elecciones de protocolos VPN y un sistema global de administración de políticas. Se puede aplicar una política con base en el tipo de conexión, el usuario/grupo, la hora del día, el tipo de encriptado y otros para otorgar o negar el acceso.</p>	<p>Para el acceso básico de marcación directa, las organizaciones deben adquirir el BorderManager, con un costo adicional de \$995 para el servidor y \$35 por usuario, o el NetWare Connect, con un precio de \$140-\$395 por cada puerto de marcación con un límite de 128 sesiones concurrentes. No existe integración entre VPN y la marcación directa; de hecho, no existe un cliente VPN. BorderManager soporta PPTP.</p>
<p>El Windows NT 4.0 Option Pack incluye servicios que permiten la personalización de la interfaz del usuario y una administración de la libreta de direcciones distribuida. Los ISPs puede mantener sus POP de la libreta de direcciones independiente de los números de marcación directa. Cuando los usuarios conectan la libreta de direcciones a partir de los ISP(s) y la red corporativa estos son sincronizados y colocados en una vista común. Los usuarios no tienen que saber si el número es de marcación directo o VPN; el</p>	<p>Windows NT Server 5.0 continúa con esta capacidad única e incrementa la sofisticación de la personalización de la interfaz del usuario. Soporta números, contratos de licencia, archivos de ayuda, aplicaciones lanzadas, menús de conexión y otras que se ajustan a las necesidades del negocio.</p>	<p>Sin funciones comparables.</p>

Windows NT Server 4.0	Windows NT Server 5.0	Novell NetWare 5.0
software del cliente se conecta automáticamente según sea necesario.		

3.5. Requerimientos del hardware Intel para Sistemas Operativos actuales, Microsoft, del 32 bits.

	Windows NT Server Ver. 4.0	Windows NT Workstation Ver. 4.0	Windows 2000 Server	Windows 2000 Workstation (Profesional)
RAM mínimo (MB)	16	12	64	32
RAM recomendado (MB)	32	16	128	64
Espacio libre mínimo en disco duro (MB)	125	80	900	900
Espacio recomendado en disco duro para la instalación completa (MB)	150	100	1200	1000
Procesador mínimo	Pentium/100Mhz	386/25Mhz	Pentium II/166Mhz	Pentium II/166Mhz
Procesador recomendado	Pentium/133Mhz	486DX/100Mhz	Pentium II/233Mhz	Pentium II/233Mhz

3.5.1 Instalando Windows NT Server y Windows NT Workstation¹⁹

Windows NT es un sistema operativo de red y tiene la capacidad de *compartir recursos, administración de usuarios y herramientas de administración de red.*

Una red Windows NT esta dividida en dominios. Un dominio es una agrupación lógica de equipos que simplifican su administración

¹⁹ Para una descripción pormenorizada de la instalación de Windows NT Server y Windows NT Workstation. Referace a Microsoft Training and Certification Network Essential Cap 5 pag 339.

El software del servidor hace posible que los usuarios de otras máquinas compartan los datos y periféricos del servidor. Por lo general todos los equipos dentro de un dominio de Windows NT contienen software Cliente/Servidor. Si las estaciones de trabajo están actuando como clientes, tendrán incluido el software para actuar como clientes y como servidores.

El programa de instalación es una aplicación que realiza la tarea de colocar al sistema operativo sobre el disco de la máquina servidor y la instalación en este dependerá de varios factores:

- Del entorno donde se va a instalar
- Del tamaño de la red.
- De los tipos de trabajos que el servidor va a realizar en la red.
- Del tipo de sistema de archivos que va a usar el servidor.
- De las aplicaciones que estarán en el servidor.
- Y de como estará dividido el espacio en el disco del servidor.

El programa de instalación plantea una serie de preguntas para determinar los parámetros de instalación.

Nota:

Es conveniente auxiliarse de un experto en este sistema operativo para discutir este tipo de parámetros al igual que los siguientes antes de instalar MS Windows NT:

- Información de nombres para el servidor.
- Responsabilidades del servidor.
- Particiones.
- Configuración de tarjeta de red.
- Instalación de TCP/IP.
- Servicio de Nombres Internet de Windows (WINS).
- Servicio de DHCP.

Información de nombres para el servidor.- Es la identificación para el servidor y de la red que este controlará, que será el nombre que se le ha asignado al segmento de red, por ejemplo un nombre de grupo o dominio. De esta forma el Sistema Operativo puede identificar al servidor dentro de la red.

Responsabilidades del servidor.- Es la asignación de funciones que se le impondrán al servidor para determinarle las responsabilidades que asumirá dentro de la red las cuales pueden ser las siguientes:

- **Primary Domain Controler (PDC) Controlador Primario de Dominio.** El cual es responsable del mantenimiento de las cuentas de los usuarios.
- **Backup Domain Controler (BDC) Controlador Respaldo de Dominio.** Respalda al servidor responsable del mantenimiento de las cuentas de los usuarios.
- **Member Server o Servidor de aplicaciones o Stand Alone.**- Es el responsable de albergar las aplicaciones.

Después de que se ha instalado un PDC se pueden instalar varios servidores de Windows NT como reserva, los cuales son los BDCs.

Particiones.- Es la información que se requiere durante la fase de instalación que servirá para que Windows NT sepa como usar los discos duros. Windows NT debe saber en que porciones y en que modalidades los discos duros del servidor o de las workstations pueden dividirse. Estas áreas son llamadas particiones. Se pueden asignar para distintos propósitos. Por ejemplo podría crearse una partición para un cierto tipo de programas.

Configuración de tarjeta de red.- Durante la instalación también se solicitará que elija y configure una tarjeta de red y también deberá elegir uno o varios protocolos de comunicación para la red.

Instalación de TCP/IP.- Instalar MS TCP/IP sobre Windows NT permite la configuración de redes empresariales y la conectividad, ya que es uno de los mejores y más robustos protocolos que existen además de ser totalmente ruteable.

Servicio de Nombres Internet de Windows (WINS).- Los equipos pueden utilizar este servicio para poder identificarse entre sí este utiliza en vez de números el nombre de las computadoras.

Servicio de DHCP.- Es un servicio de asignación dinámica de direcciones IP para las estaciones cliente que las requieren para hacer uso de ciertos servicios dentro o fuera del dominio de un servidor NT.

3.5.2 Requerimientos de hardware para instalar aplicaciones Cliente/Servidor

Servidores

Una vez tomado en cuenta lo que se hablo en el apartado de "Concepto de Servidor" y en el apartado "Características de hardware de la computadora servidor"; en los capitulos anteriores de este documento solo resta decir que el servidor o los servidores deben de equiparse con tarjetas de red de la mayor calidad posible, ya que en estas tarjetas de red se efectuará un alto porcentaje del tráfico de la red. También discos duros de gran capacidad y de muy buena calidad al igual que la memoria RAM son los elementos principales con los que un servidor se equiparía. Y por supuesto más uno o varios microprocesadores acordes a la carga de proceso que se llevará acabo en el servidor.

Como vemos la capacidad final de un servidor no esta dada exclusivamente por la velocidad del microprocesador, ya que es un error muy frecuente por parte de quienes proyectan la adquisición del servidor mirar solamente la velocidad de trabajo del microprocesador como la cualidad determinante para su elección.

Estaciones de trabajo (Clientes)

Quando se trate de computadoras que exclusivamente van a realizar la función de clientes se le puede adaptar tarjetas de red de menor calidad y por consecuencia más baratas, pero sin descuidar que deban ser tarjetas de red fáciles de instalar y de configurar. La actividad que tendrán estas máquinas dentro de la red suele estar limitada y no requieren de manipular grandes volúmenes de tráfico. Aunque siempre es recomendable realizar un análisis de las aplicaciones con las que van a trabajar los clientes, para realizar la más adecuada elección en cuanto al equipamiento de tarjetas de red.

3.6 Programación en Windows.

Programar con éxito en ambientes Windows es el primer paso para poder desarrollar aplicaciones Cliente/Servidor confiables, es necesario conocer el funcionamiento básico de Windows.

Principiaremos por conocer los tres elementos claves y básicos sobre los cuales Windows sustenta su funcionamiento que son, ventanas, eventos y mensajes.

Ventana.- Es aquella región rectangular sobre el monitor con sus propios limites. Existen diferentes tipos de ventanas, las más comunes son los siguientes:

- Ventana explorador.
- Ventana Documento.
- Ventana de Dialogo.

Ventana explorador.- Es el tipo de ventana en la cual en una parte de esta se observa un aspecto de modo general y en la otra parte de la misma se hace un detalle de lo que señala la parte general. Ejemplo de este tipo de ventanas las podemos encontrar cuando desplegamos directorios mediante el "explorador de Windows".

Ventana Documento.- Es una ventana que despliega un texto y posee un juego de herramientas reducido o amplio para darle formato al mismo.

Ventana de Dialogo.- Es aquella ventana que da un aviso al usuario a la vez que sobre la misma se despliega una serie de opciones a tomar respecto de este aviso.

Eventos.- Son todos aquellos sucesos que puede desencadenar el usuario al estar utilizando una aplicación, por ejemplo movimiento del mouse, click o doble click del mouse, la presión del teclado, etc.

Mensajes.- Como su nombre lo indica son todos aquellos desplegados de información que el sistema operativo o las aplicaciones realizan sobre la pantalla de la computadora para mantener una interacción con el usuario.

3.7 Filosofía de diseño de Aplicaciones en Windows.

La filosofía de diseño de aplicaciones en Windows NT es muy semejante en todas sus versiones. Sin embargo, haré notar cuando se tenga que tener consideración especial para algunas versiones.

Las características que menciona a continuación son características que es recomendable que deba reunir cualquier aplicación independientemente de que sea o no Cliente/Servidor.

Al diseñar aplicaciones se debe de tener atención en que reúnan ciertas características que son:

- Intuitivos en buen porcentaje.
- Flujo de control determinado por el usuario.
- No sobrecargar la pantalla por gran número de objetos.
- Deben de proporcionar una retroalimentación al usuario.
- Deben de ser confiables.

Por otra parte debemos tener siempre en consideración dentro del ciclo de vida de un aplicación tres convenciones que son:

- Convenciones acerca del ratón.
- Convenciones acerca del teclado.
- Convenciones acerca de menú.

3.7.1 Convenciones acerca del ratón

Son las acciones de ratón que han llegado a estar de algún modo estandarizadas en aplicaciones Windows, en la siguiente tabla se presentan las acciones de ratón las comunes.

Convenciones acerca del ratón en Windows

Acción	Descripción
Click izquierdo	Activa un control, hace una selección o coloca el punto de inserción dentro de un bloque de texto.
Click derecho	Abre un menú emergente de conceptos de acción que se relacionan con el objeto en que se hace click.
Doble click	Activa un control, abre una aplicación o confirma una selección.
Ctrl-click	En el proceso de selección alterna la selección actual.
Mayúsculas-click	En el proceso de selección, extiende la selección hasta la posición del apuntador del ratón.
Arrastre	Mueve un control arrastrable, encierra un área para su selección; mueve la selección.

3.7.2 Convenciones acerca del teclado en Windows

Es conveniente tener siempre equivalentes en teclado de las acciones del ratón, ya que la idea es que puedan usar una aplicación aun con la falta o avería del ratón.

Tecla	Operación estándar
Alt	Se usa en combinación con otras teclas para crear teclas de acceso rápido.
Alt + F4	Cierra la aplicación actual.
Ctrl	Se usa con las letras para crear teclas de atajo.
Ctrl + F4	Cierra la ventana actual.
Ctrl + X	Atajo para terminar una aplicación.
End (Fin)	Se desplaza al extremo derecho de la línea.
Enter (Intro)	Activa el control actualmente seleccionado.
Esc	Cierra un cuadro de dialogo; sale del modo actual; cancela la selección actual.
F1	Activa el sistema de ayuda.
F10	Activa la barra de menú.
Home (Inicio)	Se desplaza al extremo izquierdo de la línea.
Tab	En un bloque de texto inserta un carácter Tab; en un cuadro de diálogo, mueve al campo o control siguiente en el orden del tabulador.
Mayúsculas + Tab	Invierte la operación del tabulador.

3.7.3 Convenciones acerca de menú en Windows

Las aplicaciones Windows emplean habitualmente unos cuantos menús estándar, además de cualquier selección de menú que pueda ser única para un programa en particular.

Convenciones acerca de menú en Windows

Menú	Opción de menú	Descripción
Archivo	Nuevo	Crea un documento.
	Abrir	Despliega el cuadro de dialogo "Abrir" para seleccionar un documento.
	Guardar	Guarda el documento actualmente abierto.
	Guardar como	Abre el cuadro de diálogo "Guardar como", para escribir un nombre, un tipo de archivo, etc.
	Imprimir	Abre un cuadro de dialogo "Imprimir" para establecer las opciones de impresión e imprimir.
	Salir	Termina la aplicación.
Edición	Deshacer	Invierte la última operación.
	Cortar	Corta la selección y la envía al portapapeles
	Copiar	Copia la selección y la envía al portapapeles.
	Pegar	Pega desde el portapapeles a la selección o a la ubicación actual.
	Buscar	Abre el cuadro de diálogo "Buscar" para localizar texto o

		nombres de archivos.
	Reemplazar	Abre el cuadro de diálogo "Reemplazar" para buscar y reemplazar
Ventana	Cascada	Acomoda las ventanas MDI dependientes abiertas en formato de cascada, como una baraja.
	Mosaico	Acomoda las ventanas MDI dependientes para que cada una este visible y del mismo tamaño.
	Dividir	Divide la ventana actual en dos o más porciones, también divide vertical u horizontal, dependiendo de la selección.
Ayuda ó ?	Contenido	Despliega una lista de los temas de ayuda disponibles en la aplicación.
	Buscar ayuda acerca de...	Abre el cuadro de diálogo "Ayuda acerca de ..." para localizar un tema en particular.
	Índice	Despliega un índice de los temas de ayuda de una aplicación.
	Acerca de...	Despliega el cuadro de diálogo "Acerca de ..." que proporciona el autor, los derechos reservados y la información de la licencia de la aplicación.

3.7.4 Que son las DDE y OLE

Desarrollar con componentes representa una opción muy interesante que debe ser estudiada y analizada sobre todo si se proyecta desarrollar una aplicación muy grande y poderosa. Ya que el desarrollo de una aplicación de esta magnitud requerirá de grandes jornadas de trabajo y un muy buen grupo de desarrolladores.

La tecnología ha tomado otro rumbo por la imperiosa necesidad de manejar grandes volúmenes de información distribuida en diversos puntos de la organización. El desarrollo de potentes aplicaciones, busca ahora la extensión de los módulos que la componen a través de las redes de cómputo.

Es necesario conocer estas dos tecnologías (OLE y DDE) para poder implementarlas dentro de las aplicaciones Cliente/Servidor.

DDE

Dynamic Data Exchange o intercambio dinámico de datos (DDE). Es una técnica que se introdujo desde la versión 3.0 de Windows. Sirve para que las aplicaciones puedan comunicarse entre sí. El DDE no se realiza mediante funciones, sino mediante mensajes que se envían a diferentes ventanas durante la ejecución de una aplicación. El Objetivo que persigue DDE es el intercambio permanente y automático de datos entre dos aplicaciones cargadas.

Como trabaja DDE

Su funcionalidad es parecida al clipboard de Windows. Las funciones más importantes de las conexiones DDE son las de cortar y pegar, solo que estas funciones no se realizan manualmente como es el caso del clipboard o portapapeles sino que las aplicaciones participantes lo hacen automáticamente para que esto se lleve a cabo es necesario que las aplicaciones creen la conexión o un vínculo hacia la aplicación. Así si los datos de una aplicación son cambiados mediante esta conexión permanente los datos pueden ser actualizados de manera automática en la otra aplicación. Esta funcionalidad esta disponible para todas las aplicaciones que trabajen sobre Windows. Sin embargo, no todas la ocupan.

La aplicación que suministra los datos se designa como Source o fuente y la aplicación que las recibe se denomina "destination" o destino. El DDE no esta limitado a sólo una conexión por lo que las aplicaciones pueden realizar varias conexiones de este tipo.

Existen tres tipos distintos de DDE

- **Hot link (comunicación activa).**- Es en donde todas las modificaciones se transfieren de manera automática de la aplicación fuente a la aplicación destino.
- **Cold link (comunicación pasiva).**- Aquí las modificaciones de la fuente sólo se comunican al destino después de realizar un comando.
- **Warm Link (comunicación pasiva).**- A la aplicación destino se le comunica que los datos han sido modificados pero la actualización sólo se lleva a cabo mediante el comando adecuado.

La conmutación entre tareas se realiza con la barra de tareas para luego dentro de cada aplicación establecer la comunicación mediante el pegado especial con vínculos

OLE

OLE es una excelente opción en el desarrollo de aplicaciones con componentes si se esta trabajando con sistemas operativos de Microsoft. "Microsoft con OLE 2.0 y Network OLE púntean la definición de estándares para el desarrollo de componentes principalmente en plataformas operativas Windows 95 y Windows NT consiguiendo que más de 370 fabricantes de software se establecieran bajo este estándar"²⁰

OLE proporciona los mecanismos de manejo de eventos, manejo de archivos y compartición de información, OLE es una parte integral de los sistemas operativos Windows

OLE tiene el inconveniente de no soportar desarrollos Orientados a Objetos. Los componentes basados en OLE no ofrecen herencia; la capacidad de mover datos y funciones desde objetos existentes hacia objetos nuevos para su uso dentro de la aplicación. Usted puede alterar componentes OLE usando agregados, pero usted no puede extraer las capacidades de un componente OLE a través de operaciones de programación Orientada a Objetos estándar.

Los controles OLE, son una mezcla de un servidor de automatización OLE, el cual permite que un control OLE exponga sus módulos de clase a otros controles OLE.

²⁰ Según nota editorial de La Revista Byte México Año 9 No. 96 1996

OLE 2.0 es una perfecta opción si usted desarrolla aplicaciones para Windows debido a que es parte de estos sistemas operativos.

OLE tiene la desventaja de que las aplicaciones son difíciles de construir debido a que OLE 2.0 tiene una interface sumamente compleja. Sin embargo. No debemos olvidar que OLE es hoy en día la tecnología estándar para el intercambio de datos entre aplicaciones de Windows.

Ventajas y desventajas de Usar OLE para la construcción de aplicaciones Cliente/Servidor

Ventajas

- Disponible hoy en día para desarrollos de tipo comercial.
- Soporte de componentes de terceros ampliamente distribuido.
- Soporte de herramientas de desarrollo extensivo y actualmente esta creciendo.
- Es una parte integral de los sistemas operativos Windows.
- Microsoft promueve exhaustivamente esta tecnología.

Desventajas

- Dificil curva de aprendizaje.
- Depende de una interface complicada.
- No soporta herencia.

3.7.5 Que son las DLL

El sistema operativo Windows está constituido por un conjunto de librerías DLL que contienen funciones y declaraciones que permiten realizar multitud de tareas constituyendo lo que se denomina API (Application Program Interface).

Las DLL (librerías de enlace dinámico) son archivos que contienen procedimientos que pueden ser invocados en tiempo de ejecución desde las aplicaciones. Entre las ventajas de su utilización respecto de los procedimientos compilados pueden destacarse las siguientes:

- Es posible invocarlos desde distintas aplicaciones, y reducir así el espacio ocupado por las mismas.
- Es posible actualizar su código sin necesidad de volver a compilar las aplicaciones que trabajan con ellos.

Declaración de procedimientos DLL

Para poder invocar un procedimiento DLL almacenado en un archivo externo desde el código de la aplicación es necesario declararlo. En esta declaración se suministra información al compilador sobre el archivo de procedencia, y el formato de los argumentos que se desean pasar y devolver.

3.7.6 Que son las APIs

Es la arquitectura de servicios que están disponibles para aplicaciones que son ejecutadas dentro de Windows, estas son genéricamente llamadas APIs (Applications Programming Interface) Interfaz de programación de aplicaciones. Las APIs proporcionan una gama de funciones y procedimientos a los desarrollos de aplicaciones, por ejemplo, acceder directamente a los objetos de interfaz de usuario los GUI (Graphics User Interface, para controlar los procesos de impresión, y acceder a funciones de espera de microprocesador, entre otras. Estas funciones o procedimientos están albergadas en forma de librerías dentro de archivos DLL (Dynamic Link Library) o (liga de función dentro de una librería dinámica dentro a nivel del directorio "c:\windows\system". Así se tiene una serie de funciones y procedimientos preconstruidos susceptibles de poder incorporarse a cualquier desarrollo de una aplicación. Estas funciones están escritas en lenguaje C pero pueden ser invocadas desde código de Visual Basic, además de otros lenguajes.

Microsoft ha organizado las funciones API en cuatro principales librerías DLL.

- **Kernel32.DLL.**- Es la DLL principal y realiza el manejo de la memoria, el control de la multitarea de los programas en ejecución, así como muchas otras funciones que afectan el funcionamiento de Windows.
- **User32.DLL.**-Contiene funciones que controlan los sistemas de menús, temporalizadores, comunicaciones del sistema de archivos y otros aspectos Windows.
- **Gdi32.DLL.**-La librería Graphics Device Interface proporciona las funciones necesarias para realizar el dibujo de objetos de la pantalla, así como la verificación y el manejo de las zonas que deben ser redibujadas.
- **Winmm.DLL.**- Proporciona funciones multimedia para el manejo del sonido, música, vídeo en tiempo real, muestreo, etc., esa librería sólo está disponible en sistemas de 32 bits. La librería equivalente para 16 bits es denominada **mmmsystem**.

Generalmente, estos archivos están situados en el directorio Windows/system. Sin embargo, existen también otras librerías DLL de menor tamaño, y cuya utilización es menos frecuente, las cuales proporcionan otros servicios más específicos a las aplicaciones (advapi32.dll, comdlg32.dll, netapi32.dll, etc.), la referencia sobre todo de las funciones incluidas en el API de Windows está incluida en la documentación del SDK de Win32. Sin embargo, debido a la gran complejidad que se puede alcanzar en la construcción de aplicaciones mediante la utilización de API, se incluyen archivos guía que contienen todas las declaraciones del API de Windows. Este archivo se encuentra en el directorio WINAPI de Visual Basic también se acompaña de una pequeña aplicación denominada Visor de texto API, la cual permite desplazarse por los nombres de las declaraciones de procedimientos, constantes y tipos de datos de los archivos de texto, esta aplicación muestra donde podemos ver el tipo de entradas, realizar búsquedas. Por ejemplo declaraciones, constantes o tipos. Una vez seleccionado el tipo de entrada, la lista de elementos disponibles mostrará los nombres de todos los elementos encontrados. Seleccionando un nombre y haciendo click en el botón agregar, el texto completo de la declaración del API correspondiente pasará al cuadro elementos seleccionados para luego poder pasarlo al cuerpo del código de la aplicación. Para eliminar una declaración del cuadro debe situarse sobre ella el cursor y hacerle click en el botón quitar.

3.7.6.1 Ventajas de utilizar APIs

Estas funciones y procedimientos son otorgados por el ambiente operativo de Windows, así el Diseñador/Desarrollador de aplicaciones, tiene la plena seguridad de que el mismo código funcionará sin problema en versiones futuras de Windows, otorgando de esta forma a los sistemas desarrollados, una característica, muy valiosa, la consistencia en diferentes versiones de Windows.

3.7.6.2 Desventajas de utilizar APIs

Debido a que las APIs son una gran lista de funciones, es necesario contar con un asistente para localizar la función deseada así como el archivo DLL que se invocara. Existen APIs tanto para 16 como para 32 bits. El programar con APIs implica tener una destreza mayor en el conocimiento y manejo del lenguaje.

3.7.7 Registro de Windows

Es de considerarse en la etapa de diseño si será necesario para el usuario conocer la forma de configurar el entorno de trabajo de la aplicación, esto posiblemente no será necesario para los usuarios inexpertos a quienes les sea instalada la configuración por defecto, pero existirán los usuarios avanzados que requerirán de una configuración sofisticada de su aplicación.

Deberemos conocer como manipular y conocer donde es conservada la configuración de una aplicación esto para poder controlar la configuración de la aplicación ya que de esto y de la total integración al ambiente de trabajo estará determinado su buen desempeño final.

El Registro de Windows

El registro de Windows esta compuesto por dos archivos, el USER.DAT y el SYSTEM.DAT ambos archivos son almacenados por default en el directorio de Windows.

El registro es manipulado por muchas aplicaciones tales como el Panel de Control y el Setup. REGEDT.EXE puede ser usado para manipular el registro directamente. RegEdt puede aprovechar el RPC para manipular remotamente el Registro sobre otro sistema Windows. Como con el registro de Windows NT, pero hay que ser precavido ya que cambiar el registro de Windows puede ser riesgoso, debido a que si usted hace un cambio incorrecto, puede dejar inutilizado hasta el Sistema Operativo. El registro no solamente almacena las condiciones de arranque del software sino también es la base completa del hardware, software, configuración y la información del usuario. Alojando toda esta información en una base de datos hace esto sencillo, para manejar y proporcionar gran seguridad y controlar el acceso a sistemas y a las opciones de configuración del usuario.

En MSDOS y en Windows 3.11 la configuración es mantenida en una variedad de archivos, tal como Config.sys o Win.ini. Windows NT almacena la información de configuración en solamente un archivo llamado "Registro". Una modificación a este archivo modificará la forma de trabajo del Windows NT y de sus aplicaciones.

El registro, por tanto, es una base de datos donde es almacenada toda la información de configuración del hardware y software de la computadora local, este registro controla a Windows NT mediante la apropiada información de inicialización, para arrancar aplicaciones y cargar componentes tales como controladores de dispositivos y protocolos de red.

La siguiente tabla describe el tipo de información contenida en el registro:

◆ Información del hardware instalado en la computadora.
◆ Información de los dispositivos controladores.
◆ Información de las aplicaciones instaladas.
◆ Información de los protocolos de red instalados.
◆ Información de los parámetros de las tarjetas de red.
◆ La información de las cuentas de los usuarios.

El registro ayuda a simplificar el soporte mediante una seguridad, debido a que es un juego estructurado de subregistros, que posibilitan a los administradores proporcionar soporte local y remoto.

Descripción de los componentes que hacen uso del registro (Están destacados en negritas los más utilizados por el desarrollador de aplicaciones)

Ntos kernel.exe, Windows NT Kernel	Windows NT kernel durante el levantamiento de Windows extrae información desde el registro, para determinar el orden en que se cargan los dispositivos controladores.
Device drivers	Los dispositivos controladores pasan los datos al registro y también reciben la carga y los parámetros de configuración del registro. Un controlador de dispositivo le dice al registro que recursos del sistema está utilizando, tales como hardware, interruptores y que canales DMA's (access direct to memory) se están utilizando.
Setup Programs	Durante la configuración de las aplicaciones o del hardware el programa de instalación puede agregar nueva configuración al registro. Este también puede consultar el registro para ver si los componentes requeridos en la instalación ya han sido instalados.
Hardware data	Durante el proceso de arranque de Windows información temporal del hardware es almacenada en el registro. En maquinas basadas en procesadores intel el archivo Ndetec.com se encarga de proporcionar esta información. En maquinas basadas en procesadores RISC es extraída del firmware.
Hardware profile	Los perfiles de hardware son una lista seleccionada, es decir, estos perfiles pueden ser creados por computadoras que alternan entre dos o más configuraciones de hardware. Cuando Windows arranca esta selección puede ser realizada.
Perfiles de usuario	Windows NT crea y mantiene configuraciones de usuario, como lo es el medio ambiente de trabajo del usuario, los cambios son almacenados en el registro.

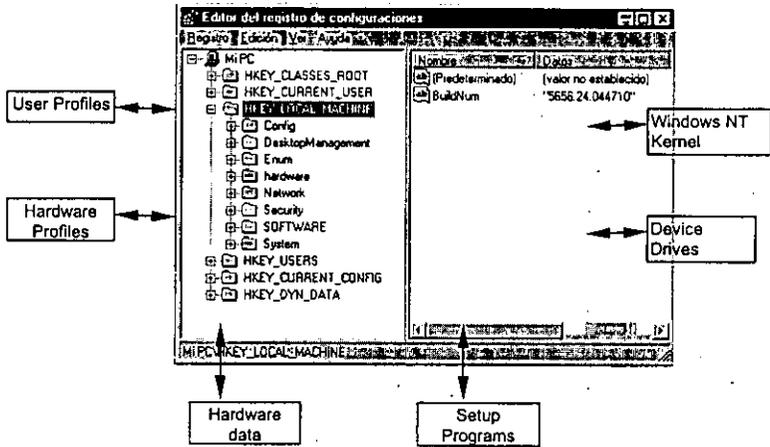


Fig. 3.6. El registro esta organizado en forma jerárquica.

Subtree	Es cada una de las 5 ramas que contiene el registro.
Hive	Es el cuerpo de las llaves, subllaves y valores.
Key, subkey	Son directorios o las carpetas y contiene llaves y subllaves
Values	Son los archivos de cada carpeta, están al final de la estructura jerárquica y pueden contener uno o más valores. Los valores pueden estar constituidos de tres partes el nombre, el tipo de dato, y el valor mismo (o parámetro)
	Tipos de valores de dato.
	REG_DWORD.- Solamente un valor es permitido con longitud de 1 a 8 dígitos hexadecimales.
	REG_SZ.- Solo es permitido un valor, se utiliza para almacenar una variable de texto.
	REG_EXPAND_SZ.- Almacena cadenas y puede contener variables reemplazables como lo son %SystemRoot%\...
	REG_BINARY.- Solamente admite un valor, es una cadena de dígitos hexadecimales y cada par representa a un byte.
	REG_MULTI_SZ.- En este tipo de dato son permitidos varios valores para una sola variable.

Subtree

Descripción

HEY_LOCAL_MACHINE	Contiene toda la configuración de datos de la computadora local. Esta información es utilizada por las aplicaciones, dispositivos controladores y por Windows NT para establecer la
-------------------	---

	configuración de la computadora. Los valores en esta llave determinan que dispositivos controladores y servicios serán cargados durante el arranque.
HKEY_USERS	Contiene 2 subllaves .default - Contiene la configuración por default, es usada cuando la pantalla de logon aparece.
HKEY_CURRENT_USER	Contiene datos acerca del usuario actualmente validado. Una copia es almacenada para cada cuenta de usuario que es validado a través de esa computadora, en el archivo Ntuser.dat.
HKEY_CLASSES_ROOT	Contiene información de todos los controladores.
HKEY_CURRENT_CONFIG	Almacena la información de como actualmente esta configurada la computadora.

La siguiente tabla describe las subllaves de la llave más importante HKEY_LOCAL_MACHINE

HARDWARE	esta subllave genera información volátil que es constituida cada vez que la computadora es iniciada. Esta subllave contiene información que cualquier aplicación puede consultar durante su instalación para determinar el tipo y el estado físico de los dispositivos conectados a la computadora local y no son salvados al disco.
SAM	El conglomerado de la base de datos SAM contiene el directorio de los usuarios grupos y políticas, derechos y perfiles de usuario, si una aplicación requiere consultar al SAM debe hacerlo a través de una apropiada API, el archivo sam.log esta ubicado en winnt_root\system32\config.
SECURITY	Contiene toda la información de seguridad de la computadora local, ningunas de la subllaves contenidas puede ser modificadas por aplicaciones. Las aplicaciones que deben consultar la información de seguridad deben usar APIs de seguridad. El conglomerado de la información de seguridad esta en el archivo security.log bajo el directorio

SOFTWARE	winnt_root\system32\config. Contiene información acerca del software instalado en la computadora local, que es independiente de cada usuario, ejemplos son el fabricante y la versión del software esta información esta ubicada en el archivo software.log, en la carpeta winnt_root\system32\config.
SYSTEM	Contiene información acerca de los dispositivos y servicios de la computadora local, esta información esta contenida en el archivo system.log y un respaldo en el archivo system.alt y ubicados en la carpeta winnt_root\system32\config.

El panel de control de Windows NT cuenta con programas que son usados para configurar secciones específicas del registro. Algunos programas del panel de control son usados para manipular el ambiente de la computadora, y, otros son usados para controlar la configuración de la computadora.

3.8 Proceso de comunicación de una Aplicación Cliente/Servidor

Describiéndolo en forma sencilla como ya lo vimos anteriormente, el proceso de comunicación de las aplicaciones Cliente/Servidor consiste de realizar una petición a algún servidor en la red desde la aplicación front-end; esta petición viaja a través de la red hacia el servidor o back-end. Este procesa la petición y regresa el resultado al front-end.

Algunas porciones de la aplicación corren en la PC cliente, esta PC cliente es la responsable de la presentación en pantalla de la mascarilla y la que recoge las entradas del usuario. La otra porción de la aplicación, el servidor, es el responsable de responder a las consultas, manejar la concurrencia de usuarios, seguridad, respaldo, restauraciones, proceso de transacciones entre otras actividades. Esto difiere de las aplicaciones basadas en servidores centralizados, donde toda la aplicación corre sobre el servidor.

Para que pueda llevarse a cabo con éxito el proceso de comunicación Cliente/Servidor se requiere que estén presentes los siguientes elementos:

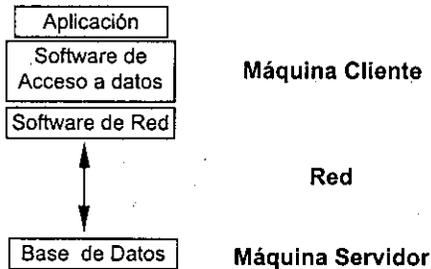


Fig. 3.7 Muestra los elementos que intervienen en el proceso de la comunicación Cliente/Servidor.

Aplicación.- Es la responsable de recoger las entradas del usuario a través de un GUI.

Máquina Cliente.- Es la PC donde una porción de la aplicación cliente se encuentra ejecutando, en los ambientes de PC puede ser para el caso de estudio de este trabajo una IBM PC o compatible.

Software de acceso a datos.- Es la capa de servicio que proporciona una interface directa para las aplicaciones, esta capa es un middleware, este juega un papel importante en el acceso a datos. Esta capa acepta datos extrae y actualiza requerimientos directamente desde la aplicación y los transmite a los largo de la red. Este middleware también es el responsable de regresar los resultados y códigos de error a la aplicación.

Red.- Es la conexión física del cliente al servidor.

Software de red.- Es el protocolo y parte de Sistema Operativo que puede ser el mismo del servidor, que permite comunicar al cliente con el servidor y viceversa.

Servidor.- Es el sistema físico donde reside el DBMS.

Base de datos.- Es el almacén de datos, los datos pueden existir en una variedad jerárquica o en DBMS relacionales

El proceso de comunicación Cliente/Servidor involucra al proceso de conectividad a la base de datos, esta es una cualidad que permite a las aplicaciones comunicarse con una o más DBMSs. La conectividad a bases de datos es un requerimiento si la aplicación usa archivos ISAM para aprovechar un entorno Cliente/Servidor o una conectividad tradicional a un mainframe. En este mismo sentido los requerimientos de conectividad a bases de datos han sido acelerados por el uso cada vez más frecuente del entorno Cliente/Servidor en las organizaciones.

Reglas de las Aplicaciones

Para interactuar con la fuente de datos una aplicación realiza los siguiente pasos:

- **Se conecta a la fuente de datos.** Esto especifica el nombre de la fuente de datos y alguna información adicional necesaria para completar la conexión.
- **Procesa una o más declaraciones SQL:**
 - La aplicación envía una cadena de declaraciones SQL en el buffer. Si la instrucción SQL incluye parámetros, la aplicación se encarga de establecer los valores de los parámetros.
 - Si la instrucción SQL regresa un resultado, la aplicación asigna un nombre de puntero para la instrucción.

3.8.1 Fallas en la Comunicación Cliente/Servidor

El desarrollo más importante de la tolerancia a fallos se ha hecho en los entornos Cliente/Servidor distribuidos. Un entorno Cliente/Servidor distribuido es un conjunto de computadoras (nodos), geográficamente dispersos, conectados por una red de comunicaciones. Los nodos son autónomos, no comparten memoria ni reloj, y se comunican entre sí con mensajes a través de la red. Cada nodo tiene alguna memoria no volátil y su propio software. Los entornos Cliente/Servidor distribuidos proporcionan redundancia de manera intrínseca.

Para el estudio de fallas un entorno Cliente/Servidor centralizado este se puede considerar como un caso particular de uno distribuido. Se pueden estudiar sus fallas:

- Por sus componentes físicos (modelo físico) o
- Por sus componentes lógicos (modelo lógico)

Comunicación Cliente/Servidor.

Si un proceso (cliente) quiere pedir un servicio a otro (servidor) y esperar bloqueado su respuesta puede usar una instrucción genérica Cliente/Servidor: envrec(datos, destino, resultado). El servidor recibirá la petición y después de realizarla enviará al cliente el resultado

Comunicación Cliente/Servidor con Fallos en Cliente/Servidor.

Cuando el cliente se desbloquea, sabe seguro que se ha realizado el servicio a pesar de fallos. Si durante la interacción existen fallos en la red o en el destino, el cliente se desbloquea recibiendo un resultado de error. Dependiendo de los protocolos la semántica de la interacción ante fallos será de:

"al menos una vez", "como mucho una vez" y "exactamente una vez"

Comunicación Cliente/Servidor "Al menos una vez"

La semántica de al menos una vez significa que el protocolo reintenta el envío ante fallos de la red y/o del servidor.

- Si el servidor devuelve un resultado, el servicio se ha realizado una o más veces.
- Si el servidor devuelve error, puede no haberse ejecutado el servicio 0, 1 o n veces.

Comunicación Cliente/Servidor "Como mucho una vez "

La semántica de como mucho una vez significa que el protocolo no reintenta el envío ante fallos de la red y/o del servidor.

Así, si devuelve un resultado, el servicio se ha realizado una vez. Si devuelve error el servicio se ha realizado en 0 o 1 vez. Para evitar fallos de la red el cliente puede reintentar diferenciando el primer mensaje de los sucesivos reintentos.

En este caso el servidor ha de estar capacitado para detectar peticiones repetidas: con número de sesión o con un bit que marca a un mensaje.

Comunicación Cliente/Servidor "Exactamente una vez "

Es la semántica por defecto y la ideal cuando teóricamente no hay fallos. Si hay fallos, el cliente reintenta, pues la acción ha de realizarse una vez, pero el servidor ha de detectar peticiones repetidas para no volver a ejecutarlas.

3.8.2 Comunicación Cliente/Servidor dentro de una aplicación

Puesto que COM (Modelo de Objetos Componente) está integrado en el sistema operativo, implementa la comunicación entre las aplicaciones cliente y los servidores que proporcionan componentes. Debido a que es un estándar, COM está incorporado a varios sistemas operativos y a varios compiladores.

En esta parte se describen los mecanismos que permiten la comunicación entre los clientes y los servidores, y el papel que desempeña el registro del sistema en esta comunicación.

3.8.2.1 Comunicación entre un cliente y un servidor en proceso

A nivel de sistema operativo, la velocidad de la comunicación entre un cliente y los componentes servidores varía si el servidor es un componente en proceso (una biblioteca DLL) o si es un componente fuera de proceso (un archivo .exe).

La comunicación entre un cliente y un componente servidor en proceso proporciona el intercambio de información más eficiente. Como el cliente y el componente servidor comparten el mismo espacio de direcciones, las llamadas entre ellos se hacen directamente.

3.8.2.2 Comunicación entre un cliente y un servidor fuera de proceso²¹

La comunicación entre un cliente y un componente servidor fuera de proceso es más compleja que con un servidor en proceso, ya que tiene que pasar los límites de dos procesos. La comunicación entre un cliente y un archivo ejecutable de un equipo remoto es aún más compleja.

Para administrar esta complejidad, los compiladores compatibles con COM insertan código proxy en los clientes con el fin de tratar las llamadas a los servidores y sus valores de retorno. Cuando un autor compila un servidor de componentes compatibles con COM, se inserta código de etiqueta para tratar las llamadas desde el código proxy de los clientes y los valores devueltos a los clientes.

A este proceso de tratamiento de los parámetros y sus valores de retorno se denomina encolado. El encolado es el empaquetamiento de todos los parámetros y sus valores de retorno, y su envío en ambos sentidos a través de los límites de los procesos.

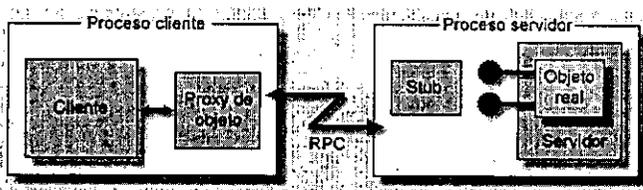


Fig. 3.8. La ilustración muestra el flujo de información desde un cliente a un servidor mediante código proxy y de etiqueta.

Esta implementación de la comunicación entre procesos es independiente de la plataforma ya que se basa en la especificación "llamada a procedimiento remoto de Entorno de Computación Distribuida" (DCE-RPC). Es posible realizar llamadas entre componentes compatibles con COM entre entornos Windows 95, Windows NT, UNIX y Macintosh, así como en otros entornos que acepten DCE-RPC.

3.8.2.3 COM, GUID y el registro del sistema

Cuando se compila un componente, el compilador crea identificadores únicos globales (GUID) para cada clase e interfaz pública del componente. Estos GUID son números de 128 bits generados mediante un algoritmo diseñado para asegurar su unicidad en todo el mundo.

²¹ Según Matering Visual Basic Versión 5 (CD).

Para que un componente esté disponible para los clientes, debe registrarse a sí mismo en el sistema operativo. Para hacer esto, actualiza la sección HKEY_CLASSES_ROOT del registro del sistema con información sobre él mismo, que incluye su ID. de programa (ProgID), que es el nombre del componente en formato legible, su ID de clase (ClassID), su ubicación y la ubicación de su biblioteca de tipos.

Cuando compila una aplicación cliente que hace referencia al componente servidor, los Id. de clase y los Id. de interfaz de los objetos creados por el componente se incluyen en el archivo ejecutable. En tiempo de ejecución, el cliente (junto con COM) utiliza la información del registro para llamar al componente servidor y solicitar a dicho componente que cree un objeto que devuelva al cliente una interfaz predeterminada.

3.9 Bases de datos utilizadas como fuentes de información Cliente/Servidor

La información ha llegado a convertirse en clave importante para que una organización sea competitiva, las organizaciones requieren acceder oportunamente a la información. Actualmente las organizaciones dependiendo de su giro están preocupadas por archivar con alto nivel de exactitud y eficacia la información de la organización tal como precios, control de calidad, análisis de mercado, planeación de la capacitación, inventario, servicios al cliente, cobranza o información sobre alumnos o profesores en una escuela, etc. Y al mismo tiempo los usuarios están demandando de mejores herramientas para acceder a la información con interfaces gráficas más potentes y rápidas. Es aquí donde toma real importancia una base de datos. Una base de datos es un archivo electrónico que almacena información de un aspecto en especial, esa información tiene relación entre sí y esta organizada en tablas, registros y campos.

En el diseño de una aplicación Cliente/Servidor, la base de datos juega un papel destacado, por que además de ser la fuente de datos que la aplicación desplazará a lo largo de la red, es también la razón por la que es necesaria una aplicación Cliente/Servidor.

3.10 Motores de Bases de Datos

El motor de bases de datos es el administrador de las bases de datos o DBMS, los hay relacionales (RDBMS) "Relational Data Base Manager System" y no relacionales (DBMS) "Data Base Manager System" son los administradores de bases de datos relacionales. El Administrador de bases de datos es el encargado de almacenar información a través de sus comandos y una estructura jerárquica organiza y clasifica esta información.

3.10.1 Concepto de OLTP

Los monitores de OLTP "On Line Transaction Process" (proceso de transacciones en línea) que agilizan el intercambio de datos en varias transacciones, por ejemplo la reservación de boletos en una línea aérea entre bases de datos y fuentes de datos a menudo proporciona facilidades de middleware. Muchas compañías de bases de datos como Grupta Technologies que, integran componentes de middleware dentro de sus líneas de productos de tal modo que cada producto puede interactuar sin problemas con sus iguales. Pero ni los monitores de OLTP ni las bases de datos con funcionalidad rudimentaria de middleware están diseñados exclusivamente como productos individuales que transportan los datos con fluidez a lo largo de una red.

3.10.2 Concepto de RPC

Las RPCs (Remote Procedure Call) constituyen la variedad de middleware mejor conocido y más ampliamente usado. Una RPC funciona como una llamada de programación, completando una sola tarea de proceso en una serie de pasos llevados a cabo por un programa. Por ejemplo una llamada de programación puede recolectar un conjunto de registros de clientes ese paso sería sólo una de las muchas funciones que una aplicación tendría que realizar para generar los recibos mensuales de una compañía.

Tradicionalmente las aplicaciones y sus llamadas de programación corren en una sola computadora. Pero con las aplicaciones Cliente/Servidor, sus registros probablemente no se encuentran ubicados en la computadora donde está corriendo la aplicación. Por tal razón es necesario un mecanismo que permita viajar a través de la red encontrar los datos y extraerlos hacia la aplicación cliente. Las RPCs ejecutan estas funciones en forma transparente para los usuarios, dando la sensación de que los datos estuvieron almacenados localmente.

Existen dos tipos primarios de RPCs y sistemas para el paso de mensajes.

El método más común por el cual el cliente solicita los servicios de un servidor es por medio de RPCs. Un RPC es un procedimiento que se ejecuta en otra máquina diferente a la que hizo la invocación del procedimiento, es decir, el cliente no ejecuta el procedimiento, sólo lo invoca desde el servidor.

3.10.3 Procedimientos Almacenados

Los procedimientos almacenados son rutinas cortas de proceso de datos escritas previamente, que ayudan a ciertos detalles del procedimiento de datos, estas rutinas deben de estar almacenadas en el servidor, para que cualquier cliente pueda usarlas, ahorran espacio en el disco donde las aplicaciones cliente están cargadas. El cliente hace llamadas a estas rutinas mediante la aplicación pero no forman parte integral de la aplicación sino que solo son invocadas. Varios clientes pueden llamar a un proceso almacenado en lugar de tener que incorporar la misma rutina en el código de cada aplicación cliente.

- Reducen el tráfico en la red por que con una sola llamada del cliente al servidor pueden iniciar varios procesos almacenados, que en otros entornos requerirían de realizar varias peticiones.
- Utilizando procedimientos almacenados se pueden realizar algunos de los procesos que normalmente realiza el cliente.
- Se pueden incorporar controles de seguridad para evitar que usuarios sin autorización ejecuten algunos de los procedimientos.

3.1.1. Conectividad en Bases de datos abiertas (ODBC)

Al hablar del desarrollo aplicaciones Cliente/Servidor es necesario tocar también el tema referente a ODBC(Open Database Connectivity) "Conectividad Abierta a Bases de Datos". Forzosamente una típica aplicación Cliente/Servidor que se desempeñe en una red LAN involucrará la conexión a una o a más bases de datos, por esta razón es que resulta de importancia mencionar este punto.

El ODBC es un API que permite proyectar un programa para la extracción de información de una base de datos. Cuando se escribe el código para interactuar con una base de datos, generalmente se tiene que agregar código que pueda comunicarse con el lenguaje propietario de una particular base de datos. Si desea que el código generado entable comunicación con tres bases de datos propietarias significará que usted debe de codificar su aplicación en tres diferentes lenguajes de bases de datos.

Cuando el diseñador de aplicaciones proyecta en su diseño la inclusión de ODBC usted necesita solamente comunicar al lenguaje ODBC las sentencias de SQL. De esta forma no requiere codificar sus aplicaciones en función de los diferentes DBMSs que utilizará.

El Open Database Connectivity (ODBC) es una tecnología de Microsoft y puede tal vez clasificarse como un middleware, aunque Microsoft no utiliza ese término, ODBC es una tecnología que permite que una aplicación se conecte a cierto número de bases de datos diferentes con sólo una interfaz de programación. En lugar de utilizar un código diferente para acceder a una base datos en particular, la interfaz de ODBC requiere solamente un código que pueda manejar los requerimientos de cliente, y para que pueda enviar información hacia este el ODBC consiste de un API y una interfase que proporciona los servicios o un manejador de ODBC que se coloca entre la aplicación y la red. El manejador ODBC puede también ser utilizado por IDAPI²². Microsoft según se sabe no tiene acuerdos con el consorcio IDAPI y por lo tanto no hay garantía de que su ODBC pueda trabajar con IDAPI.

La conexión abierta a datos (ODBC) es una interface estratégica de Microsoft para el acceso a datos en ambientes heterogéneos de sistemas administradores de bases de datos relacionales y no relacionales. Basados en la especificación CLI (Call Level Interface) llamada al nivel de interfase de SQL, ODBC proporcionando acceso abierto a una variedad de bases de datos de computadoras personales, minicomputadoras y mainframes propietarias. ODBC alivia la

²² IDAPI "Integrate Database Application Programing" (Interfase de programación para aplicaciones de bases de datos integradas) es un consorcio que ha creado una norma, esta formado por algunos fabricantes de software que han unido esfuerzos para establecer esta norma, la cual soporta las plataformas Windows, DOS, Netware y OS/2, y compañías como Oracle, Grupta y Banyan. IDAPI se Integra por tres partes; el cliente API, la maquinaria y el manejador API.

necesidad de independencia de los vendedores de software y compañías desarrolladoras para aprender múltiples interfaces de programación. Proporciona una interfaz universal de acceso a datos, posibilita que las aplicaciones Cliente/Servidor puedan acceder concurrentemente a ver y modificar información desde diferentes bases de datos. ODBC es el componente medular de Microsoft WOSA (Windows Open Service Architecture)

ODBC ofrece capacidades de misión crítica para el proceso de transacciones Cliente/Servidor en línea (OLTP).

Ventajas

- ODBC permite el acceso a datos en más de una ubicación de almacenamiento (por ejemplo en más de un servidor) desde dentro de una misma aplicación.
- ODBC permite el acceso a datos en diferentes tipos de DBMS (BD2, Oracle, MS SQL Server, DEC Rdb, Apple DAL y Dbase) desde una misma aplicación.
- ODBC simplifica enormemente el desarrollo de aplicaciones Cliente/Servidor, ya que es más sencillo para los desarrolladores de aplicaciones proporcionar el acceso a los datos en sus aplicaciones.
- ODBC es una API portable, habilita la misma interface y la tecnología de acceso.
- Aísla a la aplicación de los cambios subyacentes de red y de las versiones de DBMS. Las Modificaciones a los transportes de red, servidores y DBMS no afectarán a las aplicaciones que utilicen ODBC.
- Permite a las corporaciones proteger sus inversiones existentes en DBMSs, en desarrollos de aplicaciones Cliente/Servidor y en habilidades de DBMS que haya adquirido su personal.

3.11.1 Cómo trabaja ODBC

ODBC define un API. Cada aplicación usa el mismo código. ODBC es también considerado un manejador de controlador situado entre la aplicación y los controladores. En Windows, el Driver Manager y los controladores son implementados como un DLL.

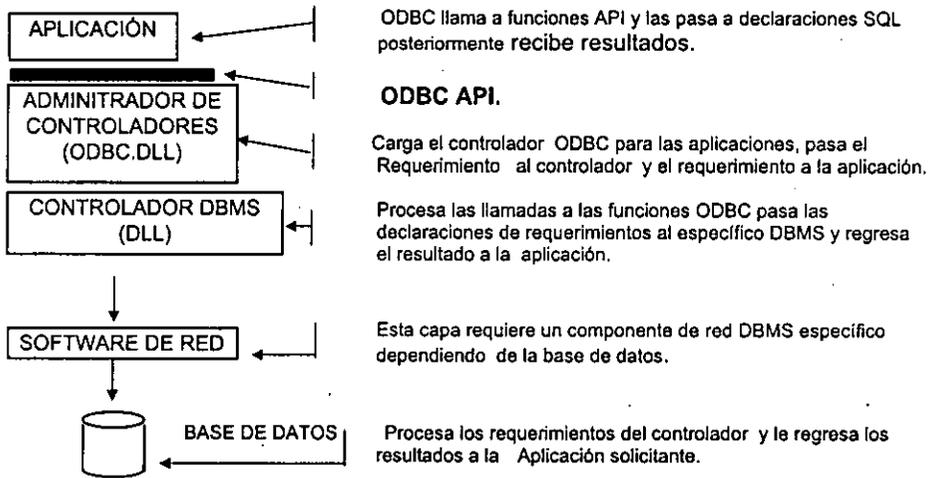


Fig. 3.9. Muestra como trabaja el ODBC.

Explicando la figura...

La aplicación llama a las funciones ODBC para conectarse a una fuente de datos, envía y recibe datos y desconecta.

El Administrador de controladores proporciona la información a una aplicación como una lista de datos disponibles, carga controladores dinámicamente como ellos son necesitados y checa el estado de la transacción.

El Controlador DBMS situado entre la aplicación y la red procesa las llamadas a funciones ODBC, maneja todos los intercambios entre una aplicación y específico DBMS y puede traducir la sintaxis estándar de SQL. Todas las transacciones son responsabilidad de este controlador.

Las aplicaciones no están limitadas a comunicarse a través de un solo controlador. Una sola aplicación puede realizar múltiples conexiones, cada una a través de un controlador diferente. O también una misma aplicación puede realizar múltiples conexiones a una misma base de datos.

Para acceder a un nuevo DBMS, es necesario instalar un controlador para el nuevo DBMS. No es necesario una nueva versión de la aplicación para acceder al nuevo DBMS, esto repercute en beneficios para el usuario final y en reducción de costos de soporte y desarrollo para la organización.

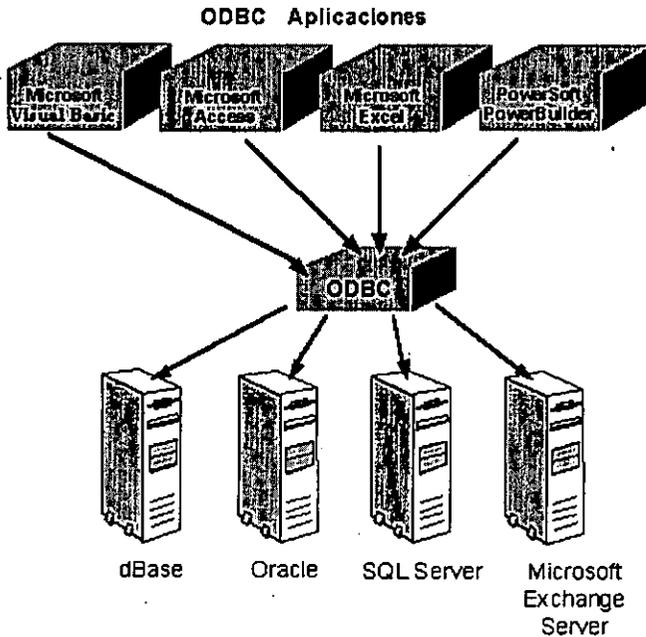


Fig. 3.10 Interacción de diferentes aplicaciones con diferentes servidores a través de ODBC

Para utilizar los controladores ODBC Desktop Database, se ha de disponer de:

- Un PC compatible con IBM.
- Un disco duro con 6 MB de espacio libre.
- Un mínimo de 16 MB de memoria de acceso aleatorio (RAM).

Requisitos de software

Para tener acceso a datos con el controlador ODBC, debe disponer de:

- El controlador ODBC.
- El Administrador de controladores ODBC de 32 bits, versión 3.0 o posterior (ODBC32.DLL). Observe que este archivo se incluye en los controladores ODBC Desktop Database.
- Microsoft Windows 95 o posterior, o Windows NT 3.51 o posterior.
- El tamaño de pila para una aplicación en la que se utilice un controlador de Microsoft ODBC debe ser, al menos, de 20K.

3.11.2 WOSA (Windows Open Service Architecture)

Debido a la ausencia de una manera formal de conectarse de aplicaciones front-end a varios servicios back-end, los desarrolladores de aplicaciones se han visto obligados a incorporar soporte de vendedores específicos de APIs dentro de sus aplicaciones. Y para que una aplicación pudiese soportar servicios adicionales; solo había dos caminos, se construía una nueva aplicación o se modificaba la existente, a fin de adaptarla a las diferentes APIs, este proceso es a la vez una labor muy intensiva y cara:

WOSA proporciona un solo nivel de interface (acoplamiento) mientras oculta la complejidad de los ambientes heterogéneos a los usuarios finales y a los desarrolladores de aplicaciones, la ventaja de usar una interface WOSA es que, una aplicación Windows de escritorio no necesita contener código especial para algunos de los tipos de red en uso en la organización, ni para los diferentes tipos de computadoras en la organización o para los diferentes servicios de back-end. Incluso si en la red, las computadoras o los servicios llegarán a cambiar, la aplicación no requiere ser modificada, en resumen WOSA posibilita a las aplicaciones basadas en Windows poder conectarse a todos los servicios a través de múltiples entornos computacionales.

Dentro de Windows una vez que los usuarios aprendieron como usar una aplicación, ellos pueden aprender rápidamente otras aplicaciones similares. WOSA presenta a los desarrolladores de aplicaciones Cliente/Servidor una sola interface para comunicarse con servicios back-end, en vez de tener que aprender un diferente API para cada implementación de un servicio, con WOSA solo es necesario aprender solo una API para todas las implementaciones, aunado a esto cuando un servicio existente es modificado o reemplazado la aplicación front-end no es afectada.

3.12 Concepto de Middleware

Middleware es otra palabra que se añadido al léxico siempre en aumento de los términos de computación, en cualquier otra tecnología naciente, no existe un concepto exacto de lo que constituye el middleware, ni siquiera en lo que significan aunque la definición es aún nebulosa, los fabricantes intentan especificar una norma.

El hecho de que middleware se encuentra en una gran cantidad de bases de datos en las compañías, queda evidenciado por el IDAPI. De esta forma middleware es la porción que se encuentra entre las APIs, la maquinaria y el "solicitante/contestante", el cual permite que la aplicación acceda remotamente a la base de datos sin que el manejador soporte suficientemente a la función.

El software de middleware puede venir de cualquier fabricante que desee desarrollar esa tecnología. No tiene que ser necesariamente desarrollada por alguno de los signatarios de IDAPI. Middleware se entiende como aquella actividad que sucede entre las aplicaciones, las bases de datos y las bibliotecas proporcionadas por el fabricante, o dicho en menos palabras es un componente que va entre el sistema operativo y la aplicación que permite al desarrollador aislarse de las peculiaridades del Sistema Operativo y del ambiente de redes.

Los fabricantes de una aplicación deben proporcionar el API que las aplicaciones pueden

utilizar para lograr la funcionalidad del IDAPI. Por otra parte IDAPI se interconecta con los manejadores que vienen de fabricantes de bases de datos.

Middleware es un software para las arquitecturas Cliente/Servidor que apoya la circulación de datos entre aplicaciones que están corriendo en una computadora y en la red. El middleware proporciona una serie de APIs estándares que ahorran tiempo de desarrollo a los programadores; puesto que no dedican tiempo en modificar las aplicaciones para adaptarlas a los diferentes protocolos de red, y también se ahorra tiempo que se dedica en probar estas adaptaciones con el protocolo o con el sistema operativo de la red, así este tiempo ahorrado puede ser invertido en las etapas de desarrollo de la aplicación. Por lo que un software de middleware puede reducir los cientos de verbos y parámetros de un protocolo a unos cuantos. Significa total transparencia del protocolo y medio de comunicación que utilizan las estaciones cliente para hablar con sus servidores. El desarrollador de aplicaciones Cliente/Servidor compra este software de middleware y lo incorporan a la aplicación Cliente/Servidor que está desarrollando.

Una ventaja de esto es que cada vez que sale una nueva versión del sistema operativo la empresa que produce el software de middleware se encarga de asegurar que este middleware sea actualizado para que el desarrollador de aplicaciones lo pueda incluir en su aplicación sin siquiera saber que cambio con la nueva versión del SO. Además, la aplicación es más robusta y eficiente si utiliza un middleware y también puede concentrar toda su atención y recursos intelectuales a las etapas de desarrollo de la aplicación.

Actualmente están disponibles dos tipos primarios de middleware

- RPCs (llamadas remotas a procedimientos).
- Sistemas para el paso de mensajes.

3.12.1 Como funciona un software de middleware

Una manera sencilla de comprender el funcionamiento del middleware; es visualizarlo como un correo electrónico entre la parte de la aplicación que corre en la estación cliente y la parte que corre en el servidor, como ocurre cuando un usuario utiliza un servicio de "mail"²³ electrónico el usuario no tiene idea de como viaja su mensaje a través de las redes que lo conectan con el destinatario, las aplicaciones Cliente/Servidor escritas sobre un middleware intercambian información obviando las complejidades que permiten que se establezca esta comunicación.

²³ Aunque es conocido y la mayoría ha hecho uso y conoce el funcionamiento básico del correo electrónico. Diré que el enviar o recibir un mail implica una relación Cliente/Servidor muy específica, donde el mensaje es escrito en un GUI que lleva el mensaje y especifica el destino; este es enviado al servidor el cual determina la ubicación del destino y lo envía al destino o lo retiene en un apartado espacial para cada usuario mail, así dependiendo del sistema de correo, el usuario consultará sus mensajes recibidos en su máquina o accederá al servidor para ello.

3.12.2 Ejemplo de un software de middleware

IBM ha desarrollado un middleware denominado MQ (Message Queuing). Cuando una estación cliente desea enviar un mensaje a su servidor, simplemente utiliza una instrucción denominada PUT_QUEUE para poner el mensaje (en cualquier formato) en la cola de salida del MQ, de ahí en adelante MQ es el responsable de hacer llegar el mensaje a su destino. El proceso de comunicación que maneja MQ trabaja normalmente de una manera asíncrona. Es decir, la aplicación simplemente ejecuta el PUT_QUEUE y recibe de nuevo control para ejecutar otras instrucciones. Sin embargo, se pueden ejecutar las llamadas de MQ de manera síncrona, donde la aplicación no recibe el control de vuelta hasta que el mensaje no haya llegado a su destino. La forma de trabajar asíncrona; resulta adecuada cuando la disponibilidad de las líneas de comunicación no es buena o cuando los servidores reciben volúmenes de transacciones muy altos en horas pico, el que escribe la aplicación no tiene que preocuparse si hay o no hay línea, o si la red está trabajando con SNA, TCP/IP, Netbios o DECNET, cuando el mensaje llega a su destino MQ despierta un proceso que le entrega a la aplicación de destino a través del uso de la instrucción GET_QUEUE. Estas instrucciones existen para compiladores C, PL/1 y COBOL y son parte de un sencillo pero robusto API's que permite incorporar fácilmente código ya existente, convirtiendo a una aplicación aislada en una aplicación Cliente/Servidor. MQ existe como middleware para 19 sistemas operativos (Windows NT, OS/400, SCO, HP-UX, SunOS, AIX, MVS, VSE, VMS, entre otros), ya que el API es idéntico para todos y cada uno de los sistemas operativos, la aplicación solo tiene que ser recompilada para poder portarla a otra plataforma de sistema operativo.

Debido a que MQ es el centro de la estrategia de middleware de IBM, se tiene la garantía de que nuevas versiones saldrán al mercado juntamente con las nuevas versiones de los sistemas operativos, garantizando así al desarrollador que su aplicación Cliente/Servidor continuará funcionando en versiones posteriores del sistema operativo.

3.12.3 Futuro del middleware

En un futuro los diferentes productos de middleware podrían desaparecer como tal y ser parte integral de los sistemas operativos, de la misma forma que TCP/IP ha llegado a ser parte integral del sistema operativo Unix.

3.13/Adecuación de la plataforma de hardware

Si se cuenta con equipo de hardware antiguo en la organización. Habrá que realizar un estudio y un análisis sobre las repercusiones económicas y sobre los objetivos que se desea alcanzar a fin de poder determinar si podrá soportar las aplicaciones Cliente/Servidor que se desean utilizar; la siguiente guía puede servir de base inicial para el estudio si se pretende entre otras metas contar con una plataforma de hardware actualizada para que pueda soportar aplicaciones Cliente/Servidor sin ninguna modificación en un futuro próximo.

Esta guía es aplicable en condiciones donde el incremento del ritmo de trabajo en la organización no será muy grande o permanecerá constante, esto también se contempla dentro del estudio y análisis.

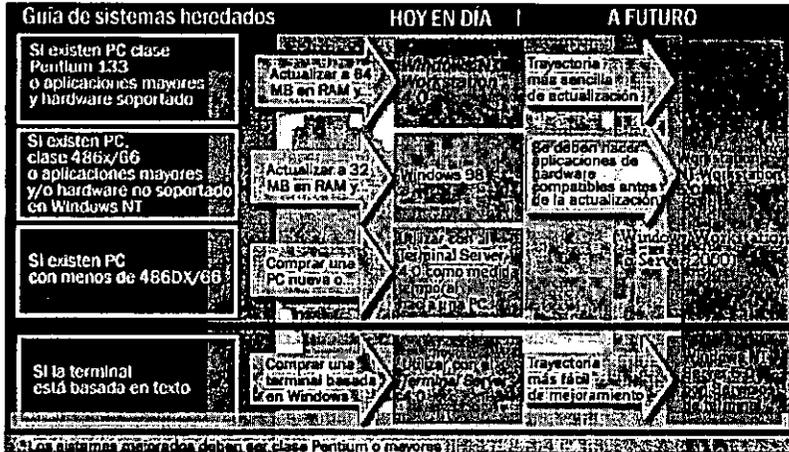


Fig. 3.11. La guía (sugerida por Microsoft) de arriba indica los pasos y acciones necesarias para actualizar los sistemas de escritorio existentes a fin de hacerlos capaces de soportar Windows NT como sistema operativo.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

4.1 Introducción

Realmente no existe una metodología aprobada formalmente y aplicable universalmente al diseño de aplicaciones Cliente/Servidor, el método que explico aquí, es solo uno de varios que existen.

El método recomendado para el desarrollo de aplicaciones Cliente/Servidor de calidad, sigue lineamientos del método tradicional de desarrollo en "cascada" o "Top Down"²⁴ junto con técnicas muy particulares que son solamente aplicables en los entornos Cliente/Servidor; debido a que los bancos de información residen en más de una computadora y la, o las aplicaciones desarrolladas residirán en diferentes sitios de la red. El método de análisis y diseño estructurado en cascada, reúne una serie de requerimientos los cuales se analizan y se subdividen de manera progresiva. Después se construye un diseño mediante los resultados del análisis. El diseño principia en un nivel abstracto y se divide sucesivamente en niveles más concretos hasta que se escribe el código. Así los elementos que intervendrán en el proceso que la aplicación este ejecutando, se convertirán en parte de una red virtual particular, estos elementos son dependientes de la aplicación por el tiempo que dure trabajando el proceso. La aplicación será capaz de crear esas redes virtuales dentro de una red física y el número de interconexiones que se generen de esta forma estará supeditado por las solicitudes de información que la aplicación realice sobre la red.

El proceso de desarrollo de una aplicación Cliente/Servidor se puede convertir en un proceso complicado y demasiado tardado si se sigue solamente el método tradicional de desarrollo de aplicaciones, y puede ser muy deficiente la calidad de la aplicación definitiva.

²⁴ Top Down es la técnica tradicional de desarrollo que se sigue para el desarrollo de aplicaciones, involucra al método estructurado de programación cuyo objetivo es utilizar estructuras de control: secuencia, repetición y selección (While, For, IF, Repeat) y sigue una secuencia de instrucciones de arriba a abajo.

4.2 Desarrollo de la Metodología para Aplicaciones Cliente/Servidor

Existen cuatro pasos esenciales en una organización para implementar un entorno de aplicaciones Cliente/Servidor.

1. Identificar los motivos de interés por la implantación de la plataforma Cliente/Servidor dentro de una organización.
2. Analizar el flujo de la información.
3. Establecer la infraestructura humana para el desarrollo de la aplicación.
4. Establecer la arquitectura de la aplicación.

1. Identificar los motivos de interés por la implantación de la plataforma Cliente/Servidor en una organización.

Conocer los intereses reales que tiene una organización para cambiar hacia un entorno de cómputo Cliente/Servidor nos da información precisa para la toma de decisiones futuras y también posibilita poder crear un plan de cambio paulatino donde se contemplan factores inherentes al proceso de migración como son:

- Planeación de costos de infraestructura de hardware (Accesorios de red, PC's instalación de cableados).
- Planeación sobre costos de software.
- Planeación sobre costos de servicios profesionales.
- Planeación sobre requerimientos y costos de infraestructura física.
- Planeación sobre costos de capacitación de personal.
- Planeación sobre costos y requerimientos de personal especializado para proporcionar soporte técnico en hardware y software para el nuevo ambiente.

Una vez que los motivos de la organización fueron identificados la tecnología puede ser orientada hacia los objetivos que quiere alcanzar la organización. Esto es a través de la identificación específica de los requerimientos de información. Para identificar los requerimientos es necesario examinar la estructura organizacional y el flujo lógico de trabajo dentro de la organización. Esto ayuda a diseñar las redes de comunicación y a determinar como los datos serán distribuidos.

2. Analizar el flujo de la información Cliente/Servidor.

En el diseño de una aplicación Cliente/Servidor se debe pensar en la localización y en el flujo de la información. Para poder direccionar la información, es necesario un conjunto de habilidades. La arquitectura de la red debe de comprender la representación física y lógica de la red, y esas restricciones pueden ser colocadas en las aplicaciones. El análisis de las bases de datos debe de ser consistente con el diseño de las estructuras de datos.

Para el total entendimiento del flujo de la información a través de la aplicación el intercambio de datos debe de ser separado internamente en conversación síncrona y asíncrona. Mientras que el intercambio síncrono proporciona inmediata información; extrae y devuelve la información, esto esclaviza a la aplicación mientras esta espera. La conversación asíncrona facilita la transferencia de grandes cantidades de datos, pero requiere cuidadosa planeación del almacenamiento y enrutamiento o encarrilamiento del regreso de datos. Tienen sus ventajas y desventajas ambos métodos; la mayoría de las aplicaciones Cliente/Servidor ofrecen mejor servicio teniendo una combinación de ambas formas de comunicación.

La implementación de aplicaciones Cliente/Servidor se convierte en una serie de desafíos que necesitan ser conocidos muy de cerca desde el punto de vista de definiciones largas y términos cortos.

3. Establecer la infraestructura humana para el desarrollo de la aplicación.

Cuando las características de la organización han sido determinadas puede iniciarse el diseño del sistema. El primer paso es seleccionando el apropiado personal para el proyecto. Un proyecto que involucre arquitectura Cliente/Servidor requiere de diferentes expertos que conozcan acerca del desarrollo de aplicaciones en los ambientes actuales de hardware de la organización. También son necesarios expertos para establecer redes y bases de datos. La más grande aplicación, la más importante aplicación es lograda por tener a los correctos expertos; si una red o una base de datos no esta bien diseñada y no es mantenida correctamente, la aplicación puede desempeñarse pobremente o puede jamas trabajar apropiadamente.

4. Establecer la arquitectura de la aplicación.

Componentes de las aplicaciones Cliente/Servidor

Las aplicaciones Cliente/Servidor están formadas por tres componentes básicos: redes, bases de datos y herramientas de desarrollo. El primer paso que lleva a tener una firme arquitectura es seleccionar productos de red y sistemas manejadores de bases de datos probados. Esas dos tecnologías proporcionarán la estructura en la cual se construirán las aplicaciones Cliente/Servidor.

- *Establecer la arquitectura de red para la aplicación.*

Seleccionar los productos de red puede resultar un proceso complejo, que consume cierto tiempo. Para el desarrollo de aplicaciones Cliente/Servidor se debe de poner especial atención sobre la capa de transporte²⁵ de la red LAN. Los productos de red deben de proporcionar una GUI para las aplicaciones. Este acceso es provisto desde la capa de transporte. La capa de transporte es importante para algunas aplicaciones que son ejecutadas sobre la red por que estas habilitan a los datos para que estos sean pasados entre dos aplicaciones.

²⁵ La capa de transporte según el modelo OSI es la capa 4. El nivel de transporte asegura que los paquetes se entreguen sin errores, secuencialmente y sin perdidas.

- *Establecer la arquitectura DBMS de la aplicación.*

Pensar en la selección de un manejador de bases de datos es crítico debido a que a largo plazo, de este dependerá en gran medida del éxito o fracaso de las aplicaciones Cliente/Servidor que se ejecuten. El producto de base de datos seleccionado debe de poder permitir el acceso a datos simultáneo a múltiples usuarios. Además de soportar múltiples usuarios. Un entorno Cliente/Servidor puede tener altos volúmenes de transacciones. Así que un manejador de bases de datos deberá manipular altos volúmenes de datos, por lo cual se debe de pensar en un manejador de bases de datos suficientemente robusto.

4.3 Decida como los Datos estarán Distribuidos.

Otra consideración importante cuando estamos diseñando la **plataforma** Cliente/Servidor es determinar como el poder del proceso y los datos serán distribuidos. El lugar del procesamiento estará basado sobre las necesidades organizacionales, sistemas existentes y el crecimiento en futuras aplicaciones. En muchos medios ambientes corporativos, el proceso en cómputo abarca desde microcomputadoras, minicomputadoras hasta mainframes, la clave esta en determinar la mezcla más efectiva de plataformas para el proceso de datos y aplicaciones. Muchas implementaciones Cliente/Servidor son destinadas a los ambientes PC. Para efectivamente integrar las workstation o PCs existentes dentro de una expandible arquitectura, el más bajo común denominador debe ser usado para determinar la distribución del proceso, esto revelará los cuellos de botella en el flujo y manipulación de la información entre las múltiples plataformas. Prestar atención a los ambientes de sistemas mainframes es un punto importante ya que estos extensos sistemas actuando como servidores de proceso a gran escala, pueden ser incluidos en una arquitectura Cliente/Servidor en una LAN como almacén de datos y servidor de aplicaciones, esta es una excelente forma de darles un nuevo aliento de vida dentro de las aplicaciones existentes.

4.4 Ciclo de vida de las aplicaciones Cliente/Servidor

Ciclo de vida de los sistemas tradicionales.

El ciclo de vida de un sistema tradicional son las fases necesarias para la construcción de un sistema o son las fases para la integración de diversas aplicaciones.

Estas fases comprenden los pasos necesarios desde que se inicia hasta que se termina el proyecto también. Además, el ciclo de vida también comprende la previsión y la consideración de las fases necesarias para que esta aplicación en el futuro sea reemplazada. Cada vez que se emprende un proyecto el ciclo de vida debe ser determinado y previamente analizado.

Las fases del ciclo de vida de un sistema tradicional son:

- Planeación y diseño
 - Análisis Previo.
 - Análisis Funcional.
 - Análisis Orgánico.
- Programación y codificación real del sistema.
- Prueba de los sistemas.
- Instalación definitiva de los sistemas.
- Mantenimiento del sistema.

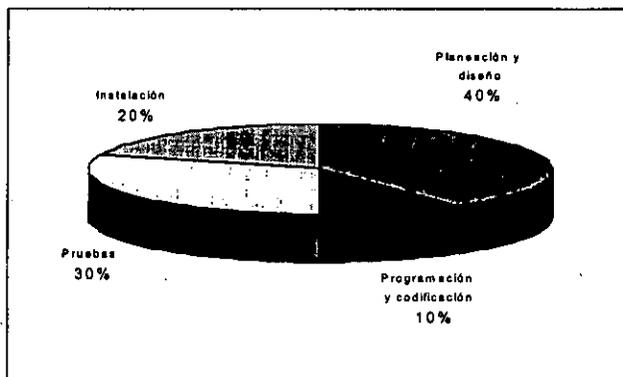


Fig. 4.1. Fases del ciclo de vida de un sistema tradicional

Planeación y diseño

La planeación y diseño de un sistema tradicional es la fase más larga en la construcción de un sistema generalmente se toma aproximadamente un 40% del tiempo del ciclo de desarrollo. Y este comprende las siguientes etapas:

Análisis Previo.- Es un estudio que debe de ser realizado desde un punto de vista económico y técnico, ya que es la iniciación de un nuevo esfuerzo de desarrollo.

Análisis Funcional.- Esta es una actividad que se lleva acabo siempre que exista o no una aplicación utilizándose dentro de la organización. Se estudian las funciones que realizan los usuarios con el sistema o se estudian las funciones manuales o semiautomáticas, después se analizan y definen nuevas funciones, necesarias para mejorar el proceso.

Análisis Orgánico.- Es un estudio que se realiza para determinar la arquitectura de software y hardware que el sistema requerirá así también define la estructura lógica y las especificaciones de las funciones del sistema.

Programación y codificación real del sistema

Se trata de la transcripción a un determinado lenguaje de programación del algoritmo y diagrama de flujo. Esta tarea comprende casi el 10% de la tarea del ciclo de la vida del sistema. En otras palabras esta etapa esta destinada a desarrollar los códigos fuente de los programas, también generar un material de formación y el manual de procedimientos para el usuario.

Esta fase puede descomponerse en diversas tareas cuyo tiempo aproximado de cada tarea será el siguiente:

■ Estudio de especificaciones	10%
■ Organigramas	25%
■ Codificación y preparación de pruebas	25%
■ Pruebas de los procedimientos (por separado)	15%
■ Compilación, pruebas del sistema y documentación	25%

También es conveniente que en esta etapa se designen responsables para cada una de las partes que constituirán el sistema definitivo con el fin de controlar y organizar el mantenimiento de cada subrutina o proceso. Hay que hacer notar que la programación es una tarea altamente técnica que requiere en ocasiones más tiempo del programado por lo que se debe de procurar darle un enfoque adecuado para poder incrementar la productividad. Por tal razón se recomienda que se cree una planificación para la instalación y para la realización de pruebas exhaustivas, inteligentes y cuidadosas. Para el caso de las aplicaciones Cliente/Servidor esta planificación se aplica en todas las fases del ciclo de vida.

Los objetivos finales que se persiguen al codificar cada aplicación del sistema y los cuales son establecidos en esta etapa son los siguientes:

- Utilización mínima de memoria.
- Tratar de que cada aplicación del sistema cuente con lo más reducido de instrucciones que se pueda.
- Que cuente con tiempo corto de ejecución.
- El código y la lógica de la aplicación deben ser claras.
- Las salidas deben también ser claras.

Debemos hacer notar que la claridad de las salidas es importante para los usuarios a su vez la claridad del código y la lógica de la aplicación es importante para los programadores ya que de esta forma las aplicaciones pueden ser realizadas con mayor facilidad y en menor tiempo, además que si por circunstancias de cualquier tipo el programador no puede continuar otro pueda sustituirlo y continuar el trabajo. El código y la lógica entendible se consigue documentando el programa fuente a través de comentarios en las instrucciones o subrutinas claves.

El conjunto de personas que se desempeñará en esta etapa serán:

- Los programadores y
- Los analistas.

Los programadores.- Son un conjunto de personas encargadas de codificar los programas estos pueden ser solo programadores o analistas-programadores. Es necesario que este conjunto de personas tengan suficientes conocimientos del lenguaje de programación que se elegirá para la codificación. En lo que respecta a codificación de aplicaciones Cliente/Servidor, el conocimiento en lenguajes y técnicas Cliente/Servidor será un factor importante en la elección de los programadores.

Los analistas.- Son los encargados de realizar el análisis orgánico y son los concedores de las características estudiadas en el análisis, además de que deben ser capaces de modificar eventualmente las soluciones ya definidas con anterioridad.

Incluir a los analistas en el equipo de programación puede ser conveniente ya que la documentación de una aplicación dentro de un sistema no debe de ser tan extensa, puesto que ellos ya han conocido primero las características de la aplicación que se diseñará por su análisis previo. Sin embargo. Puede ser también contraproducente si se le encomienda a una misma persona tanto el análisis orgánico como la programación ya que en la redacción de los correspondientes expedientes puede llegar a omitir parte de la documentación necesaria, debido a que en el expediente de la programación también se encuentra reflejada información del análisis orgánico. Y esto puede complicar fuertemente si a futuro se desean realizar ampliaciones o modificaciones a las aplicaciones sin la completa documentación que permita el control de los posibles errores de diseño.

Los programadores deben de principiar esta etapa con ya una información completa de la aplicación del sistema que desarrollarán como son los aspectos siguientes:

- Objetivos de los futuros o actuales usuarios del sistema.
- Sistema actual que utiliza la organización (sí es que existe).
- Clasificación del trabajo.
- Características de los ordenadores donde se desempeñará el sistema final.
- La herramienta de desarrollo empleada (si es que se trata de un sistema existente) o la herramienta de desarrollo que se piensa utilizar.
- La clasificación de los datos que manipulará cada aplicación del sistema
- Determinar si existirá relación entre otras aplicaciones del sistema existentes o proyectadas.

Determinar si el volumen de trabajo será grande es una tarea importante que se debe de realizar para definir si se crearán subequipos que se encargarán de cada una de las unidades funcionales y coordinar estos subequipos que generalmente es el "Jefe o líder de proyecto" o un programador muy experimentado en el desarrollo de proyectos de esa naturaleza.

El jefe de proyecto se hará responsable de diseñar, codificar y programar los segmentos más difíciles de cada aplicación del sistema, y también es el responsable de toda la administración del equipo de trabajo, del correcto desarrollo del trabajo y de vigilar el cumplimiento de los trabajos en los tiempos establecidos.

El equipo de programadores debe de mantener registrado los avances de sus tareas. Y por otra parte deben también tener un registro en forma de índice o directorio de las librerías utilizadas, de los diagramas de flujo y de las especificaciones de cada módulo. Ya que los programadores a menudo consultan las librerías y almacenan información para el equipo de trabajo.

Las versiones antiguas de los proyectos deben ser archivadas, de manera que no se pierda información que pueda ser útil, también deben ser archivadas copias de los datos utilizados en las pruebas, describiendo el estado actual de los programas y del proyecto.

El equipo de programación sigue las indicaciones del responsable, trabajan en módulos diferentes y pueden leer los códigos que han escritos otros programadores involucrados en el proyecto.

Es conveniente establecer, utilizar y respetar *normas de programación*, como son las siguientes: *normas de organización y normas del lenguaje*, las primeras las establece la empresa desarrolladora del sistema y son aquellas que tienen que ver con los nombres y numeración de los módulos de las aplicaciones. Y las *normas del lenguaje* que se refieren a las reglas para la realización de los programas para que estos siempre se adecuen a la plataforma definida inicialmente, así cada programador no podrá utilizar un método de construcción propio o diferente, ya que a futuro esto provocará problemas si se desea ampliar el sistema.

Finalmente. La rapidez con que se realicen todas las fases de esta etapa dependerá de la planificación; se pueden utilizar diagramas de Grant o el método de PERT para realizar esta planificación, para que a cada programador se le deba de asignar una unidad de programación o hasta una unidad funcional, esto dependerá del número de personas que constituyen el equipo de programación. La asignación de tareas a los programadores es sobre la base del dominio que tengan sobre la herramienta de desarrollo y el lenguaje de programación.

Pruebas de los sistemas

En esta fase se prueba el sistema en conjunto. Esta tarea comprende la serie de circunstancias reales a las que esta o podrá estar sometido el sistema. Esta fase comprende aproximadamente un 30% del tiempo del ciclo de desarrollo.

Se debe tener una seguridad plena del funcionamiento correcto del sistema. Para realizar las pruebas es conveniente tener una descripción completa y actualizada del programa utilizando para esto los diagramas de flujo.

El método de pruebas consiste en disponer de una serie de elementos que sirven para recorrer todos los módulos del programa, pudiendo existir varias formas de comprobación, empezando por las rutinas, esta comprobación se puede realizar independiente pero teniendo en cuenta la dependencia dentro del sistema y así se prosigue hasta comprobar la totalidad del sistema.

Para cada rutina es necesario comprobar tanto la parte lógica como todas las entradas, inicializaciones y salidas.

Instalación definitiva de los sistemas

Esta fase comprende la manera en que se acoplará el sistema ya desarrollado a la o a las plataformas en donde el sistema se desempeñará, esta fase comprende el resto del tiempo del proyecto, que corresponde a un 20%.

Ciclo de vida de las Aplicaciones Cliente/Servidor

Un principio del desarrollo Cliente/Servidor establece que la calidad debe ser tomada en cuenta desde los inicios del ciclo de vida de la aplicación donde los programadores deben estar muy comprometidos con la calidad durante todas las fases de desarrollo. Esto en teoría es excelente pero en la práctica va en contra del modelo tradicional del desarrollo en cascada de software, que coloca la fase de pruebas del software cerca del final del proceso de desarrollo, como ya ha sido mencionado.

Tradicionalmente dentro de una compañía dedicada a la producción de software, el departamento de desarrollo construye un producto y una vez terminado lo encamina al departamento de pruebas, en estas compañías el departamento de pruebas trabaja al final de la línea de desarrollo, sometiendo a revisión y en su caso a ser re trabajados aquellos productos que no pasen las pruebas a las que son sometidos.

Para adoptar las nuevas técnicas de desarrollo de software las compañías desarrolladoras deben de eliminar en primer lugar toda hostilidad que exista entre la gente de desarrollo y la gente de pruebas, donde el departamento de desarrollo deja el aseguramiento de la calidad final del producto como total responsabilidad del departamento de pruebas. Las nuevas técnicas de desarrollo de software demandan una armonía mucho mayor entre los ingenieros de pruebas y los de desarrollo, esto también provoca que el departamento de pruebas se involucre más temprano en el proceso de desarrollo.

Para armonizar mejor las áreas de pruebas y desarrollo es esencial hacer llegar la información disponible a todo los miembros del equipo de desarrollo y de pruebas. Los sistemas de rastreo de errores permiten a los ingenieros de pruebas comunicarse con los ingenieros de desarrollo, sin importar su ubicación física.

De esta forma en una conclusión personal podemos apreciar que el ciclo de vida de las aplicaciones Cliente/Servidor queda comprendido en tres fases solamente; debido a que cada módulo codificado de la aplicación es probado en línea. Es decir. Paralelamente a la escritura del mismo. De esta forma la gráfica del ciclo de vida de los sistemas tradicionales se modifica ligeramente.

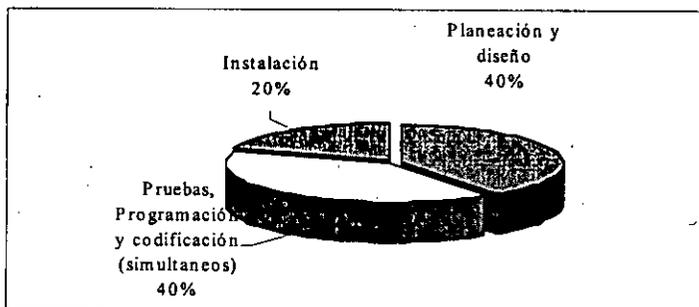


Fig. 4.2. Fases del ciclo de vida de un sistema Cliente/Servidor.

Herramientas de las compañías Alstra software, Rational Rousses software y Pure software permiten a los ingenieros de desarrollo tener ejecución en segundo plano de los códigos de aplicaciones Cliente/Servidor que en ese momento están codificando.

4.4.1 Documentos requeridos para los proyectos de desarrollo Cliente/Servidor

Llevar un registro escrito de los proyectos debe de convertirse en requisito indispensable para nuestro equipo de pruebas, los ingenieros de desarrollo deben estar en coordinación permanente con el equipo de pruebas para que a su vez los diseñadores de la ayuda de la aplicación sepan sobre que ámbitos el usuario requerirá ayuda. Los registros que deben de llevar son los siguientes:

- Requerimientos la aplicación.
 - Definición de hardware o software que requerirá la construcción aplicación.
 - Definición de la interfaz gráfica de usuario.
 - Definición de los ambientes operativos de la aplicación.
 - Alcances y limitaciones que tendrá la aplicación.
 - Definición previa de hardware y software que requerirá la aplicación final.
- Inventario de recursos y requerimientos para el proyecto.
- Bitácora de avance.
- Registro del nombre y funciones de cada procedimiento.
- Registro de dependencias de código de la aplicación.
- Registro de la ubicación y plataforma de cada base de datos.
- Calendario de pruebas en los desarrollos parciales.
- Registro de pruebas parciales.
- Registro de pruebas parciales en las plataformas proyectadas.
- Mantenimiento inicial de la aplicación.
- Puesta en Marcha. (liberación)

En la medida de lo posible es recomendable que en las fases que así lo requieran realizar la diagramación por bloques a detalle de cada subrutina.

Estos expedientes o documentos deben de contener los resultados que se hicieron en todos los estudios y etapas desarrolladas de una aplicación. Las aplicaciones Cliente/Servidor como ya lo hemos mencionado son más complejas; por lo que requieren de más control en la parte de la documentación por lo que no sería mala idea tener una documentación de programación con los siguientes apartados que van desde los aspectos aún más triviales que muchas veces a la larga no parecen serlo tanto hasta los sumamente necesarios.

- Presentación del sistema.
- Personal que participo en las diferentes etapas de programación
- Planificación provisional y real de la programación.
- Archivos utilizados por la aplicación.
- Localización de las bases de datos, tipos de DBMSs y versiones utilizados
- Configuración actual de la red (topología, fallas actuales, nombres de maquinas etc.)
- Documentar las agrupación de rutinas en unidades funcionales.
- Documentar la ubicación de almacenamiento de los subprogramas y secuencias de instrucciones.

- Documentar las variables utilizadas.
- Realizar una lista de los programas compilados.
- Realizar una lista de la ejecución de los programas.
- Incluir el o los diagramas de flujo de la aplicación.
- Presentar todos los puntos en forma clara.
- Usar normas de escritura que permitan evitar confusiones por ejemplo: I, Z, S, O podrían confundirse con 1,2,5 y 0.

Esta documentación para el desarrollo de una aplicación Cliente/Servidor se debe de hacer tanto en la parte cliente como en la parte servidor omitiendo aquellos aspectos que se refieren exclusivamente a su contraparte.

4.4.2 Técnicas de análisis y diseño aplicables al diseño Cliente/Servidor

4.4.2.1 Diagramación

Utilizar un diagrama de flujo representa una ayuda sustancial ya que este mecanismo contiene la representación gráfica de todas las soluciones posibles para obtener una aplicación que solucione la necesidad y requerimientos del usuario. A través de este organigrama se evalúan y se seleccionan las mejores y más viables alternativas desde un punto de vista económico y técnico para la solución del problema. El diagrama se utiliza durante la etapa de pruebas, después sirve para proporcionar las estructuras y contenido de los procesos y por último ayuda al mantenimiento de la aplicación.

Pueden utilizarse dos métodos de diagramación:

- Los diagramas de bloques.
- Los diagramas a detalle.

Los diagramas de bloques.- Sirven para realizar una representación global del problema por medio de los módulos que los componen, sin detallar cada uno de los tratamientos. Estos módulos son rutinas o subrutinas que deben de ser contenidas posteriormente en su organigrama de detalle. El diagrama de bloques siempre constituye una ejecución secuencial de los distintos procesos.

Los diagramas a detalle.- Es aquel diagrama que especifica como su nombre lo dice a detalle o pormenorizados los tratamientos de cada proceso representado en el diagrama de bloques:

Seguir las siguientes normas en la construcción de diagramas es recomendable

- Primeramente construir el diagrama de bloques y a partir de este construir el diagrama de detalle.
- Las rutinas y subrutinas deben de efectuar aquellas operaciones relacionadas con el problema en turno.
- Los problemas deben de dividirse en partes identificables y de ser posible independizarlos.
- Usar el mayor número posible de rutinas y programas estándar.
- Reducir la comunicación operador-computadora, puesto que puede implicar excesivos controles y mensajes de explicación para el usuario.

- Desarrollar al máximo las medidas de control para la corrección y recuperarse de esos errores el sistema.
- Diagramar en base a una solución generalizada que sirva para cualquier lenguaje (puesto que en esta etapa no se tiene con exactitud definido el lenguaje de programación a utilizar).
- Evitar flujos de información complejos y la repetición de decisiones.
- Insertar comentarios en los organigramas de detalle que faciliten su comprensión.
- Revisar los organigramas de detalle cuando se realice una ampliación a la aplicación.

Ya que se finaliza la construcción de la diagramación es conveniente efectuar una comprobación y revisión de la estructura lógica de este antes de pasar a la codificación esto detectará errores inconscientes y también evitará codificaciones innecesarias. Debiendo poner especial atención en los aspectos siguientes:

- Cuando inicia el proceso.
- Tratamiento de los datos iniciales de entrada.
- En las operaciones repetitivas.
- En los puntos de entrada y salida de cada rutina y subrutina.
- El valor de las variables de entrada y salida como parámetros de las rutinas y subrutinas.
- En el final del proceso.

4.4.2.2 Codificación

Esta tarea consiste en transcribir los procesos detallados de los organigramas a instrucciones del lenguaje de programación que se elija. Una técnica recomendada para lograr mayor rapidez en la codificación y mantener una cierta uniformidad en el método de codificación es siguiendo las siguientes reglas:

- Comenzar la codificación hasta que sea comprobado el organigrama de detalle
- Seguir las reglas de identificación de los elementos que componen la aplicación (archivos, campos, registros etc.)
- Escribir comentarios.
- Aplicar las normas definidas por el lenguaje de programación.
- Seguir la estructura del diagrama.
- Optimizar el programa después de la compilación y ejecución ya que es entonces cuando se sabe con más certeza de la memoria utilizada, tanto de disco como de RAM
- Poner especial atención a las instrucciones de identificación del programa, tratamiento de ficheros y declaración de variables internas.

4.4.2.3 Técnicas de Programación

Las técnicas de programación que se utilizan tradicionalmente, son también aplicables al desarrollo de aplicaciones Cliente/Servidor. Existen dos técnicas generalizadas en su uso dentro de las técnicas tradicionales.

- La Programación Modular.

■ La Programación Estructurada.

Programación Modular.- Esta técnica marca que se deben de dividir el problema en varias partes funcionales, con objeto de analizarlas, programarlas y depurarlas independientemente. El módulo puede ser una rutina o subrutina. Al utilizar esta técnica se consiguen los siguientes objetivos:

- Se disminuye la complejidad.
- Disminuye el costo de programación
- Aumenta la claridad del código de la aplicación.
- Aumenta también la confiabilidad del código.
- Se tiene mucho mejor control sobre el proyecto.
- Se facilita la posterior ampliación y la corrección de la aplicación.

Para determinar el orden de las sentencias en el interior de cada módulo pueden utilizarse dos tipos de diseño: descendentes ("Top-Down") y ascendentes ("Bottom-Up").

Con el diseño Top-Down es posible comprobar el funcionamiento de los módulos mediante módulos ya comprobados.

Con el diseño Bottom-Up se consigue llegar desde el diseño detallado al problema total y la comprobación se realiza independientemente de cada módulo.

Al utilizar programación modular cada programador fundamentalmente se dedica a confeccionar el módulo que se le fue asignado.

Hay que mencionar que utilizando técnicas de diseño modular el ciclo de vidas de las aplicaciones en las fases de planeación, diseño y programación cambia puesto que se requiere de mucho tiempo en la etapa de diseño; pero tenemos una ventaja que es la de llegar a tener una documentación muy exacta para cada módulo otra ventaja es que se facilita la codificación y el mantenimiento de las aplicaciones.

El ciclo de vida de las aplicaciones utilizando esta técnica de diseño modular es el siguiente:

- Análisis de las necesidades. (Es la creación de la documentación completa)
- Diseño (creación de las tareas de cada módulo y estableciendo el enlace entre módulos).
- Programación (Codificación y depuración de cada módulo)
- Integración (Conexión de todos los módulos) y prueba final de la aplicación.
- Mantenimiento (Corrección de errores, cambio y adición de nuevas funciones, entran aquí los aspectos cambiantes en la red)

Hay que recordar que estas pruebas van implícitas desde la fase de diseño hasta la integración comprenden todos los ámbitos de la aplicación así como todos los aspectos de los entornos de funcionalidad.

Los frecuente problemas que podemos encontrar al utilizar esta técnica son los siguientes:

- Algunos problemas al querer realizar modificaciones a la aplicación (debido fundamentalmente a la magnitud de la misma y al medio ambiente de red cambiante).

- Problemas al formar al personal de desarrollo para que conozca a fondo el problema a resolver.
- Problemas en la integración de la aplicación final.
- Problemas para realizar la documentación.

Deberemos realizar la integración de todos los módulos de la aplicación una vez que ha sido codificado y probado cada módulo, esta es una parte crucial que requiere de cierto tiempo ya que módulos que trabajan bien por separado pueden fracasar al trabajar en conjunto. Por tal razón se deben de crear módulos con relaciones muy simples que faciliten al máximo su acoplamiento.

Como *Desventaja* tenemos que al utilizar esta técnica las aplicaciones llegan a crecer demasiado y también llegan a ser demasiado lentos ya que los módulos contienen demasiadas instrucciones y muchas llamadas a otros módulos. Haciendo muy caras las aplicaciones. Pese a esto es una técnica ampliamente utilizada debido a que constantemente están bajado los costos del hardware.

Una red LAN con la incorporación de nuevas aplicaciones Cliente/Servidor o con la incorporación o desincorporación de equipos que hagan uso de recursos de la red se experimentan desbalances los cuales dependen de la dinámica organizacional, este es un factor importante que debe ser contemplado durante el diseño de aplicaciones Cliente/Servidor.

Es recomendable que esta técnica se aplique en aquellas aplicaciones que se piense que sufrirán modificaciones constantes en el futuro.

Programación Estructurada.- Esta técnica utiliza al máximo los recursos del lenguaje que tenga esta característica. Utiliza un conjunto básico de tres instrucciones. En este caso los lenguajes aquí tocados cuentan con esta característica.

- Secuencial.
- Alternativa (Pueden ser Simples o Múltiples, - IF Then Else -).
- Repetitiva.

Utilizando esta técnica se parte de lo general a lo particular descendiendo en la estructura del programa y descendiendo también en su nivel de detalle siguiendo los pasos siguientes:

- Ir de lo general a lo particular.
- Se parte de la definición del problema a un esquema algorítmico.
- Se diseña por niveles, dejando los detalles para después.
- Por último se realiza un trabajo de recomposición del algoritmo completo.

Por supuesto que debe de utilizarse un lenguaje de programación que soporte la programación estructurada, eso realmente en la actualidad no es un problema ya que la mayoría de las herramientas de diseño soportan estructuras de control.

Cuando se utilizan técnicas de programación estructuradas se consiguen los siguientes objetivos:

- Justificación de los costos de hardware y software.
- Se consiguen plenamente cubrir las necesidades previstas por el usuario.
- Aumentan la calidad de las aplicaciones y disminuyen el número de errores.
- Facilitan el mantenimiento de las aplicaciones.

Son susceptibles de utilizarse para el desarrollo Cliente/Servidor así como también las diferentes metodologías de programación estructurada²⁶ como son:

- **Jackson** (Emplea módulos en un orden jerárquico dentro de los diferentes niveles donde se encuentren, utiliza *secuencias*, *repeticiones* y *alternativas*. El diseño de una aplicación con esta metodología se basa en la estructura de los datos).
- **Bertini** (Es descomponer un problema por niveles teniendo cada uno de ellos un inicio, un conjunto de procesos y un final, aquí se representan la estructura de los programas y no las operaciones de tratamiento, utiliza *secuencias*, *repeticiones* y *alternativas*. El diseño de una aplicación se basa en una estructura de árbol).
- **Warnier** (Es descomponer un problema por niveles solo que en cada nivel se detallan los tratamientos que permiten la solución del problema, esto se realiza por medio del número de veces que se ejecutan las instrucciones y por la situación de las instrucciones en ese nivel).

4.4.2.4 Análisis mediante componentes

Antecedentes

En un principio los diseñadores de aplicaciones se dieron cuenta que ciertas funciones y procedimientos básicos y algunos no tanto, eran continuamente utilizados y constantemente copiados de un programa fuente a otro, así a través del tiempo; estas funciones y procedimientos llegaron a formar conjuntos llamados "librerías", donde, se reunían los procedimientos y funciones más utilizados. La gente de desarrollo mantiene en un directorio especial dentro de su propia computadora de trabajo estas librerías y llevaban una lista alterna (los más ordenados) de la finalidad de cada función de cada librería, esta práctica era de lo más común dentro de las empresas que se dedicaban al desarrollo de software. Pronto se dieron cuenta que esta situación era presente en todas las compañías que se dedicaban al desarrollo de software, así que empezaron a intercambiarse estas librerías de manera informal para evitar que se invirtiera tiempo inútilmente en la reinención de la misma función o procedimiento. Posteriormente este mecanismo de distribuir librerías de funciones se formaliza en tal forma que surgieron empresas desarrolladoras de software que se especializaron en generar este tipo de librerías probarlas, mantenerlas y distribuir las. Es en este momento que tienen un costo económico de adquisición.

Así, el mismo nombre también evoluciona y de librería se convierte en "componente"; lo cual implicaba que este código puede ser manipulado añadido y borrado con un solo click del ratón el cual es agrupado en un cuadro de "herramientas de Visual Basic" bajo la denominación de controles (OCX, VBX o ActiveX) y la descripción de sus metas en un entorno de desarrollo, como es el caso de Visual Basic.

Definición de Componente

Un componente transporta una funcionalidad bien definida que puede involucrar dos aspectos datos y algoritmo, Así un componente puede proveer un servicio o hacer uso de servicios

²⁶ Si el lector desea profundizar en estas metodologías de programación puede recurrir a la Encyclopedia of Software Engineering, Autor: John J. Marciniak Volumen, Editorial : Editor-in -chief.

ofrecidos por otros componentes: Los componentes se comunican con otros a través de interfaces (este tipo de interfaces son acoplamientos y nada tiene que ver con las) GUIs, esas interfaces definen y difunden estándares que permiten interactuar (relacionarse) con componentes de cualquier fabricante. La interface entre componentes puede ser vía paso de mensajes, o llamadas a procedimientos sin tener nada que ver con la localización física de los componentes.

Los componentes son procedimientos o funciones definidas por el diseñador de la aplicación, y están segmentados por áreas funcionales que permiten flexibilidad y futuras escalabilidad
Cómo un componente implementa su propia funcionalidad es poco importante
el método está encapsulado dentro del componente esto permite hacer cambios y mejoras sin afectar a otros componentes con los que mantiene una interface.

Con el uso de un middleware transparente para proporcionar comunicaciones entre componentes el esquema de un diseño lógico basado en componentes Cliente/Servidor hacia el diseño físico será relativamente sencillo.

Un Middleware puede variar en su nivel de sofisticación, por ejemplo:

simplemente pasando a través de accesos de bases de datos o convirtiendo entre diferentes DBMS interfaces o proporcionando capacidades de procesos totales de transacciones en línea (OLTP)

Idealmente, cuando se diseña una aplicación la representación física deberá ser solo importante cuando cumpla con el diseño físico. El diseño lógico y el procedimiento de análisis deben ser incambiables, y por lo tanto existen metodologías que pueden ser potencialmente usadas por soluciones en entornos Cliente/Servidor, como hemos mencionado anteriormente.

Durante la duración del proyecto las características del diseño de la aplicación se pueden convertir en una restricción a la solución o como parte del problema que se desea solucionar. Hasta entonces no se tendrá un real impacto sobre el diseño lógico, pero será considerado durante el diseño lógico de la aplicación para hacer la traslación del diseño físico más efectivo. El diseño físico puede ser tratado como una representación de objetos lógicos por ejemplo partes de una presentación (GUI), aplicación, base de datos o localizaciones físicas (por ejemplo sobre un cliente o servidores).

El componente presentación que de aquí en adelante lo llamaremos como **[P]** (es el GUI) es esa parte de la aplicación que tiene que ver con la interface hacia el mundo exterior, tanto para un humano como para una computadora. La presentación puede tener la apariencia de una pantalla de computadora o la impresión de un dispositivo que transmita datos vía telefónica (por ejemplo fax-modem o file transfer).

La Aplicación **[A]** tiene que ver con la funcionalidad definida por el usuario.

El componente base de datos **[D]** contiene los datos del usuario y el sistema manejador de bases de datos. Los componentes residen en plataformas que conjugan el hardware y los sistemas de software requeridos para soportarlos a estos.

Los símbolos entre paréntesis son provenientes de la notación LEGHAWK²⁷ el cual puede ser usada para describir arquitecturas Cliente/Servidor. Ejemplos de LEGHAWK serán usados para describir la variedad de arquitecturas discutidas.

La figura 4.3 muestra las principales posibilidades para arreglar esos componentes sobre plataformas cliente y servidor. Tenemos ignorado los dos casos extremos donde todos los componentes son situados sobre la misma plataforma. Tomando cada uno de los casos en turno.

En algunas publicaciones este modelo es presentado como modelo "Cliente/Servidor de Gartner Group", con ciertas modificaciones²⁸

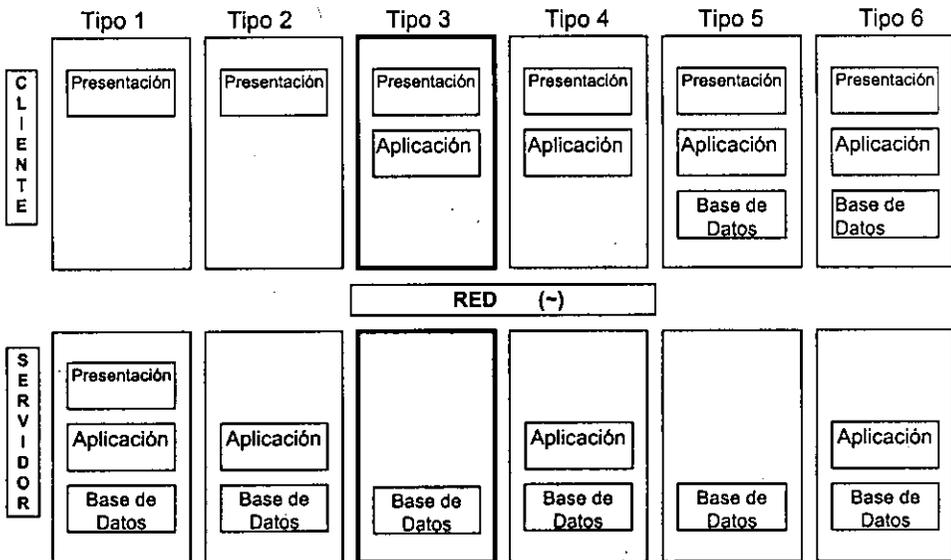


Fig. 4.3. Muestra las combinaciones principales Cliente/Servidor.

Tipo 1. Presentación Distribuida [P] ~[PAD]. En la cual la arquitectura de el componente presentación es dividido entre la plataforma cliente y la plataforma servidor, con componentes aplicación y base de datos localizados completamente en la plataforma servidor. Un ejemplo tal es un sistema donde una presentación GUI mejorada es adicionada a un sistema hereditario existente y lo comunica a una emulación de terminal.

²⁷ Una total descripción de la Notación LEGHAWK es encontrada en el apéndice 2 de Client/Server a handbook of Modern Computer Design, de Clive Evans Ed. Prentice Hall.

²⁸ Es el caso de la revista "DataMation Strategic Solutions for enterprise computing professionals" en su artículo: Client/Server Unchained Finally, and hardware Independence Junio 1995. Y en la conferencia impartida por el Ing. José Ramon Riego, EDS de México. Del ciclo de conferencias DBC/S 96 Unix.

Ventajas:

- Mejora la interfaz gráfica de usuario (GUI).
- No hay cambio a la aplicación Host.
- Es una interfaz amigable.

Tipo 2. Presentación Separada (Presentación remota) [P]~[AD]. Aquí el componente Presentación esta localizado completamente sobre la plataforma cliente, con la aplicación y la base de datos completamente en el servidor. Esto nos permite hacer buen uso relativamente de workstations de bajo costo las cuales soportan presentaciones GUI, software Windows como presentación controlada desde una aplicación remota.

Ventajas:

- Mejora la interfaz gráfica de usuario.
- La presentación se muda del servidor al cliente.
- Es una interfaz más amigable.

Tipo 3. Base de Datos Separada (Manejo remoto de datos) [PA]~[D]. Aquí ambas la presentación y la aplicación están localizadas completamente sobre la plataforma cliente con solo la base de datos localizado sobre la plataforma servidor. Esta es probablemente la más común de las arquitecturas Cliente/Servidor actualmente en por lo que también es soportada por muchos vendedores de herramientas manipuladoras de bases de datos (DBMS) y herramientas para usuarios.

Ventajas:

- Mejora la interfaz gráfica de usuario(GUI).
- Manejo de datos centralizado.
- Integración de datos en varias plataformas.

Tipo 4. Aplicación Distribuida (Funcionalidad Distribuida) [PA]~[AD]. En este caso la presentación y parte de la aplicación están localizadas en el cliente, con la base de datos y el resto de la aplicación localizada en el servidor. La aplicación funcionalmente puede ser dividida verticalmente (por ejemplo cada función esta completamente en cliente o servidor) u horizontalmente (donde alguna parte de la función está sobre el cliente y comunicaciones con la otra parte en el servidor). Generalmente, la división será una combinación de dos, con al menos una forma de activación siendo comunicación desde la aplicación cliente a la aplicación servidor. Esto conlleva a permitir funciones localizadas sobre las más apropiadas plataformas y una interface (transacción de negocios) será definida en un apropiado nivel, quizás ocultando detalles de la implementación del servidor desde el cliente.

Ventajas:

- Mejora el desempeño al minimizar la carga de trabajo de la red.
- Centraliza los datos de la aplicación en el servidor.
- Es una interfaz gráfica de usuario más amigable.
- Mayor flexibilidad en el diseño.

Tipo 5. Base de datos Distribuida [PAD]~[D]. Este coloca ambas la presentación y la aplicación completamente sobre la plataforma cliente, con la base de datos dividida entre las dos plataformas. Los datos pueden ser divididos verticalmente (por ejemplo diferentes tablas sobre cada plataforma) u horizontalmente (diferentes renglones de las tablas sobre diferentes plataformas) o replicado (con algunos datos apareciendo sobre diferentes plataformas). Hay

muchas que bien son una combinación de las tres. Esto permitirá que los datos sean colocados donde son más convenientes, particularmente para mejorar el rendimiento mediante la colocación de los datos más cerca de donde serán usados.

Ventajas:

- Los datos están más cercanos al cliente.
- Permite tener redundancia y trabajar de forma local en lo que se vuelve a tener la comunicación entre el servidor y el cliente.
- Es una interfaz gráfica de usuario más amigable.

Tipo 6. Aplicación Distribuida y Base de datos (Proceso Distribuido) [PAD]~[AD]. Aquí ambos la aplicación y la base de datos son divididos en las dos plataformas, con la presentación completamente en el cliente. Este puede ser una combinación de aplicación distribuida y arquitectura de bases de datos distribuida y es un caso más general. Idealmente, los componentes pueden ser movidos desde una plataforma a otra transparentemente. Esto es lo último en arquitectura Cliente/Servidor con el máximo de flexibilidad para el cambio.

Ventajas

- Arquitectura flexible
- Cualquier componente puede cambiarse a donde tenga mayor sentido de negocio.
- Es una combinación de base de datos distribuida con funcionalidad distribuida

Hay otras posibles combinaciones con dos plataformas, arquitecturas plus que involucran más de dos plataformas. Sin embargo, tales arquitecturas son cualquiera de las dos improbables en la práctica (por ejemplo [PD] ~[A]) o pueden ser tratadas como combinaciones de las de arriba (por ejemplo [P] ~[A] ~[D] aparece como [P] ~[AD] desde el punto de vista de la plataforma [P]. o

[PA] ~[D] desde la plataforma [D]. O ambas desde el punto de vista de la plataforma [A]). De las seis arquitecturas, ejemplos prácticos pueden ser divididos en dos principales tipos.

Enseguida explicaremos las combinaciones Cliente/Servidor más encontradas en las organizaciones.

La combinación Cliente/Servidor más común (Tipo 3 de la figura anterior) Executive Information Systems (EIS) [PA] ~[D]

Este tipo de combinación es conocido como Executive Information Systems o sistema ejecutivo de información por sus siglas es (EIS). Este requiere análisis y presentación de datos en bruto que han sido consolidados de alguna forma. A menudo los datos están modelados y los análisis de requerimientos son altamente cambiables tanto como las nuevas técnicas de manejadores y métodos de monitoreo que involucran y se mueven fuera de la moda. Desde el punto de vista de los requerimientos de arquitectura.

Características

- La velocidad de acceso a datos no es crítica. En pocos minutos se descarga una serie de datos para manipular, analizar y modelar. La pregunta apropiada es ¿si será más rápida que

OLTP?, ya que no será dentro de 1 a 3 segundos el tiempo de respuesta requeridos para los sistemas OLTP.

- Los sistemas EIS generalmente no actualizan los datos. El uso de SQL en bruto sin integridad y chequeo de validación es por lo tanto aceptable, aunque almacena procedimientos y reglas de negocio que puede enriquecer los servicios proporcionados por la base de datos.
- Las consultas SQL a menudo son conocidas en lo avanzado, así los procesos del cliente deben ser capaces de construir SQL dinámico. Trabajando en conjunción con los usuarios.
- Los procesos del cliente están sobre la workstation o PC, así que ellas pueden tomar ventaja del sofisticado formateo y técnicas de sofisticación disponibles. En EIS, la presentación y formateo de los datos están íntimamente ligados a la misma aplicación y reside en el mismo procesador. El proceso cliente EIS tendrá sus propios métodos interconstruidos de presentación o tomará ventaja de productos provenientes de terceros. Hay una gran cantidad de productos EIS comercialmente disponibles para workstations que solo requieren SQL para acceder a los datos. Alguna forma de programación de alto nivel (tales como macros) diseñadores de interfaces para construir aplicaciones cliente.

La combinación Cliente/Servidor Proceso de Transacciones en Línea (OLTP) [P] ~[AD] (Tipo 2) y [PA] ~[AD] (Tipo 4)

Los sistemas OLTP tienen las siguientes características:

- Muchos usuarios usan la misma aplicación.
- Las transacciones son generalmente cortas.
- Los accesos a los datos son predecibles.
- Rápida respuesta al usuario el tiempo es esencial (algunas veces los usuarios han sido acostumbrados al teléfono).
- La integridad de los datos es muy importante.

Los sistemas OLTP son generalmente implementados mediante aplicaciones que se encuentran corriendo sobre los servidores en una misma plataforma como lo es la base de datos. Debido a que el servidor está esperando por la petición del cliente, no hay sobre demanda en el cargado del programa, la inicialización y el SQL "parsing" requieren algunas interfaces de bases de datos. Los servidores son los responsables de la integridad de los datos y pueden ejecutar complejos métodos de validación de acuerdo a las reglas de la organización para después hacer la escritura física en la base de datos. El ciclo de la transacción "commit" (escritura física de la información contenida en RAM) asegura que la base de datos es consistente en todo o nada de la transacción es ejecutada. Indexando para un sencillo acceso a datos debe ser igualados contra la necesidad de actualizar esos índices cuando los datos son añadidos o cuando son cambiados. Así la ejecución SQL es predecible, pero posiblemente convirtiéndose en una base de datos cara de mantener.

Muchas aplicaciones Cliente/Servidor requieren conceptos de ambos modelos el EIS y OLPT. Las aplicaciones Cliente/Servidor que están fuera de este esquema, por lo general tienden a concentrarse sobre los requerimientos de EIS²⁹, como este es comparativamente simple de implementar, además. Hay una clara demanda para esta combinación desde la comunidad de los negocios. Las aplicaciones con OLPT son mucho más difíciles de implementar, pero desde que las aplicaciones servidor están en su lugar, ellas pueden ofrecer un alto nivel de abstracción para las tareas de consulta de los clientes EIS, debido a que ellos pueden proporcionar información acerca de los objetos de negocios, son más que simplemente tablas de bases de datos. Por ejemplo, un costoso análisis de un requerimiento puede entrelazar la división que hay entre EIS y OLPT en inesperadas maneras. Una aplicación para un grupo de almacenes puede tener toda la información en su base de datos que todos los almacenes descargan al final de cada día, y generar nuevos índices durante la noche para complejas consultas el próximo día. Esto puede ser principalmente una aplicación EIS, con poca necesidad de inmediata actualización.

4.4.2.5 Herramientas requeridas

Por último la elección de las herramientas requeridas para el desarrollo de aplicaciones Cliente/Servidor mediante la técnica de componentes dependerá del grado de complejidad que demande el proyecto, que es determinado mediante el estudio de los documentos del proyecto.

Durante el proceso de desarrollo es importante poder conocer y tomar en cuenta las dependencias de código, "Discover" una herramienta que fabrica Software Emancipation Technology, esta destinada para proporcionar a los desarrolladores una vista de las dependencias de código y permite compartir la información entre los miembros de un equipo distribuido. Esta herramienta es útil tanto en el desarrollo de código nuevo, como en la corrección de código heredado³⁰.

Al elegir la herramienta adecuada para el desarrollo de aplicaciones Cliente/Servidor; asegúrese que soportará todos los manejadores de bases de datos usados en su organización, por ejemplo herramientas de desarrollo tales como **Delphi Cliente/Servidor suite 2.0 de Borland y Visual basic 4.0 y 5.0 Enterprise edition** pueden tener acceso hasta ocho manejadores de bases de datos. Es importante contar con herramientas que brinden un soporte sólido en SQL, productos que ofrecen esto son **Delphi, Visual Basic y SQL Windows 5.0**.

²⁹ Según apreciación de Setrag Khoshafian autor de "A guide to developing Cliente/Servidor SQL applications".

³⁰ El código heredado son programas fuentes que debe un diseñador de aplicaciones incorporar o retrabajar dentro de una nueva aplicación, por lo general quien retoma este código no es el autor.

4.5 Diferencias entre aplicaciones controladas por Procedimientos y Controladas por Eventos

Este tema presenta el modelo de programación controlada por eventos y lo contrasta con un modelo de programación por procedimientos.

Aplicaciones controladas por Procedimientos

En las aplicaciones tradicionales o basadas en procedimientos, es la aplicación quien controla qué fragmentos del código se ejecutan y la secuencia en que se ejecutan. La ejecución de la aplicación comienza con la primera línea de código y sigue una ruta predefinida por EL DESARROLLADOR de la aplicación, llamando a procedimientos a medida que son necesarios. Las aplicaciones que hoy en día están basadas en procedimientos exclusivamente dejan de aprovechar en gran medida funcionalidades con las cuales no se contaba en el pasado (apartados 3.6, 3.7 y 3.8 de este documento).

Aplicaciones Controladas por Eventos

En una aplicación controlada por eventos, la ejecución no sigue una ruta predeterminada, sino que ejecuta distintas secciones de código como respuesta a eventos. Los eventos pueden desencadenarse por acciones del usuario, mensajes del sistema o de otras aplicaciones, o incluso desde la propia aplicación. La secuencia de estos eventos determina la secuencia en la que se ejecuta el código. Por tanto, la ruta a través del código de la aplicación difiere cada vez que se ejecuta el programa.

Una parte esencial de la programación controlada por eventos consiste en escribir código que responda a los posibles eventos que pueden ocurrir en una aplicación. **Visual Basic** facilita la implementación de un modelo de programación controlada por eventos.

Este modelo es ampliamente recomendado y generalmente utilizado para el desarrollo de aplicaciones en ambientes Windows independientemente de que sean o no aplicaciones Cliente/Servidor.



Fig. 4.4: La ilustración muestra algunas acciones que generan eventos los cuales ejecutan una secuencia de código especial para este evento. Estos eventos pueden producirse en cualquier orden.

4.6 Ambientes de desarrollo de Aplicaciones Cliente/Servidor

Uno de los retos más importantes a superar en el desarrollo de aplicaciones Cliente/Servidor es la distribución, debido a que las aplicaciones desarrolladas deberán de funcionar sin importar la distribución geográfica. En el desarrollo de aplicaciones Cliente/Servidor intervienen dos factores la ingeniería y la administración de programas, ingeniería debido a que la meta es conseguir software de calidad mediante técnicas y herramientas de desarrollo de software y administración de programas ya que hay que hacer uso de técnicas y herramientas de administración de información.

Debido a que en un ambiente Cliente/Servidor la información es traída y llevada de un lugar a otro, uno de los aspectos más costosos en el diseño de aplicaciones Cliente/Servidor son los mecanismos de comunicación subyacentes ya que resulta tardado y difícil escribir directamente secuencias de código para un protocolo de comunicación como por ejemplo TCP/IP o SPX (Sequenced Paket Exchange) y las soluciones de comunicación ligeramente superiores como los Sockets y los TLP (Transport Level Interface) ofrecen una mejoría muy marginal, de esta forma la única solución idónea esta en un mecanismo de comunicación fácil de implementar que es lo que se conoce como middleware³¹.

Los ambientes de desarrollo para aplicaciones de bases de datos Cliente/Servidor, se utilizan para construir clientes sofisticados que permiten tener acceso a servicios de bases de datos, tales son los casos de **SQLBase de Grupta** y **SQL de Microsoft**. Estas herramientas incrustan capacidades de middleware dentro de funciones de nivel más alto. Los mejores ambientes de desarrollo incluyen Delphi Cliente/Servidor Suite 2.0 de Delphi; Visual Basic Enterprise Edition 4.0 o 5.0 de Microsoft, SQL windows 5.0 de Grupta y Power.

³¹ Ver apartado 3.12 de este documento.

4.7. Desarrollo de Aplicaciones en ambiente MS Windows

Introducción

El entorno MS Windows es un entorno gráfico. En su evolución se han añadido mejoras que lo dejan muy lejos respecto de su primera versión; el MS Windows de 1980 que fue un gran fracaso, las versiones posteriores de Windows aprendieron grandemente de los fracasos anteriores cuyos frutos se ven en las últimas versiones de MS Windows a partir de la versión 3.0, su interface gráfica, sencilla, fácil de manipular e intuitiva rápidamente lo convirtió en la favorita del usuario. Posteriormente esta definió la tendencia hacia el futuro, con su creciente presencia y su rápida aceptación por los usuarios dentro de las organizaciones definieron también la tendencia en el desarrollo de software para este ambiente.

Este fue un punto importante para la consolidación como un ambiente de trabajo en toda la extensión de la palabra, pero faltaba aún algo, debido a que este ambiente de trabajo solo se permitía a nivel monousuario, y para poder trabajar en un ambiente de red era necesario hacer uso de software de red generalmente de otros fabricantes, no obstante que ya existía MS Lan Manager pero este además de ser un producto de 16 bits no contaba con la suficiente popularidad y con el suficiente trabajo de mercadotecnia atrás como lo es para MS Windows, así que un paso lógico posterior en la evolución de MS Windows fue enfocado hacia el desarrollo de características que permitieran la independencia de MS Windows del software para red de otros fabricantes.

Es así como nace en 1992 la primera versión de "MS Windows Para Trabajo en Grupo", el cual ya contaba con características básicas para desempeñarse en red llegaba con un reducido conjunto de protocolos, generalmente utilizados en redes LAN para compartir y utilizar recursos de estas, es en este momento que el entorno MS Windows inicia una nueva fase adentrándose en la arquitectura **Cliente/Servidor** de manera formal. Sin embargo, Windows, solo era el ambiente gráfico y el sistema operativo continuaba siendo MS-DOS generalmente la versión 6.x o 5.x con un conjunto de utilerías como el "Smartdrive", "himem.sys", entre otras que permitían utilizar la memoria más allá de los 640KB y poder utilizar y generar memoria virtual para los mayores requerimientos de recursos de hardware que demandaba "MS Windows Para Grupos de Trabajo".

Microsoft decide empezar a desplazar y a dejar en desuso los sistemas operativos de 16 bits posteriormente y por ende también el desarrollo y el uso de aplicaciones de 16 bits, así es que reúne en un solo producto un entorno gráfico un sistema operativo y accesorios y utilerías para poder trabajar compartiendo y accediendo a recursos en una red LAN. Es así como nace el sistema Operativo MS Windows 95 en el verano de 1995 con una interfaz gráfica revolucionada, el cual tuvo una muy buena acogida por parte de los usuarios. Windows 98 continuo esta tendencia posteriormente, pero con una muy marginal aceptación a pesar de la gran promoción.

Por otro lado hasta ese momento Windows NT continuaba su evolución solamente expectante de los que era bien aceptado en el mundo de Windows 9X para también incorporarlo a Windows NT y de esta forma continuarán teniendo mucha similitud ambos sistemas operativos.

La estrategia de Microsoft era clara, encaminar a todos los usuarios de su más popular sistema operativo MS-DOS a su sistema operativos de Nueva Tecnología, el Windows NT, teniendo como intermediarios a toda la gama de Windows 9x.

El hardware común existente en el mercado cuando nació Windows NT no era tolerable por Windows NT ya que los requerimientos de hardware para Windows NT requerían fuertes inversiones que solo las organizaciones grandes o medianas podían realizar para tener Windows NT y Windows NT workstation como estación de trabajo. Aunado a esto las aplicaciones en su mayoría eran de 16 bits y su GUI de texto.

Poco a poco se evoluciona en el hardware y al mismo tiempo se abarata constantemente llegando a tener una aceleración impresionante a partir de mediados de 1998 y continuando hasta la fecha. Por lo que en este momento la PC más barata del mercado es capaz de aceptar Windows NT como sistema operativo con toda facilidad. Microsoft terminará de hacer convergir a todos los usuarios de MS-DOS, MS Windows, MS Windows For WorkGroups, y los Windows 9x hacia Windows NT con la liberación próxima de la serie Windows 2000 que no es más que el Windows NT con otra cara y un tanto más "evolucionado".

4.7.1. Cómo se desarrolla una Aplicación Cliente/Servidor en un ambiente MS Windows

Mencionaremos en primera instancia cómo se desarrolla una aplicación en MS Windows en un ambiente monousuario y posteriormente **destacaremos las características** que debe de reunir cualquier aplicación a fin de poder desempeñarse en una arquitectura Cliente/Servidor. Además de hacer especial énfasis en que las pruebas del sistema se harán constantemente a diferencia del modelo tradicional de desarrollo que marca las pruebas casi hasta el final del desarrollo de la aplicación.

Desarrollar una aplicación para una arquitectura Cliente/Servidor, implica una técnica de desarrollo diferente a la aplicada para desarrollar una aplicación para un sistema monousuario; debido a que para el desarrollo de una aplicación de este tipo los datos no se encuentran en un solo punto sino que debemos de trabajar con una colección de datos dispersos o distribuidos que van desde una hasta varias bases de datos en diferentes PCs. Y cuyo medio de enlace y recolección de datos es la red. Esto es lo que hace en cierta medida complicado el desarrollo de aplicaciones Cliente/Servidor.

Desarrollar una aplicación en MS Windows requiere de los elementos siguientes:

1. Plataforma MS Windows instalada.
2. Elegir un lenguaje que permita el desarrollo de aplicaciones en un entorno MS Windows.
3. Aplicar técnicas de desarrollo de software para un entorno MS Windows.
 - Generar GUI (Interface Gráfica del Usuario).
 - Establecer propiedades a la GUI.
 - Generar Código.
4. Ubicar los posibles datos distribuidos (para Cliente/Servidor).

4.7.1.1 Creación de la Interfaz de usuario

La interfaz de usuario es quizás la parte más importante de un aplicación, por su carácter de ser la constantemente visible. Desde la perspectiva del usuario la GUI es la aplicación y seguramente a la mayoría no les interesa lo que ocurra atrás de la aplicación para arrojar los datos que le solicitaron, que código se ejecuta y que algoritmo se siguió para su diseño, independientemente del tiempo y del esfuerzo que haya empleado en la escritura y en la optimización del código. Pero olvidándonos un poco de las injusticias destacaremos que la facilidad de uso de una aplicación depende de su GUI.

Lo que se debe de considerar en la GUI cuando se diseña una aplicación:

Cuando se diseña una aplicación tiene que tomar muchas decisiones relacionadas con la GUI.

- Primeramente es necesario definir con exactitud el propósito de la aplicación.
- El estilo de documento que se debe de utilizar, el estilo único o el estilo múltiple.
- Los formularios diferentes que se utilizarán.
- Decidir que comandos se incluirán en los menús.
- Decidir que barras de herramientas se utilizarán para duplicar funciones de los menús.
- ¿Cuál será la forma de los cuadros de dialogo que interactuarán con el usuario?
- ¿Qué nivel de asistencia necesita proporcionar?
- Definir si se tratara de una aplicación de uso constante o de uso esporádico o casual.
- Así como los requisitos de diseño que debe de cumplir dicha aplicación según sea su propósito.

El diseño de una aplicación de uso diario difiere del diseño de una aplicación de uso ocasionalmente por breves periodos de tiempo, así también el diseño de una aplicación destinada a presentar información será diferente del diseño de una aplicación destinada a extraer u obtener información.

Generalmente el definir estas características de la aplicación debe de partir de una audiencia y entrevista con las personas que utilizarán la aplicación, donde plantearán las necesidades y perspectivas que debe de cubrir la aplicación una vez terminada, ahí se determinará por parte del diseñador si la aplicación estará destinada a usuarios expertos, medios o principiantes, definitivamente una aplicación destinada a principiantes demandará de técnicas de diseño y de código más elemental que el de aquella aplicación destinada a usuarios experimentados debido a que esta última demandará de diseño y de código más avanzado.

Por otra parte existen expectativas que pueden influir en el comportamiento de la aplicación por ejemplo si se piensa en la probabilidad de distribuir la aplicación fuera del país de origen, se debe de considerar el factor del idioma y la cultura para ser más agradable y entendible la interfaz gráfica de usuario.

Generalmente se considera la mejor técnica de diseño de la interfaz gráfica de usuario, a aquella basada en un **proceso de desarrollo interactivo**.

4.7.1.2 Estilos de Interfaz de usuario

Existen dos tipos tradicionales de interfaces de usuario en Windows.

1. La interfaz de documento único (SDI) por sus siglas en inglés Single Document Interface y
2. La interfaz de documento múltiple (MDI) por sus siglas en inglés Múltiple Document Interface.

Existe otro tipo de interfaz aún no muy común el cual es denominado estilo de interfaz de **explorador**.

SDI, se refiere a aplicaciones que solo pueden cargar un documento a la vez, por ejemplo la aplicación Block de Notas de Windows ó el wordpad.

MDI se refiere a aplicaciones que son capaces de cargar varios documentos cada uno en su propia ventana generalmente son aplicaciones más robustas como por ejemplo, Excel, Word, PowerPoint.

Para determinar cual es el mejor estilo de interfaz para una aplicación tiene que determinarse el propósito preciso de la aplicación, por ejemplo una aplicación para procesar reclamaciones en ventas se sugiere bien un estilo **MDI**, ya que puede darse el caso de un empleado que trabaje con más de una reclamación a la vez.

A diferencia de una aplicación para un calendario o la carátula de un reloj para la cual se sugiere un estilo de aplicación **SDI**.

En el estilo de Interfaz **explorador**, se presenta la información en una sola ventana dividida en dos secciones izquierda y derecha o arriba y abajo, que consiste en una visita jerárquica a la izquierda/arriba y una presentación pormenorizada en forma de árbol a la derecha/abajo. Ejemplo de esta interfaz gráfica de usuario es el Explorador de Windows, el Internet Mail o el Outlook.

4.7.1.3 Desarrollo Interactivo

El proceso de desarrollo interactivo de aplicaciones se puede dividir en 3 fases, codificación, compilación y comprobación del código, Visual Basic es diferente a otros lenguajes de programación por que utiliza una aproximación interactiva para el desarrollo de aplicaciones, combinando las tres fases conforme se escribe el código.

Visual Basic por su naturaleza permite que se desarrolle una aplicación de manera interactiva. Es decir. Permite que la aplicación sea ejecutada continuamente de tal forma de, ir comprobando los efectos del código según se escriba.

En otros lenguajes donde se utiliza propiamente un compilador como el generador del código objeto de la aplicación; cuando se comete un error durante la codificación del código, el compilador interceptará este error en el momento de la compilación y al encontrar y corregir el error debe nuevamente iniciar el proceso de compilación, repitiendo el proceso hasta que el código quede libre de errores y así pueda generarse el código objeto. En cambio un interprete

como es el caso de Visual Basic el código se interpreta a medida que se escribe, interceptando y resaltando la mayoría de los errores de sintaxis, aunado a esto Visual Basic compila parcialmente el código según se escribe; para detectar errores de ejecución, de esta forma cuando se termina de codificar una aplicación su código objeto y su archivo ejecutable es generado más rápidamente, si el compilador encuentra un error, quedará resaltado en el código, se puede corregir y seguir compilando sin tener que comenzar de nuevo.

Debido a que en la fase de programación del ciclo de vida de las aplicaciones Cliente/Servidor se requiere probar paralelamente a la escritura del código resulta Visual Basic a partir de su versión 5 una herramienta muy útil para el desarrollo de este tipo de aplicaciones, sobre todo en un rango de pequeñas a medianas.

4.8 Desarrollo de una Aplicación Cliente/Servidor en VB

¿Por qué en Visual Basic?

Visual Basic se ha convertido en el lenguaje de programación más popular de Microsoft para el desarrollo en plataforma Windows. Quizás la más importante vía para que Visual Basic se haya popularizado es por su característica para proporcionar acceso directo a las API de Windows.

Aquellos quienes están acostumbrados a usar Visual Basic de 16 bits pueden ser confundidos por la documentación de Visual Basic que describe la convención para procesar el Visual Basic de 32 bits como algo relativamente simple como lo es cargar un proyecto existente, obteniendo controles OLE de 32 bits para reemplazar a lo VBXs usados por el programa y convirtiendo alguna llamada API a Win32.

El desafío de reemplazar VBXs con controles OLE, especialmente si sus VBXs son personalizados o no provienen de uno de los principales vendedores de componentes de software, es un interesante reto que no es abordado en esta ocasión. Es el "simple" proceso de portar llamadas APIs provenientes de Windows de 16 bits a Windows de 32 bits, desafortunadamente no necesariamente es tan simple como algunas de las documentaciones de Visual Basic pueden sugerir.

Las aplicaciones desarrolladas en MS Visual Basic versión 5 son 20% más rápidas que las que son desarrolladas en la versión 4. Según publica Microsoft en la parte de ayuda de la versión 5.

Utilice los dos básicos pasos siguientes para crear una aplicación de Visual Basic:

1. Crear la interfaz gráfica de usuario de la aplicación.

2. Escribir código que responda a las acciones realizadas en la interfaz gráfica de usuario.

Siempre que se genere una aplicación en Visual Basic, empiece por crear la interfaz gráfica de usuario.

Mediante el cuadro de herramientas, dibuje o coloque controles en un formulario para crear los elementos visuales de la aplicación. El cuadro de herramientas sólo está disponible en tiempo de diseño.

Usar el cuadro de herramientas

El cuadro de herramientas contiene controles incorporados de Visual Basic y los controles ActiveX o los objetos insertables que haya agregado al proyecto. Si está cerrado, puede abrirlo si hace click en el comando **Cuadro de herramientas** del menú **Ver**.

Para colocar un control en un formulario, puede hacer doble click en el control o hacer un único click y dibujar el control en el formulario.

Agregar controles al cuadro de herramientas

Puede ampliar el cuadro de herramientas si agrega controles ActiveX. La Edición profesional de Visual Basic proporciona varios controles ActiveX adicionales. También puede adquirir controles ActiveX de otros proveedores.

Para agregar un control ActiveX al cuadro de herramientas

1. En el menú **Proyecto**, haga click en **Componentes**.

Visual Basic mostrará el cuadro de diálogo **Componentes**.

2. En la ficha **Controles**, haga click en el control que desee incluir y haga click en **Aceptar**. Visual Basic agregará el control al cuadro de herramientas.

Asignar nombre a los controles

Cuando asigne nombre a los controles, siga las convenciones estándar para asignar nombre a la interfaz gráfica de usuario y los controles. Esto hará que el código sea más fácil de leer y depurar.

Para obtener información acerca de las convenciones de nombres de los controles, busque **Convenciones de código** en los Libros en pantalla de Visual Basic y vea **Convenciones de nomenclatura de los controles**.

Cuando genere la interfaz gráfica de usuario para una aplicación de Visual Basic, deberá establecer las propiedades para los objetos que cree.

Establecer las propiedades en tiempo de diseño

Algunas propiedades se pueden establecer en tiempo de diseño. Para ello, puede utilizar la ventana Propiedades o las páginas de propiedades.

Para tener acceso a la ventana Propiedades, haga clic con el botón secundario del *mouse* (ratón) en un objeto y después haga click en **Propiedades**. La ventana Propiedades únicamente muestra las propiedades que son comunes a todos los controles seleccionados. Cualquier cambio que realice a una propiedad se aplicará a todos los controles.

También puede establecer propiedades para varios controles al mismo tiempo. Para seleccionar varios controles, haga click y arrastre el *mouse*.

Establecer las propiedades en tiempo de ejecución

En tiempo de ejecución es posible escribir código para establecer o recuperar propiedades. En el siguiente ejemplo de código se establece la fuente de tipo negra para el cuadro de texto denominado

`txtData:`

```
txtData.Font.Bold = True 'Pone texto en negra.
```

El siguiente ejemplo de código establece la propiedad **Text** del cuadro de texto denominado `txtData`:

```
txtData.Text = "Hola a todos" 'Establece el valor de texto.
```

Si omite el nombre de la propiedad establecerá la propiedad predeterminada del control. La propiedad predeterminada de un cuadro de texto es **Text**. La propiedad predeterminada de una etiqueta es **Caption**.

El siguiente ejemplo de código establece las propiedades predeterminadas **Text** y **Caption** para un control de cuadro de texto y de etiqueta, respectivamente:

```
txtData = "Establece la propiedad Text del cuadro de texto"  
lblData = "Establece la propiedad Caption de la etiqueta"
```

Obtener propiedades en tiempo de ejecución

Puede utilizar el siguiente código para obtener propiedades en tiempo de ejecución:

```
Dim CuerdaName As String  
CuerdaName = txtName.Text
```

Herramientas de Desarrollo de Aplicaciones Cliente/Servidor

5.1 Introducción

Las herramientas de desarrollo de aplicaciones también han tenido su evolución importante, sus técnicas análisis y desarrollo han cambiado significativamente así como las herramientas que se utilizan para tal efecto. Hoy las empresas dedicadas exclusivamente a la producción de software poseen más que simplemente herramientas de desarrollo ya que elaboran aplicaciones Cliente/Servidor en auténticos ambientes de desarrollo donde un grupo coordinado de ingenieros y diseñadores gráficos actúan intercambiando y mejorando código independientemente de la ubicación geográfica en la que se encuentren.

La tendencia es que todas las herramientas modernas enfatizan el diseño sobre la programación, y cada vez es menor la cantidad de código por escribir de forma manual y mayor el que generan las herramientas de desarrollo, cambiando de manera radical la proporción de tiempo empleado en análisis, diseño, construcción y mantenimiento de aplicaciones.

En este capítulo se tocarán aspectos relacionados con un conjunto de herramientas destinadas al desarrollo de aplicaciones Cliente/Servidor, Estas herramientas son de las más conocidas dentro del ámbito de desarrollo, en este sentido se hablará de las principales características de cada una de estas, y cuáles son sus ventajas más significativas.

Espero que aquellos que conocen bien de algunas de estas herramientas sepan disculpar las raquíticas exposiciones y las imprecisiones que aquí se exponen.

5.2 Antecedentes:

Para el desarrollo de aplicaciones Cliente/Servidor se puede emplear un rango de herramientas que van desde lenguajes de 3ª generación hasta, ambientes de desarrollo, siendo exclusiva de ciertas herramientas la capacidad de desarrollo en los entornos Cliente/Servidor.

En la década de los 90s y 80s han estado presentes dos tipos fundamentales de lenguajes en el mercado

- Los lenguajes de 3ª Generación. (3GL)
- Los lenguajes de 4ª Generación. (4GL)

Los lenguajes de 3ª Generación más populares son COBOL, C, C++, Pascal y RPG común. En la categoría de los de 4ª Generación son los considerados totalmente estructurados existe mucha diversidad debido a que casi todos han sido lenguajes propietarios asociados a un manejador de bases de datos (RDBMS), por ejemplo informix 4GL. Y también han existido lenguajes abiertos 4GL. Es decir, capaces de trabajar sobre diferentes fuentes de datos, como son archivos planos, archivos indizados y diversos manejadores de datos.

5.3 Categoría de Herramientas de Desarrollo:

Existen herramientas de los siguientes tipos desde el punto de vista del ámbito en que se desenvuelven:

- Herramientas para el desarrollo de aplicaciones personales.
- Herramientas para el desarrollo de aplicaciones de grupo.
- Herramientas para el desarrollo de aplicaciones departamentales.
- Herramientas para el desarrollo de aplicaciones corporativas.

Herramientas para el desarrollo de aplicaciones personales.

Son aquellas que se ejecutan solo en la máquina del usuario y no tienen acceso a ningún servidor. Para este tipo de aplicaciones se emplean herramientas como C, C++, MS

Visual Basic, MS Visual C y lenguajes de manejadores de bases de datos como lo son dbase, Paradox, MS FoxPro, MS Access y CA Clipper sin librerías netlib.

Herramientas para el desarrollo de aplicaciones para grupos de trabajo

Son aquellas herramientas que desarrollan aplicaciones en las cuales se comparten las bases de datos entre un grupo reducido de usuarios. Para el desarrollo de este tipo de aplicaciones es

posible utilizar las herramientas que se utilizan para el desarrollo de aplicaciones personales pero con la condición de que se diseñen pensando en su desempeño en una red, lo cual implicará definitivamente el uso de bloqueo de registros ó de los archivos de las bases de datos, Oracle PowerObjects.

Herramientas para el desarrollo de aplicaciones departamentales

Son aplicaciones que manejan generalmente bases de datos que van del rango de medianas a grandes, que proporcionan servicio a un número considerable de usuarios. Para el desarrollo de estas aplicaciones se emplean herramientas que permiten el trabajo de grupos de desarrollo. Por ejemplo. Informix New Era, Gupta SQL Windows, PowerBuilder y Borland Delphi Client/Server, estos productos acceden a través de enlaces de alto desempeño a bases de datos como Sybase Oracle o MS SQL Server.

Herramientas para el desarrollo de aplicaciones corporativas

Son aplicaciones que emplean bases de datos distribuidas que dan servicio a un gran número de usuarios, por lo que se requiere que la aplicación desarrolle un muy alto desempeño. Las herramientas utilizadas para el desarrollo de estas aplicaciones deben por lo tanto permitir el acceso a bases de datos distribuidas e incluso archivos heterogéneos, algunas de estas herramientas son propietarias, pero poco a poco este panorama se está abriendo, entre las herramientas que podemos encontrar son : UnifAce Six, Oracle, CDE, Symantec, TEAM-Enterprise developer, PowerSoft PowerBuilder Enterprise y Borland Delphi Client/Server.

5.4 Herramientas "Front-End" (Cliente).

Existen un conjunto de aplicaciones, herramientas y programas clasificados técnicamente como "front end" que hacen que los entornos Cliente/Servidor tenga más funcionalidad, tales como las siguientes:

- **Herramientas de consulta.** Estas herramientas incorporan consultas predefinidas fijas y desplegados de informes para ayudar a los usuarios a tener acceso a los datos de manera más rápida, se utilizan en clientes donde las consultas son rutinarias y fijas.
- **Aplicaciones de usuario.** Programas integrados de algunos fabricantes como Microsoft Excel, Lotus 123 entre otros proporcionan acceso "front end" a bases de datos. Manejadores de bases de datos medianos como Microsoft Access, incluyen su propio SQL para proporcionar una interfaz con sistemas de administración de bases de datos de varios fabricantes.
- **Herramientas de desarrollo de programas.** Muchos Entornos Cliente/Servidor necesitan aplicaciones "front end" especiales y personalizadas para sus tareas de

obtención de datos, tales como Microsoft Visual Basic, Borland C Builder, SQL Windows ayudan a los programadores y administradores de sistemas de información a desarrollar "front-ends" para tener acceso a los datos de una base de datos.

5.5 Eligiendo el Correcto Front-End

Después de que los componentes han sido identificados, es tiempo de pensar en una herramienta de desarrollo. Mientras no halla un claro líder en el mercado de herramientas de desarrollo que destacadamente sus capacidades la hagan más atractiva que otras. Las herramientas de desarrollo son diferenciadas por la interfaz gráfica de usuario (GUI) o por el más transparente proceso (Back-End). Las herramientas también se eligen sobre la base de los siguientes tres criterios, según sea el enfoque o el destino.

Hay tres tipos principales de herramientas Front-End

- Herramientas basadas en SQL.
- Edificadores de interfaces de usuario.
- Herramientas integradas de desarrollo.

5.6 Ambientes de Desarrollo

Los ambientes de desarrollo son un caso aparte ya que no sólo incluyen lenguajes de cuarta generación sino que también incorporan herramientas que facilitan el trabajo en grupo, utilizando editores, depuradores además de que pueden compartir código, a través de las redes independientemente de la ubicación geográfica. Ejemplo de ambientes de desarrollo son Sybase, APT Workbench, Cognos, PowerHouse y UnifAce

5.7 Herramientas CASE

Computer-Aided Software Engineering, "Ingeniería de software asistida por computadora" (CASE) es tanto un estudio de campo como un sitio en el mercado. Como un campo de estudio es una colección de investigaciones desarrolladas a lo largo de muchos años en materia de software como lo es desarrollo, manejo métodos, ambientes y herramientas. En general CASE es la industrialización de las técnicas de ingeniería de software y de la tecnología de cómputo para proporcionar y automatizar la práctica de los desarrollos de sistemas de información. Esto es una integración tecnológica que incorpora desarrollo técnicas de software. CASE atrae tecnologías de diseño asistido por computadora,

ingeniería de software, sistemas operativos, bases de datos, redes y otras disciplinas. Así que CASE no es un nuevo campo, aunque si es un campo de aplicación, integración y síntesis.

Como un sitio en el mercado, CASE representa un variado conjunto de productos y servicios de desarrollo de software e integración de sistemas. El término CASE cubre una variedad de necesidades de herramientas de software y de programas para incrementar la productividad. En este contexto CASE es generalmente considerado como cualquier uso de tecnología de cómputo para auxiliar al desarrollo de software.

Hay muchos términos equivalentes activos usados en la literatura técnica para identificar a los ambientes CASE tales como SDEs Software Development Environments, ambientes de desarrollo de software, SEEs, Software Engineering Environments, y PSEs, Programming Support Environment.

Hay una marcada división entre los ambientes que soportan las actividades del ciclo de vida del front-end y las que soportan al back-end. Ya que las actividades del front-end son las de modelado y análisis del problema, planeación de sistemas, análisis de sistemas, especificación de requerimientos y un diseño lógico. Las actividades de back-end incluyen el diseño físico, construcción (programación y codificación), depuración, pruebas, integración de componentes de software, puesta en marcha y mantenimiento.

Existen herramientas CASE de todo tipo que van desde aquellas que solo su función es diseño de aplicaciones hasta aquellas que generan automáticamente el código en lenguaje C o en otros lenguajes de 4ª generación por medio de compuertas gateways, ejemplo de una herramienta es el SilverRun que genera aplicaciones en UnifAce.

5.8: Gupta SQL Windows.

Las herramientas integradas de desarrollo tales como Gupta's SQL Windows y el PowerSoft de Power Builder son usados para construir aplicaciones que se conectan a un selecto juego de bases de datos SQL.

Los constructores de Interfaces de usuario tales como Case PMW, se centran exclusivamente en la construcción de las funciones de presentación de una aplicación. Los constructores de interfaces de usuario trabajan mejor cuando usted necesita de proporcionar conexiones a múltiples ambientes no estándares.

Las herramientas integradas de desarrollo proporcionan funciones de presentación y múltiples protocolos para la integración dentro diversos ambientes. Las herramientas de desarrollo proporcionan simple conexión a múltiples ambientes a costa de la flexibilidad.

5.9: C++

El C++ es un ambiente que representa dificultades de uso y requiere mucho más tiempo y disciplina para el desarrollo de aplicaciones Cliente/Servidor. Muchos no aceptan la escasez de soporte para RAD³². Mucha gente concibe a C++ como una herramienta para construir productos destinados a un mercado masivo, como pueden ser sistemas relacionados con funciones tales como dispositivos controladores, DLLs y componentes tales (como activeX) para especializadas herramientas Cliente/Servidor.

5.10: Lotus Notes

Ejemplo de Interfaz gráfica de usuario

Hoy en día Cliente/Servidor e interfaces gráficas de usuario amigables GUI van siempre juntas, y relacionado con la plataforma de desarrollo de la estación cliente.

Uno de los aspectos que el diseñador de aplicaciones Cliente/Servidor debe conseguir es el de poder separar el ambiente de la plataforma de desarrollo del ambiente del GUI, además de que la aplicación que desarrolla sea fácil de aprender y comprender por los usuarios finales, adicionalmente esta debe intentar estar bien incorporada al resto del ambiente de oficina (correo electrónico, procesador de palabras, accesorios calculadoras, gráficos rústicos etc.), aquí Lotus Notes puede desempeñar un papel interesante en el desarrollo de aplicaciones. Los desarrolladores de aplicaciones Cliente/Servidor conocen que es relativamente fácil y rápido construir una aplicación utilizando las formas de Lotus Notes versión 4 donde viene incorporado un enlace "link" con MQ³³, el desarrollador de aplicaciones solo debe diseñar la forma y después simplemente la envía a un servidor como una pieza de correo a un username denominado MQ_LINK y la forma pasa a ser un mensaje en la cola de entrada de MQ, el desarrollador, en el lado de la estación cliente ni siquiera tiene que utilizar el API de MQ, ya que Lotus Notes lo trae incorporado. Dado que Lotus Notes es el estándar de groupware y correo en muchas empresas. Por otra parte dado que Lotus Notes corre tanto en OS/2 como en Windows y Unix una aplicación desarrollada en Lotus Notes se mantiene dentro del concepto de portabilidad.

El lenguaje de desarrollo en Lotus Notes es Lotus Scripts que no es más que un superset de instrucciones del lenguaje Visual Basic.

³² RAD (Desarrollo Rápido de Aplicaciones)

³³ Mencionado en el apartado 3.2.2

5.11 Microsoft SQL Server

Actualmente existe un estándar de SQL en el mercado llamado ANSI 92, sin embargo, la versión de Microsoft de SQL Server tiene muchas extensiones que permiten realizar labores más precisas y sofisticadas las aplicaciones pueden ser desarrolladas y optimizadas para trabajar en conjunto con MS SQL Server.

5.12 VisualAge

VisualAge es un ambiente integrado de desarrollo de aplicaciones Cliente/Servidor, emplea tecnologías de programación visual, de construcción por partes y programación orientada a objetos. Se logra una gran productividad en el desarrollo de sus aplicaciones.

Características de VisualAge:

- Crear aplicaciones de negocios avanzadas para ambiente Cliente/Servidor en muy poco tiempo.
- Diseño intuitivo y consistente de interfaces gráficas de usuario(GUI).
- Aprovecha la inversión hecha en las aplicaciones existentes o tradicionales (código heredado).

Incrementa la productividad

VisualAge a parte de ser un diseñador de interfaces gráficas de usuario (GUIs) es una herramienta para crear aplicaciones complejas como transacciones en línea (OLTP) y de soporte a la toma de decisiones. Con VisualAge se pueden realizar una aplicación en un promedio de dos semanas, lo que normalmente tomarían meses usando lenguaje C ó COBOL.

Programar con VisualAge es una tarea fácil, e intuitiva según destaca IBM proporciona todo lo necesario para construir, la porción del cliente en las aplicaciones Cliente/Servidor en un ambiente totalmente orientado a objetos.

- Soporta múltiples plataformas.
- Soporta el desarrollo interactivo.
- Contiene biblioteca de partes prefabricadas amplia y extendible.
- Cuenta con un generador de partes prefabricadas.
- Contiene un prototipo de aplicaciones.

Programación Visual

Muchas aplicaciones no requerirán de codificar, simplemente se tiene que seleccionar las partes de la biblioteca y hacer las conexiones apropiadas en la pantalla. Cuando nuevas partes

necesitan ser creadas podrán ser hechas con el lenguaje Smalltalk que está dentro ó podrá usarse lenguajes como C ó COBOL y VisualAge automáticamente hará que estas partes se comporten como objetos.

La biblioteca de partes prefabricadas incluyen:

- Consultas a bases de datos.
- Interfaces gráficas.
- Múltiples protocolos de comunicación.
- Multimedia.

Requerimientos:

Para desarrolladores:

- Mínimo procesador 80486, 25 MHZ con 16 MB RAM.

Para los usuarios:

- Mínimo procesador 80386, 66 MHZ donde la memoria dependerá de los programas de aplicación generalmente 8 MB de RAM.

Sistema operativo:

- Windows NT, DOS, OS/2

5.13 Delphi: Client/Server Suite

Delphi es una poderosa herramienta de dos vías esto quiere decir que puede ser usada gráficamente o mediante su código, teniendo así el beneficio de ambas partes, la facilidad de uso de las herramientas gráficas y el manejo fino que se consigue a través de la manipulación directa del código. Delphi permite el acceso directo a bases de datos locales como Dbase, Paradox y otras por medio de ODBC.

Delphi es superior en muchos aspectos respecto a otras herramientas de desarrollo de aplicaciones Cliente/Servidor. El control de navegación de Delphi permite desplazar, añadir, actualizar, borrar o cancelar los cambios a los registros con un solo click del ratón y sin necesidad de escribir código. Un editor de campo permite una rápida colocación de campos sobre las formas y especificación fácil de campos calculados y de búsqueda. Una vez que el desarrollador ha especificado una tabla o un objeto de consulta, puede seleccionar varios campos y llevarlos hasta la forma. Delphi viene equipado con un explorador de bases de datos una herramienta para la administración de bases de datos. Delphi cuenta con una sola cuadrícula de control de datos que es básicamente una cuadrícula de miniformas, donde se pueden colocar campos múltiples, etiquetas y otros objetos, en la celda superior izquierda de la cuadrícula en el orden que se desee, así cada celda de la cuadrícula despliega un registro diferente. Cuenta con facilidades para poder vincular campos en el control maestro de esta forma no será necesario realizar un procedimiento cada vez que se pase a un nuevo registro maestro y poder realizar consultas a detalle. También cuenta con un par de herramientas para realizar reportes y controles para poder integrar los reportes a las aplicaciones, es difícil el

diseño de los reportes por esta razón es conveniente contar con una herramienta extra llamada ReportSmith que es ejecutado desde las aplicaciones para la realización de los reportes.

Pero la desventaja es que Delphi no puede ejecutar un procedimiento de evento cada vez que el usuario se mueve hacia un registro nuevo, no obstante puede un programa buscar cualquier valor siempre y cuando se trate de un campo indexado.

5.14 PowerBuilder Enterprise

La versión 4.0 de PowerBuilder Enterprise hace uso de dos interfaces principales para diseñar formas y Data Windows (formas para visualizar información de las bases de datos), que permiten realizar consultas colocando los campos de resultado en una ventana. Las interfaces gráficas de usuario de PowerBuilder son de las más intuitivas y amigables para el usuario. PowerBuilder tiene la capacidad de poder implementar con SQL estándar, donde la aplicación ejecuta una declaración SQL y puede moverse en la base de datos según lo determine el manejador de bases de datos, esto es denominado implementación de **cursores**. Utilizando estos cursores el desarrollador de la aplicación debe de mover los datos que llegaron del cursor a los campos de la forma, para que una vez que se ejecute la aplicación esta despliegue en su GUI la información. Viene incluido un sofisticado diseñador de reportes, incluyendo la capacidad de agrupar reportes de muchos niveles que los usuarios pueden ejecutar directamente. Sin embargo. La vista preliminar, visualización e impresión de las paginas seleccionadas requiere de un programa por separado, InfoMaker que viene incluido dentro de PowerBuilder pero que es difícil de ejecutar desde aplicaciones hechas por este.

5.15 Java

Java es un lenguaje de propósito general. Sin embargo. Por su característica de multiplataforma es preferido para el desarrollo de aplicaciones para el Web debido a que es un lenguaje de programación fuertemente orientado a objetos ya que trabaja en ambientes Cliente/Servidor, fue creado por Sun Microsystems es multiplataforma por lo tanto independiente del hardware. El compilador Java genera un código que es traducido a través del interprete de instrucciones de máquina virtual (virtual machine) las cuales pueden ser ejecutadas en cualquier arquitectura de hardware (Risc, Intel) a la cual haya sido portado el interprete.

En la computadora cliente Java crea una máquina virtual que esta compuesta por los siguientes elementos:

- Un conjunto de instrucciones (bytescodes).
- Una pila.
- Un área de métodos (Zona de memoria donde almacena los métodos).
- Un conjunto de registros (virtuales).
- Un recolector de basura (garbage collected heap).

Las aplicaciones de Java conocidas como applets, que residen en el servidor son incorporadas al WWW y ejecutadas por cualquier navegador que tenga soporte para Java, (Netscape a partir de la versión 2.0 e Internet Explorer a partir de la versión 3.0) así los applets son descargados automáticamente desde el servidor Web hacia la máquina cliente y ejecutados en esta.

5.16 Visual Basic

Microsoft Visual Basic en su Edición empresarial versión 5.0 proporciona el entorno de programación y las herramientas integradas que precisa el programador corporativo para generar, probar y desplegar aplicaciones distribuidas Cliente/Servidor a gran escala.

Con estas herramientas y técnicas, ahora es posible generar aplicaciones fácilmente a partir de "bloques constitutivos" que se pueden compartir, reutilizar y reubicar en la red.

Visual Basic concuerda con el modelo de tres niveles para el diseño de aplicaciones distribuidas, junto con las herramientas y técnicas necesarias para sacar el máximo partido del diseño basado en componentes.

Cuenta con 20 mejoras respecto de la versión 4. Algunas de estas están destinadas a la edición empresarial que es la que permite desarrollar más sofisticadas aplicaciones Cliente/Servidor.

Las mejores características para Cliente/Servidor de la Edición empresarial

Migración a COM en un entorno distribuido (Edición empresarial) Compatibilidad total con componentes distribuidos mediante COM: todas las herramientas de Automatización remota admiten ahora la comunicación remota mediante COM en Windows 95 y NT 4.0.

Mejoras en RDO y RDC (Edición empresarial) La biblioteca de Objetos de datos remotos (RDO) y la característica de Control de datos remotos (RDC) versión 2.0 mejoran la compatibilidad con cursores locales, las actualizaciones optimistas por lotes, las conexiones independientes y los objetos de consulta.

Diseñador de conexiones (Edición empresarial) Diseñador ActiveX integrado que le permite crear objetos de conexión independientes con métodos consistentes en consultas definidas por el usuario o en procedimientos almacenados.

Depuración de SQL (Edición empresarial) Herramienta de depuración interactiva que le permite depurar código Transact-SQL desde el entorno de diseño de VB.

Repository (Edición empresarial) Objetos **Repository** integrados que le permiten guardar, organizar, buscar y recuperar datos acerca de su proyecto de Visual Basic. Un modelo de objeto extensible y una API le permiten un acceso total por programa al objeto **Repository**. Es posible crear nuevos modelos de objetos que le ayudan a administrar sus herramientas de software.

5.17 Rendimiento General de Algunas Herramientas.

Rendimiento General

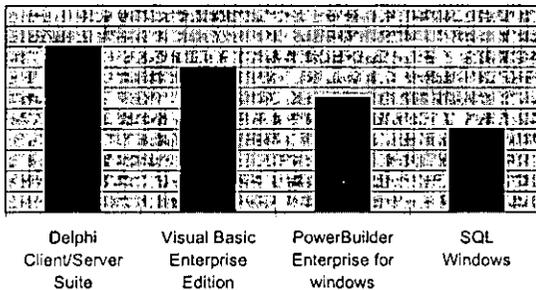


Fig. 5.1. Muestra la razón de rendimiento de diferentes herramientas de desarrollo Cliente/Servidor³⁴.

Nota:

El tema de herramienta de desarrollo es amplio y puede en algunos casos volverse complejo. Debido a que depende de otros factores además de las facilidades y ventajas que brinde intervienen factores de tipo social, cultural económico, experiencia, adecuación y gusto por tal o cual herramienta que determinará su elección. Por lo que se puede finalizar este apartado diciendo que no hay una herramienta de desarrollo "mejor a otras" actualmente. Pero debe

³⁴ Clasificación según la revista Byte.

siempre considerarse que la base de la elección de una herramienta de desarrollo no debe ser una decisión que deba tomarse a la ligera ya que deben de ser tomados en cuenta varios puntos, significativos de los más importantes mencionaremos los siguientes:

Elegir la herramienta de desarrollo sobre la base de:

1. El ámbito que deberán cubrir las aplicaciones.
2. El tipo de base de datos a emplear.
3. El DBMS a emplear.
4. El tipo de cliente y servidor en el que la aplicación se desempeñará.
5. El precio de la herramienta y el soporte de esta.

Requisitos de rendimiento y seguridad de las Aplicaciones Cliente/Servidor

6.1 Introducción

Unos de los aspectos que deben ser considerados desde que se principia el desarrollo de las aplicaciones Cliente/Servidor son los aspectos de rendimiento y seguridad. Una aplicación Cliente/Servidor formal debe de reunir ciertas normas mínimas de seguridad y debe tener un buen rendimiento ya que de otra de forma será una aplicación que estará condenada al fracaso. Por lo que si es una aplicación que realice procesos innecesarios la aplicación Cliente/Servidor será una aplicación en respuesta muy lenta. Las organizaciones siempre demandan de tener aplicaciones rápidas y eficientes.

Este capítulo habla sobre los requisitos que debe poder reunir una aplicación Cliente/Servidor para que use de manera más eficiente los recursos de los entornos Cliente/Servidor, a la vez que cumpla con un estándar de seguridad también contempla como saber el impacto que registran las aplicaciones ya implantadas. Esto se analiza desde diferentes vertientes, entre los que se consideran los aspectos de hardware de redes y de equipo de cómputo, así como la disposición tanto física como lógica de la red Windows NT.

6.2 Requisitos de hardware para una Aplicación Cliente/Servidor.

Una vez obtenido el análisis final de fiabilidad de instalación por parte de la dirección de la organización esta decide si se instalará la aplicación. No todas las aplicaciones requerirán de la selección y adquisición de nuevo equipo de cómputo para su implantación salvo en aquellos casos que se equipare la implantación de la aplicación a la evolución tecnológica natural de la propia organización. Por lo general la instalación de la aplicación Cliente/Servidor se realiza sobre los equipos de cómputo ya existentes. Pero consideremos el caso extremo, donde si se requiera realizar esta adquisición de equipo nuevo pensado que se tratará de la formación de un nuevo departamento o una organización naciente.

Cito los siguientes requisitos, por supuesto tomando en cuenta que las PCs que se adquieran deben de satisfacer los requisitos de la aplicación. Los requisitos a contemplar serán:

- Requisitos de demanda de hardware por parte de la aplicación Cliente/Servidor.
- Requisitos funcionales.
- Requisitos de adaptación.

Requisitos de demanda de hardware por parte de la aplicación Cliente/Servidor .-

Contemplan los siguientes puntos:

- Cantidad de unidades de información que puede procesar en un periodo determinado.
- Rapidez de respuesta a las solicitudes de los usuarios.
- Capacidad para realizar cálculos complejos.

Requisitos funcionales.- Simplemente se refiere a que el equipo sirva para las tareas que la aplicación demandará.

Requisitos de adaptación.- Se refiere a que el equipo debe de rebasar por cierto margen a las necesidades de la aplicación Cliente/Servidor, para que pueda con facilidad cubrir los requerimientos futuros de la aplicación cuando esta requiera ser ampliada o el sistema operativo tenga que ser evolucionado junto con otras herramientas a fin de prolongar la vida útil de la aplicación Cliente/Servidor, como son los puntos siguientes:

- Suficiente espacio en disco duro.
- Superior espacio de memoria RAM al recomendado por al aplicación Cliente/Servidor.
- Monitor VGA de buen tamaño.
- Capacidad para realizar actualización en su micropocesorador y otros componentes que interfieran en su rendimiento.

Los diversos requisitos ayudan también a determinar la clase de equipo auxiliar que se adquirirá como por ejemplo impresoras, y dispositivos de comunicación.

El jefe de proyecto puede solicitar cotizaciones de equipo de tres maneras:

Requisición de equipo específico.- Es en donde se solicita a los vendedores de equipo de cómputo propuestas de precio de equipo con características determinadas.

Requisición basada en objetivos de rendimiento.- El líder de proyecto convierte los requisitos de la aplicación en objetivos de rendimiento y solicita que los vendedores propongan la clase de equipo que debe de conseguir mejorar esos objetivos.

Requisición a un determinado vendedor.- Donde se selecciona a un vendedor de equipo de cómputo muy competente para que proponga un número reducido de alternativas basadas en la tecnología que comercialice y que satisfaga las necesidades de la aplicación.

Una vez que se han solicitado ofertas a los vendedores y que se ha decidido seleccionar algunas de ellas que se acercan o cubre todos los requisitos. Entonces se debe de seguir con la prueba de esta selección. Para probar el equipo que cada vendedor propone existen dos métodos: *comparación de productividad y simulación.*

El método de comparación de productividad. Se basa en ejecutar determinados problemas de prueba en los diferentes equipos propuestos. Se prueba con algunas aplicaciones la carga de trabajo esperada como son el sistema operativo, las herramientas de desarrollo, los paquetes de aplicación y de servicio auxiliares. Aquí el principal factor determinante será el tiempo que consume el equipo de cómputo en la ejecución de las aplicaciones Cliente/Servidor para resolver el problema.

El método de simulación. Este método hace uso de programas de tipo matemático que aceptan cierto tipo de entradas, por ejemplo la estructura de los archivos, tamaño, frecuencia de acceso a los ficheros etc., a estos modelos son sometidos los equipos de cómputo con el fin de conocer su tiempo de respuesta. Estos módulos deben ser acordes con los métodos estándar de Cliente/Servidor.

Estos dos métodos requieren de una adecuada planeación tanto de espacio como de tiempo para que estas pruebas no se realicen apresurada y torpemente arrojando resultados irreales, Si se trata de proyectos grandes los vendedores propondrán la incursión de uno o algunos de sus expertos en el equipo de pruebas de hardware-aplicación que conozcan suficientemente el hardware que estos proponen

Tal vez los resultados de los diferentes vendedores sean muy parecidas en costo por los que recomiendo valorar los siguientes criterios que son determinantes. Junto con otros criterios que son validos para seleccionar son:

- **Modularidad (escalabilidad).** Es la posibilidad que tiene el equipo de aceptar componentes de hardware adicionales, a fin que el equipo se amplíe para cubrir las necesidades de la aplicación.
- **Compatibilidad.** Debe de existir concordancia entre los equipos que soportarán la aplicación. No puede elegirse un equipo el cual no pueda aceptar o proporcionar datos de la aplicación Cliente/Servidor transparentemente.

- **Confiabilidad.** Esta cualidad depende del fabricante, saber si este cuenta con métodos de control de calidad ampliamente probados y reconocidos.
- **Mantenimiento, asesoría y servicio por parte del fabricante y del vendedor.** Este quizá es uno de los valores agregados más preciados ya que la asesoría del vendedor tiene gran importancia cuando se trata de elegir el equipo, esto también debe incluir la asesoría y el entrenamiento para la instalación y la ayuda que podrá proporcionar el vendedor tiempo después de la instalación.

6.3: Requisitos técnicos que debe reunir una Aplicación Cliente/Servidor.

- Integración
- Flexibilidad
- Facilidad de uso
- Flujo de trabajo
- Que cuente con asistentes (Wizards)

Integración

La aplicación Cliente/Servidor debe de fomentar la armonía en las tareas dentro de la organización, es decir. Evitar la duplicidad de datos y esfuerzos, eliminando así inconsistencias de información y expandiendo mejor la contribución de datos respetando el flujo personal de cada usuario de tal manera que se cumplan con los objetivos de la organización a través de la aplicación. También la integración se refiere a tratar de incluir a la aplicación Cliente/Servidor dentro del resto del ambiente de oficina (correo electrónico, procesadores de texto, calculadoras, etc.)

Flexibilidad

Es la capacidad que tenga la aplicación Cliente/Servidor para adaptarse a los cambiantes y a veces impredecibles necesidades de la organización, mediante herramientas integradas o utilerías que conviertan a la aplicación en una solución dinámica, para generar nuevos reportes, formas de captura, bases de datos, e incluso fabricar interfaces gráficas para el usuario sin necesidad de escribir una sola línea de código.

Facilidad de uso

Es hacer lo posible por que la GUI de la nueva aplicación Cliente/Servidor se parezca a los sistemas profesionales para Windows, que el usuario ya conoce y a los cuales ya esta acostumbrado, logrando una forma de operar natural, intuitiva y relajante para el usuario, contribuyendo a reducir de forma drástica la curva de aprendizaje e incrementando la velocidad de trabajo.

Flujo de trabajo

La aplicación le permite controlar de manera óptima e intuitiva el flujo de trabajo asignable a cada usuario. Hay desarrolladores que implementan un navegador en su aplicación Cliente/Servidor para ir hacia atrás o hacia adelante dentro de los flujos de información.

Asistentes (Wizards)

Existen ciertas capturas o adecuaciones de la aplicación que requieren de un asistente que le facilite y le sugiera la captura por su naturaleza compleja de las mismas, por ejemplo si la aplicación se está utilizando en el departamento de cobro y un usuario requiere dar de alta un cobro pendiente. Al utilizar el asistente le mostrará las cuentas pendientes y le ayudará a distribuir el monto de la mejor manera.

6.4 Cinco niveles de clasificación de las organizaciones tomando en cuenta el grado en el cual han asimilado los procesos de análisis y Diseño de Aplicaciones.

Estas cinco clasificaciones son hechas desde los puestos gerenciales o de dirección donde muchas veces no están totalmente conscientes de conservar y contar con todos los documentos de desarrollo de la aplicación Cliente/Servidor que se implante en su organización, en ocasiones es debido a la poca importancia que le dan a la aplicación ya que piensan que será muy pasajero su uso o por que no cuentan con un capacitado departamento de sistemas que cargue con esta responsabilidad

Esta es una falta de previsión sumamente grave ya que si la gente que desarrollo la aplicación pertenece a la organización y en un momento dado se va de la organización si es necesario en el futuro modificar la aplicación será un trabajo complicado a tal grado que generalmente se opta por rehacer la aplicación.

Las clasificaciones son las siguientes:

Primer nivel. Las compañías que solo necesitan la aplicación pero que no les interesa en lo absoluto la metodología de diseño que se haya implementado para el desarrollo de la aplicación Cliente/Servidor.

Segundo Nivel. En esta categoría están empresas que se preocupan y se interesan un poco más en la metodología aplicada a la aplicación que están adquiriendo o desarrollando internamente y recaban algunos informes y al final terminan con una documentación de desarrollo incompleta que de poco les ayuda si se necesita modificar o ampliar la aplicación.

Tercer Nivel. Son organizaciones que se encuentran en un nivel intermedio, si les interesa que exista una metodología pero sin dedicarse mucho a ésta.

Cuarto Nivel. Son organizaciones que se encuentran en un nivel bastante aceptable. Se lleva a acabo una metodología y esta se documenta, sin desarrollar su máxima capacidad.

Quinto Nivel. Son organizaciones que dan mucha importancia a la metodología y a partir de ella desarrollan en un 100% sus sistemas.

En México la mayoría de las empresas se encuentran clasificadas en el primer nivel.

6.5 Confiabilidad de Windows NT

Windows NT Workstation 4.0 es más confiable que Windows 95 y Windows 98

El sistema operativo de Windows NT Workstation Ver. 4.0 es destinado fundamentalmente a estaciones de trabajo se protege a sí mismo, así como a los controladores y dispositivos, contra caídas del sistema. Esto significa que en caso de que falle alguna de las aplicaciones, el resto del sistema seguirá funcionando normalmente. Esto da como resultado más confiabilidad, un uso más satisfactorio por parte del usuario. Lo más importante es que esta confiabilidad puede tener un impacto directo en la organización.

¿Qué tan confiable es Windows NT Workstation Ver. 4.0?. Un estudio independiente sobre la productividad empresarial de los usuarios en grandes y medianas organizaciones fue realizado recientemente por la compañía de investigación de mercados Intersearch, el cual revela lo siguiente que muestra la gráfica siguiente:

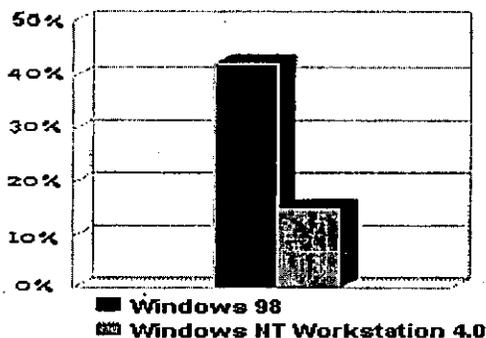


Fig. 6.1. Los usuarios de Windows NT Workstation sólo tienen la mitad de probabilidades de perder tiempo productivo o datos porque su PC haya dejado de funcionar adecuadamente. Fuente: Microsoft. CD Service Pack 4 para Windows NT ver. 4.0

Casi todos los usuarios (97%) piensan que las PC que se ejecutan en sistemas operativos Windows de 32 bits son confiables.

Los usuarios de Windows NT Workstation 4.0 informan tenerle más confianza que a Windows 95.

Los usuarios de Windows NT Workstation 4.0 tienen tres veces menos probabilidades que los usuarios de Windows 95 de reportar pérdida de tiempo o de datos debido a que sus PC dejaron de funcionar adecuadamente.

Del subgrupo de usuarios que dejaron su sistema encendido en todo momento, los usuarios de Windows NT Workstation 4.0 tienen tres veces menos probabilidades que los usuarios de Windows 95 de perder tiempo productivo o datos debido a que su PC dejó de funcionar adecuadamente.

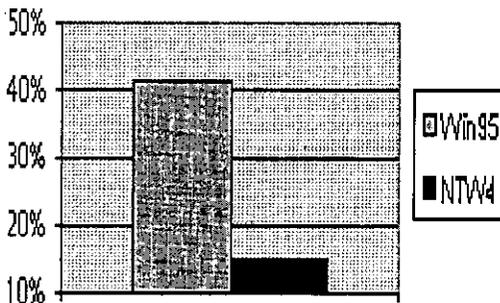


Fig. 6.2. Frecuencia con la que dejó de funcionar la PC normalmente en un periodo de estudio de un mes. Fuente: Microsoft (CD Service Pack 4.0 para Windows NT ver 4.0)

6.6 Medidas de rendimiento y seguridad de las Aplicaciones Cliente/Servidor

Rendimiento y seguridad son elementos íntimamente relacionados entre sí. La seguridad durante el diseño se puede minimizar y será un factor que determine si la aplicación Cliente/Servidor sea o no más rápida, ya que es un factor para incrementar el rendimiento de esta. Y viceversa incrementando los puntos de seguridad bajará la rapidez de respuesta de la aplicación Cliente/Servidor.

Debido a que las aplicaciones Cliente/Servidor se ejecutan en redes y todos los usuarios de esta potencialmente tendrían acceso a todos los recursos si no hubiese una seguridad adecuada. Creo que un justo medio entre ambos aspectos y acorde a las necesidades de seguridad de la organización será el punto óptimo.

6.6.1 El Rendimiento

El rendimiento de las aplicaciones Cliente/Servidor en Windows NT dependerá entre otros aspectos también de la configuración lógica y física que guarden sobre la red los clientes y los servidores. Por tal razón es importante conocer cual es la funcionalidad que tienen en la red cada uno de los elementos que proporcionarán servicios y reciben peticiones por parte de las aplicaciones Cliente/Servidor.

Como ya vimos dentro de una red Windows NT las computadoras destinadas a ser servidores son de 3 tipos³⁵:

- **Primary Domain Controler (PDC) Controlador Primario de Dominio.** La cual es responsable del mantenimiento de las cuentas de los usuarios.
- **Backup Domain Controler (BDC) Controlador Respaldo de Dominio.** Respalda al servidor responsable del mantenimiento de las cuentas de los usuarios.
- **Member Server o Servidor de aplicaciones.** Es el responsable de albergar las aplicaciones.

Es posible que todas estas entidades se estén desempeñando en el mismo equipo de cómputo lo que por lo general ocurre en organizaciones pequeñas por falta de presupuesto suficiente. Y esto repercute en el desempeño de las aplicaciones Cliente/Servidor sin duda.

El rendimiento dependerá de la labor que se haga en los siguientes rubros:

- Ubicación física
- Administración de usuarios
- Monitoreo.
- Herramientas de monitoreo.
- Red.
- Calculo del número de usuarios soportados por una aplicación.
- Cantidad de Elementos a monitorear en un servidor de aplicaciones.
- Medidas de rendimiento adicionales.

■ Ubicación física

La ubicación física de un servidor también repercutirá en el desempeño de una aplicación Cliente/Servidor. Ya que si determinados que una aplicación esta encomendada a un grupo particular de usuarios solamente estos obtendrán mejor respuesta en tiempos si el servidor esta más cerca físicamente de los clientes. Por lo que si es necesaria su reubicación como requisito de rendimiento de la aplicación debe realizarse.

Generalmente los problemas típicos que se generan no aparecen en los componentes individuales del sistema sino en los vínculos que los conectan. Esto esta relacionado con un principio fundamental de desempeño que es el principio de **localidad** el cual destaca la importancia de la cercanía y este principio enuncia simplemente lo siguiente "Entre más cercano es más rápido", así una técnica fundamental de este principio es reunir aquellos

³⁵ El tipo de hardware recomendado para servidores fue mencionado en el apartado 2.5.1

componentes que deban de estar juntos para que trabajen juntos, ya sea en el mismo disco, en el mismo servidor, o en la misma red LAN.

Existen cuatro tipos de localidad los cuales tienen influencia sobre el desempeño

- Localidad referida al espacio.
- Localidad referida al tiempo.
- Localidad referida al grado.
- Localidad referida al efecto.

Localidad referida al espacio.- Es la propiedad de poder hacer que los componentes de la aplicación se encuentren los más cercanos posibles unos de otros.

Localidad referida al tiempo.- Es la propiedad de poder realizar el menor tiempo posible en cuanto a la comunicación con los diversos componentes de la aplicación Cliente/Servidor.

Localidad referida al grado.- Es la propiedad de tener una adecuada capacidad de infraestructura informática que soporte debidamente a la aplicación.

Localidad referida al efecto.- Esta propiedad es relacionada específicamente con el propósito o fin que la aplicación persigue, es decir la aplicación debe de impactar realmente aquellas áreas deseadas de la organización.

Necesariamente una aplicación Cliente/Servidor que se desempeñe bajo una red Windows NT estará sometida a las condiciones y estructuras lógicas que gobiernan una red Windows NT.

■ Administración de usuarios

Poder mantener una administración adecuada de las cuentas de los usuarios de un servidor (En caso de que sea un servidor con múltiples funciones) es fundamental, mantener un control sobre el crecimiento de la base de datos de usuarios para que esta no llegue a crecer demasiado y golpee también el desempeño del servidor y repercuta negativamente en el desempeño de las aplicaciones Cliente/Servidor. La tabla siguiente muestra el espacio ocupado por las cuentas de usuarios en una red Windows NT.

Objeto	Espacio Usado
Cuenta de Usuario	1.0 KB
Cuenta de Computadora	0.5 KB
Cuenta de Grupo Global	512 Bytes más 12 Bytes por cada usuario
Cuenta de Grupo Local	512 Bytes más 36 Bytes por cada usuario

El tamaño del incremento del SAM (base de datos de las cuentas y perfiles de los usuarios) en un servidor Windows NT ver. 4.0 dependerá de los siguientes factores:

- **Tipo de micropcesador.** Un Pentium procesará validaciones de cuenta de usuarios más rápido que un procesador 486.

- **De la cantidad de memoria del servidor.** Una computadora de 32 MB de RAM no podrá soportar tantos usuarios como una de 128 MB de RAM.

La base de datos de los usuarios (SAM) por recomendación nunca debe de rebasar los 40 MB de espacio. De igual manera habrá que mantener un control sobre los diferentes dominios en los que esta organizada la red Windows NT.

■ Monitoreo

Antes de estar presente desempeñándose totalmente en la red Windows NT la nueva aplicación Cliente/Servidor, es conveniente realizarle simulaciones junto con un monitoreo de los momentos de más alta demanda para determinar su grado de afectación al desempeño del servidor y a la misma red. Para que una vez liberada completamente no provoque cuellos de botella que reduzcan el desempeño general de la red y de los servidores.

Herramientas de monitoreo

Poder determinar cuellos de botella y reducir o eliminar estos va a mejorar definitivamente el rendimiento. Debemos de utilizar herramientas especiales para este fin y olvidarnos de utilizar como primera opción métodos empíricos o indirectos para determinar las causas de los cuellos de botella que estén afectando el rendimiento de una aplicación Cliente/Servidor en la red. Esto nos provocará tal vez más cuellos de botella, debido a que no sabremos la causa exacta del bajo rendimiento. Hago especial énfasis en este punto debido a que es frecuente utilizar estos métodos empíricos como primer recurso ante una situación nueva de bajo rendimiento. Mi recomendación es que es plenamente justificado utilizar estos métodos cuando el problema ya es conocido.

Para monitorear el rendimiento en una red Windows NT el **Performance Monitor** y el **Network Monitor** son herramientas especiales muy útiles para poder apreciar el porcentaje y la cantidad de uso que tienen los diferentes recursos de una red y un servidor cuando están atendiendo una aplicación Cliente/Servidor.

Para poder monitorear los recursos utilizados por una aplicación, la herramienta de monitoreo utiliza contadores y objetos los cuales están relacionados con el equipo y la aplicación que se esta estudiando. No hay que preocuparse mucho por determinar cuales son estos contadores ya que la propia herramienta de monitoreo hace esta tarea por nosotros colocando todos aquellos recursos medibles solo habrá que hacer un selección posterior para realizar el en estudio de desempeño.

Ventajas de monitorear

Cuando se monitorean los recursos de la red es posible saber que servicios las aplicaciones Cliente/Servidor evocan y cuáles de estos provocan más tráfico sobre la base de estas medidas es posible pronosticar la medida del crecimiento futuro de la red y de las aplicaciones Cliente/Servidor.

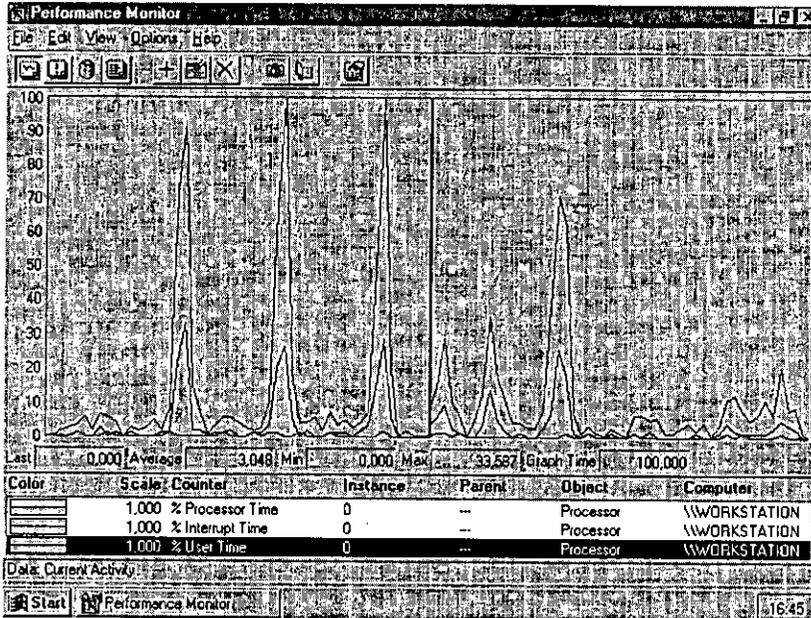


Fig. 6.3 Herramienta incluida con Windows NT denominada Performance Monitor.

■ Red

La optimización de la red, es un factor que influirá favorablemente sobre el desempeño de la aplicación Cliente/Servidor que se proyecte implantar.

La tarjeta adaptadora de red afecta en forma significativa el rendimiento de las aplicaciones Cliente/Servidor sobre la red. Si la tarjeta es lenta, los datos manipulados por la aplicación Cliente/Servidor no se transmitirán con velocidad sobre la red. Por ejemplo en una red de computadoras de topología Bus, en la que nadie puede utilizar el medio de transmisión, es decir. El cable, sino hasta que este es liberado por la aplicación en turno, una tarjeta adaptadora de red lenta puede aumentar considerablemente el tiempo de espera de las aplicaciones Cliente/Servidor ya sea en los clientes, en los servidores o en ambos simultáneamente. Por lo que después de conocer los requisitos físicos y las características técnicas de la tarjeta de red que se ha decidido utilizar es necesario considerar los siguientes puntos que afectan las capacidades de la tarjeta adaptadora de red.

Puede mejorar la velocidad de transmisión de las tarjetas de red si estas son capaces de soportar los siguientes métodos:

- **Acceso directo a memoria DMA.** Así colocará directamente los datos contenidos en el buffer de la tarjeta de red a la memoria del equipo sin pasar por el CPU.
- **Memoria compartida del adaptador.** Como la tarjeta de red tiene cierta cantidad de memoria RAM que comparte con la del equipo. El equipo reconoce dicha memoria como si estuviese instalada en la propia computadora.

- **Memoria de sistema compartida.** Aquí la tarjeta de red utiliza una parte de la memoria del equipo para procesar datos.
- **Control Bus.** Mediante este método la tarjeta de red toma el control de manera temporal del Bus de la computadora y mueve los datos entre su buffer y la memoria de la computadora.
- **Almacenamiento en buffer RAM.** El tráfico de la red viaja demasiado rápido que en general es demasiado para la mayoría de las tarjetas de red, así que le puede incrementar la memoria del buffer con chips extras.
- **Microprocesador en la tarjeta.** Colocándole un microprocesador a la tarjeta de red siempre y cuando lo permita, no necesitará de la ayuda del procesador del equipo de cómputo para administrar los datos.

La mayoría de los métodos antes mencionados muchas veces pueden ser invocados a través de su propio programa controlador de tarjeta de red. En cual por lo general si se trata de un buen controlador será instalado en el directorio Panel de Control de Windows.

Aunque todas las tarjetas de red se adaptan a ciertas especificaciones estándares, algunas características adicionales de la tarjeta pueden mejorar considerablemente el rendimiento de un servidor y por lo tanto también de las aplicaciones Cliente/Servidor.

■ **Calculo del número de usuarios soportados por una Aplicación**

Los usuarios que una aplicación Cliente/Servidor en el servidor pueda soportar, dependerá del hardware con que este cuente, y del análisis previo que se haya realizado, esto es desde el punto de vista de la infraestructura. Pero la aplicación Cliente/Servidor puede limitar también el acceso a los usuarios así misma esto se establece en la fase de desarrollo por medio de licencias. Sin embargo. Una manera muy directa pero que a la vez puede ser considerada superficial es utilizar las siguientes reglas:

- Probar con un sólo usuario que realice transacciones normales sobre el servidor utilizando su front-end contra la máquina servidor.
- Capturar los datos de rendimiento en el servidor utilizando los contadores³⁶ apropiados relativos a la aplicación y a los recursos del servidor que se desean medir (esto se realiza a través de la herramienta para monitoreo llamada Network Monitor).
- Multiplique las medidas de uso de los recursos (RAM, CPU, HDD, y RED) utilizados por un usuario o un grupo de usuarios y expándalas al número de usuarios que contemple atender normalmente.

³⁶ Estos son variables para almacenar los valores de comportamiento de los recursos del servidor.

■ **Elementos a monitorear³⁷ en un servidor de Aplicaciones Windows NT (o Member Server)**

Primero. Procesador.- Las aplicaciones Cliente/Servidor corriendo sobre el servidor desencadenan varios threads o hilos de proceso que el microprocesador se encarga de atender. Por lo que un gran número de usuarios entrando a las aplicaciones Cliente/Servidor en el servidor. Incrementará los valores de los contadores de monitoreo para el microprocesador.

Segundo. Memoria.- Este es el segundo elemento afectado cuando en un servidor de aplicaciones, debido que las aplicaciones Cliente/Servidor son muy demandantes de memoria RAM asegúrese de colocar suficiente memoria RAM, el propio "Network monitor" le dirá si hace falta hacerlo a través del análisis de su contador para memoria RAM.

Tercero. Subsistema de discos. Los discos duros y sus controladoras resentirán el uso de las aplicaciones Cliente/Servidor pero en menor medida si los discos duros y las controladoras son rápidas por ejemplo SCSI II. Aunque las aplicaciones Cliente/Servidor accedan a grandes cantidades de lecturas y escrituras de información.

Cuarto. Subsistema de Red. Por último quien menos se ve afectado es la red, ya que normalmente solo es utilizada para enviar los datos ya procesados.

Sobre la base de esto diremos que cuando el estudio que se realice con las herramientas de monitoreo arroje resultados tendientes a formar cuellos de botella en el momento que se encuentren ejecutando aplicaciones de 16 bits la recomendación será instalar un **microprocesador más potente**. Y por el contrario si el cuello de botella lo provoca la ejecución de las aplicaciones de 32 bits habrá que agregar un **microprocesador adicional** al servidor.

Medidas de rendimiento adicionales

- Implementar un arreglo de discos en el servidor así se distribuirá la carga de trabajo de los accesos de lectura y de escritura a los discos.
- Implementación de Clustering Servers (varios servidores para atender las solicitudes de las aplicaciones Cliente/Servidor).
- Eliminación de protectores de pantalla en los servidores de aplicaciones.

6.6.2 La Seguridad

Para adoptar un plan de medidas óptimo se debe de realizar un estudio de seguridad donde se contemplarán los posibles riesgos a la seguridad de la información que pueden producirse al trabajar con las aplicaciones Cliente/Servidor ya implantadas. Ya que si el diseño de la aplicación no es el adecuado, puede estar propenso a errores que dejen escapar o no permitan obtener información aun a los usuarios autorizados, además de que no responderá a las mínimas necesidades de seguridad del usuario y tal vez como consecuencia se tenga demasiada dependencia de las personas que han desarrollado la aplicación.

³⁷ Debemos mencionar que los elementos a monitorer en cada extremo(cliente o servidor) variarán dependiendo de la arquitectura Cliente/Servidor que se elija ver Fig. 4.3

En cada una de las siguientes fases del estudio de seguridad se incluyen una serie de requisitos que permitirán realizar un itinerario de seguridad completo.

Análisis Previo.- Aquí se realiza un estudio de los riesgos que puede producir la aplicación Cliente/Servidor en la etapa de desarrollo si no se contemplan ciertos puntos generales de seguridad o muy particulares de la organización. Algunos de estos riesgos son: Personal que trabajará en el desarrollo de la aplicación Cliente/Servidor y sus repercusiones en la empresa por ejemplo si a través de este personal puede producirse fuga de información estratégica, información de recursos etc.

Análisis Funcional.- En este análisis se detallan todas las especificaciones del *análisis previo*. Se crea una documentación donde se registra el diseño de los datos y las características del itinerario de seguridad. De esta forma se adecua la seguridad de la aplicación a las necesidades del usuario y evitará también la obsolescencia de la seguridad de la aplicación cuando se realicen futuras modificaciones.

Análisis Orgánico.- Es determinar aquellas unidades de programación que resultan críticas para la seguridad debido a la información que procesan o por la probabilidad de la aparición de errores, este análisis Cliente/Servidor se complementa con la determinación de los procesos de recuperación. En las aplicaciones también se deben de incluir controles que sirvan para la verificación de los datos de entrada y para los accesos a la información confidencial. La utilización de "passwords". Es decir. Las palabras clave para los accesos deben de reforzarse más en las aplicaciones Cliente/Servidor que en las que no lo son, aunque estas claves también pueden ser vulneradas. Resultará más difícil si pretendemos sistemáticamente de hacerlas menos propensas a sufrir este tipo de vulnerabilidades. Es necesario seguir ciertas reglas para la construcción de claves por parte de los usuarios las cuales deben ser soportadas por las aplicaciones Cliente/Servidor, por ejemplo un tamaño de cadena de caracteres de 6 a 25 caracteres para las claves y el no uso o el uso de cierto tipo de caracteres denominados especiales que debe de poder siempre aceptar la aplicación Cliente/Servidor. Mencionare por ejemplo algo que no puede ser utilizado como password y cuya validación la puede realizar la misma aplicación mediante el chequeo de la información del usuario en una base de datos personales para de esta forma advertirle al usuario que puede ser riesgoso la utilización de tal palabra como password.

- Números telefónicos
- Números de identificación
- Nombres conocidos (el propio, el de la esposa, el de los hijos, el de la mascota, etc.)

Realizando estos análisis se puede comprobar las rutinas de seguridad de la aplicación, así como el correcto funcionamiento de estas. La aplicación Cliente/Servidor previo a su implantación se deben de someter a estas pruebas. Utilizar un juego de ensayos para realizar pruebas a la seguridad puede ser muy conveniente.

Al realizar pruebas de seguridad antes de poner en marcha la aplicación y realizarlas también una vez este implantada la aplicación. Es posible tener una máxima certeza de que la aplicación Cliente/Servidor cumple con los requerimientos de seguridad establecidos en la organización.

Dejar constancia de todos estos estudios y pruebas en una documentación adjunta a la general de la aplicación Cliente/Servidor es otra medida más de seguridad.

La plataforma de sistema operativo para el desarrollo de aplicaciones Cliente/Servidor de Microsoft es esencialmente Windows NT, siempre el rendimiento final de una aplicación no solo estará supeditado por el diseño de la aplicación sino también por el sistema operativo en el se este ejecutando la aplicación.

Desde el punto de vista del cliente, NO hay duda que **Windows NT Workstation ejecuta más rápido y de forma más confiable las aplicaciones de 32 bits que las de 16 bits**. Sin embargo, se pueden ejecutar una amplia variedad de aplicaciones de 16 bits en forma más rápida y confiable, ya que cada aplicación de 16 bits puede ejecutarse como un proceso separado multi-tasking en su propio espacio de memoria y aislado de otras aplicaciones activas. Este proceso (multi-tasking) también acelera la velocidad y respuesta de la aplicación, y proporciona máxima protección de datos y de las aplicaciones al tiempo que conserva la inversión monetaria realizada en ciertas aplicaciones críticas para la organización de 16 bits.

Asimismo, Windows NT Workstation 4.0 da soporte a diversos hardware y periféricos de los más populares en todo el mundo: más de 9,000 marcas de PC de escritorio y laptop, y aproximadamente 6,500 periféricos y dispositivos incluyendo tarjetas de red, módems, adaptadores de vídeo, exploradores y computadoras centrales. Los nuevos dispositivos se prueban constantemente y el soporte se agrega en un promedio de quince productos por semana.

Windows NT se ejecuta en arquitecturas basadas tanto en Intel como en RISC, proporcionando flexibilidad máxima y disminuyendo el número de plataformas de sistemas operativos que usted necesita soportar.

Nota final

Para que tengan el mejor desempeño en aquellas aplicaciones Cliente/Servidor que utilizan ODBC, el desarrollador debe de comprender muy bien la arquitectura del ODBC, ya que sin entender esto las cuestiones de desempeño se convertirán en materia de discusión. Entendiendo esta arquitectura las aplicaciones Cliente/Servidor y los controladores pueden ser afinados de maneras muy específicas a fin de ampliar su nivel de desempeño.

6.7 Seguridad en los datos que desplazan las Aplicaciones Cliente/Servidor

Si hay, o no hay línea en el momento en que la estación cliente desea enviarle un mensaje al servidor, no es algo que tenga que repercutir o preocupar a la aplicación Cliente/Servidor ya que el propio software de middleware se encargará de enviar nuevamente el mensaje cuando vuelva a haber línea. Mientras no haya línea los mensajes quedan almacenados ya que las colas son almacenadas en el disco duro de la máquina ya sea la máquina cliente o la máquina servidor, de tal manera que el proceso de envío de información se hace inmune a caídas de los clientes o caídas de los servidores. Si se cae un servidor o una estación cliente, los mensajes permanecen en el disco duro y cuando vuelva a levantar el

servidor o el cliente, el middleware tratará de enviar los mensajes pendientes lo mismo ocurre si el servidor es incapaz de procesar las peticiones por saturación de demandas.

Los mensajes de la cola de salida en el cliente no son eliminados de esta, sino hasta que se recibe la confirmación de llegada en la cola de entrada del servidor.

Buenos ejemplos de estas negociaciones pueden ser encontradas en pruebas de **benchmark** de alta fiabilidad de transportes (TCP/FTP,SNMP etc.) y una muestra son aplicaciones personalizadas (Usando NetBIOS LU6.2 etc.) Para garantizar disponibilidad de información completa, algunas aplicaciones requieren del uso de múltiples servidores intermedios como puntos de unión. Esta ordenación de almacén y mecanismos de envío proporcionan un nivel adicional en la integridad de los datos de esta forma los servidores intermedios actúan como un **buffer** en la cola de tráfico de la aplicación Cliente/Servidor.

6.8 Seguridad en las Aplicaciones Cliente/Servidor

Seguridad de la Información

Cuando se diseña una aplicación Cliente/Servidor y se comparte información a través de una aplicación distribuida, la integridad de los datos es un asunto predominante. Si múltiples bases de datos son conectadas a la aplicación, el diseño debe de asegurar que la información este correcta. Una simple actualización nocturna en la base de datos del mainframe desde el sistema manejador de bases de datos (DBMS) de la LAN puede satisfacer este requerimiento. Si la aplicación requiere una mayor frecuencia de intercambio de datos, la tarea será localizada al nivel de red o de workstations, donde se puede tomar ventaja de las facilidades que ofrece el (DDE) intercambio dinámico de datos por sus siglas en ingles (Dynamic Data exchange).

Desempeño

Para la total evaluación de la capacidad de un diseño, se considera la posibilidad de falla debido a los altos volúmenes de transacciones y a las condiciones de error. Para evitar esas caídas identifique los puntos altos de tráfico y considere alternativas de ruteo dentro de la aplicación para que la aplicación Cliente/Servidor pueda tomar ventaja del software comercial de ruteo y de ruteadores inteligentes.

El análisis del flujo de datos debe también direccionar el volumen de datos, velocidad y duplicación de datos. El volumen de datos es un elemento tangible que debe de ser medido y planeado sobre la base de "el caso del peor escenario". Esto es importante para el desarrollo adecuado de la transferencia de almacenaje de los datos y para la planeación de los chequeos de fiabilidad.

Probar el software que tiene como propósito la conectividad (**middleware**) para la transferencia de altos volúmenes de datos y así asegurar que los niveles de rendimiento sean aceptables, es un importante factor en el rendimiento y en la velocidad de acceso, la cual a

menudo es un bache para la fiabilidad, por lo que la meta esta en saber equilibrar entre los apropiados tiempos de acceso y de fiabilidad.

6.8.1 Protección contra Virus.

El tema de los virus es algo que no puede faltar en ningún estudio de seguridad. Los virus dañinos en redes Windows NT es raro de encontrarlos, debido a la propia estructura de seguridad de Windows NT que no permite el acceso directo al hardware por parte de las aplicaciones esta cualidad libra de manera natural en buena medida de estos contagios. Sin embargo. NO se debe de confiarse y debe de tenerse en mente el posible contagio cuando se desarrollen procedimientos de seguridad para las redes. En Windows NT estarán más propensos a ser afectados por los "gusanos informáticos". Lamentablemente ningún programa antivirus informático puede impedir el ataque de estos. Ya que solo pueden evitar que el virus se active, eliminar el virus, reparar los daños en la medida de lo posible o mantener bajo control el virus una vez que se ha activado, pero un "gusano informático puede pasar desapercibido.

Hay que tener bien entendido que Windows NT no esta libre de ser infectado por virus informático. Y por el sólo hecho de tener instalado este sistema operativo no quiere decir que estemos exentos de sufrir contagios por virus en los equipos de cómputo. La ventaja de contar con Windows NT es que los virus que ingresen no se activaran mientras estén bajo Windows NT. Es decir. Si es ingresado un archivo infectado con virus informático a una red Windows NT el archivo infectado existirá como tal y el virus también. Windows NT por si solo no detecta ni elimina virus alguno, por lo que se solo observará un comportamiento anormal cuando se trabajen con ese archivo por ejemplo que se trabe la aplicación sin ninguna razón, que no se pueda ni abrir el archivo, etc. Por lo que es recomendable instalar vacunas informáticas para Windows NT.

Una medida de seguridad adicional es convertir las particiones de los discos duros a NTFS en vez de dejarlas en FAT que es la opción por default al instalar Windows NT, mediante el comando de consola **convert.exe**, puede solicitar el juego de opciones con la /, en esta forma **convert /?**.

La mejor manera de evitar el ataque de virus es impedir el acceso no autorizado mediante las siguientes medidas:

- Contraseñas para reducir las posibilidades de accesos no autorizados.
- Cuidadosamente asignar accesos y privilegios a los usuarios.
- Crear perfiles de usuario para los entornos de red para mantener constante un entorno de inicio de sesión en donde se incluyan también las conexiones de red y los iconos de acceso para las aplicaciones cuando se inicie la sesión.
- Reglas que determinen el software que se puede cargar localmente.
- Reglas para establecer (configurar e instalar) vacunas informáticas tanto en los servidores como en los clientes.

Control de los costos en la Ejecución de las Aplicaciones Cliente/Servidor

7.1 Introducción

Conocer como calcular los costos económicos presentes y futuros que representará para una organización el implantar un entorno donde se ejecuten aplicaciones Cliente/Servidor y el costo mismo de las aplicaciones Cliente/Servidor es un aspecto final y muy importante que abordare en este trabajo, debido a que de este aspecto dependerá la completa viabilidad del proyecto. Una aplicación que resulte demasiado costosa en su diseño, en su mantenimiento y en los requerimientos de arquitectura terminará por desaparecer.

Debemos de planificar los costos en los entornos Cliente/Servidor por sus características muy particulares los costos muy fácilmente pueden salirse de control por una inadecuada planificación. En los entornos Cliente/Servidor es donde más recursos de cómputo convergen por esta razón es de suma importancia tocar este aspecto. Aquí intervienen distintos tipos de costos principalmente son: Los costos de hardware, los costos del software, los costos de desarrollo y los costos de capacitación, en general se pueden dividir en dos. Los costos de arquitectura y los costos de infraestructura.

7.2 Presupuestos

Para quienes tienen asignado un presupuesto propio para las cuestiones de cómputo es conveniente que tomen en cuenta una cierta planificación para que puedan dar cabida a todos los requerimientos monetarios que el proyecto de implantar aplicaciones Cliente/Servidor produzcan. Esta tarea en ocasiones recae en los líderes de proyecto que por lo general no tienen suficiente conocimiento de procesos administrativos financieramente hablando. Aclaro que no pretendo profundizar en estos menesteres presupuestales puesto que tampoco es mi especialidad. Me limitare simplemente a abordarlo superficialmente. Es importante que el líder de proyecto se asesore en esta materia por gente especializada.

Un presupuesto es un plan de acción en términos monetarios que abarca un periodo de tiempo específico. Esto sirve para la gestión financiera de una organización, en el que se especifica cuanto dinero debe de gastarse en bienes y servicios además de cuando se debe de gastar.

El presupuesto señala los gastos a realizar durante un tiempo determinado, el control preciso de los gastos permite que se apliquen en aquellos momentos en que son más benéficos para la organización

Los presupuestos pueden ser de tres tipos:

- **De operación.**- Indica las operaciones planificadas para el período siguiente.
- **De efectivo.**- Indica las fuentes y aplicaciones previstas para el dinero en efectivo.
- **De capital.**- Muestra los cambios planificados para diversos recursos, además del efectivo.

El presupuesto es un instrumento de control y planificación para operar sobre la rentabilidad. El proceso de desarrollo del presupuesto del área de sistemas varia según el método que emplee la organización. Cuando se vaya a preparar el presupuesto destinado al área de sistemas debe de contemplarse en ese momento todos los costos relacionados a la implantación de las nuevas aplicaciones Cliente/Servidor, para que la dirección considere el posible impacto de los diferentes factores como son:

- Avances tecnológicos.
- Costos de mantenimiento más elevados.
- Escasez de recursos.
- Aumento o disminución de los procesos.
- Creciente utilización de las telecomunicaciones.
- Cambios en las aplicaciones.
- Mantenimiento a cargo del personal propio de organización.
- Utilización de programación contratada.
- Cambios en la productividad de la organización.

7.3 Costos de hardware en Cliente/Servidor.

En la adquisición de hardware intervienen consideraciones de carácter financiero, la adquisición de hardware es una relación entre el costo y la vida útil de este. Actualmente se vive una aceleración en innovación tecnológica bastante acentuada en la que en cuestión de unos pocos meses se hace obsoleta la llamada "tecnología de punta", esta curva de aceleración tecnológica debe tenerse muy en mente en los momentos de trabajar los costos para la adquisición de hardware. Actualmente se aprecia que cada trimestre se esta incrementando en un rango que va de 1 a 2 peldaños las etapas de velocidad de los microprocesadores compatibles con Intel, siendo estas etapas cada vez más distantes una de otra (de su inmediata anterior), Esto es debido fundamentalmente a la utilización de una nueva tecnología en la reducción de los componentes de los microprocesadores. Por ejemplo en el mes de Abril a Mayo de 1999 la velocidad comercial tope de un microprocesador era de 450Mhz siendo 6 meses después la velocidad tope de 750Mhz. Por supuesto que el rendimiento total de una computadora no dependerá del microprocesador sino también de los subsistemas de discos, los subsistemas de red, y de la memoria RAM, como ya mencione anteriormente. Pero este aspecto nos da pauta para poder estudiar a más consciencia el grado de obsolescencia que sufrirá el hardware y ser uno de los aspectos básicos en la adquisición.

El método de adquisición es determinado independientemente de la selección del equipo, y puede caer en cualquiera de las dos opciones siguientes:

- **Compra.-** Esta es la mejor opción si siempre se trabajará con la aplicación. Existe la adquisición mediante recursos propios y la adquisición mediante recursos ajenos. Esta es una opción debido a que no se generan gastos constantes mensuales y desde su adquisición se considera patrimonio de la empresa.
- **Alquiler.-** Representa ciertas ventajas e inconvenientes, dentro de las primeras están: **1.** No existe un desembolso inicial de capital, **2.** Reduce al máximo el peligro de obsolescencia tecnológica, porque puede cambiarse la computadora en cualquier momento.
Los inconvenientes son: **1.** En un contrato demasiado largo el costo del alquiler puede rebasar el costo del equipo, **2.** El costo del alquiler es una salida fija de la caja, **3.** El alquiler puede subir
- **Alquiler con opción a compra.-** Esto es a través de un contrato de financiamiento en el que se efectúan pagos mensuales al finalizar se adjudica el equipo.

Cada una de las alternativas consideradas presenta desembolsos diferentes

La siguiente tabla muestra configuraciones para ser destinadas a un equipo cliente que soportarán con facilidad la mayoría de aplicaciones Cliente/Servidor-estándar.

Tabla de cualidades a considerar para la comparación de equipos que pueden fungir como cliente.

	Opción A	Opción B	Opción C	Opción D
Procesador	Intel Pentium II Celeron	AMD K6-2	Intel Pentium II Celeron	Intel Pentium II
Velocidad del reloj	466Mhz	450Mhz	433Mhz	400Mhz
Capacidad de memoria central	96 MB (SDRAM)	64 MB (SDRAM)	96 MB (SDRAM)	64 MB (SDRAM)
Capacidad de Disco Duro	15 GB Ultra DMA	8 GB	9.6 GB	6.4 GB
Tipo de Monitor	Color SVGA, 17", 1MB Video RAM	Color SVGA, 14", 4MB Video RAM	Color SVGA, 14", 8MB Video RAM	Color SVGA, 14", 4 MB Video RAM
Interfaces	2 Puertos RS-232 1 Paralelo 4 USB	2 Puertos RS-232 1 Paralelo	2 Puertos RS-232 1 Paralelo 2 USB	2 Puertos RS-232 1 Paralelo 2 USB
Sistema Operativo	MS Windows 98*	MS Windows 98*	MS Windows 98*	MS Windows 98*
Ampliaciones de Memoria RAM, Video RAM	384 MB,4MB	384 MB,4MB	384 MB,4MB	768 MB,4MB
Accesorios	CD ROM 32X,Fax/Modem, Zip Drive 100MB	CD ROM 32X,Fax/Modem,	CD ROM 32X,Fax/Modem	CD ROM 32X,Fax/Modem
Tarjeta de Red	10/100Mbps	10/100Mbps	10/100Mbps	10/100Mbps
Marca	Compaq	Compaq	HP	IBM
Precio con configuración Mínima	1,805 USD	1,290 USD	1,479 USD	1,791 USD

* Este sistema operativo viene instalado de fabrica. Se sugiere realizar una consulta con el vendedor a fin de adquirir el equipo con Windows NT Workstation Ver. 4.0 preinstalado, esto incrementa un poco el precio final aquí expuesto.

7.3.1 Tipos de Costos

Debemos de contemplar en nuestra planificación de costos a aquellos costos que no son tan visibles pero que, sin embargo, son considerables, como los que se mencionan a continuación:

- **Costos de desarrollo.**- Son aquellos generados por el trabajo de programación y de formación de personal.
- **Costos de operación.**- Son aquellos generados por el personal que trabaja en operación de la aplicación Cliente/Servidor por ejemplo mantenimiento de esta y de la base de datos, instalaciones etc.
- **Costos de mantenimiento.**- Son aquellos que se destinan, al cuidado, conservación y reparación del sistema tanto hardware como software.

Una distribución porcentual típica de costos totales que efectúa un departamento de sistemas en el la parte de aplicaciones Cliente/Servidor es la siguiente:

- Costos por mantenimiento de hardware. 22%
- Costos del equipo auxiliar. 2%
- Costos del personal de desarrollo. 35%
- Mantenimiento de las aplicaciones Cliente/Servidor. 10%
- Material de papelería. 3%
- Formación de recursos humanos tanto personal de desarrollo como usuarios de las aplicaciones. 20%
- Mobiliario. 8%

La siguiente tabla es una estimación de costos entre Microsoft y SUN. De acuerdo con esta tabla se puede observar que un servidor trabajando sobre Windows NT es 1.9 más barato que un servidor trabajando sobre Solaris (Unix).

Componente	Microsoft	Costo Total	Sun	Costo total
Sistema Operativo	Windows NT Server 4.0	\$2950 2950	Solaris (incluido en el servidor)	\$0
Servidor Internet	Internet Information Server 3.0	Gratis (incluido en Windows NT)	Netscape Enterprise Server 3.0	\$1,295.00
Correo y Colaboración	Microsoft Exchange 5.0 Enterprise Server	\$6400 6400	Lotus Notes Domino 4.5 Mail Server	
Base de Datos	SQL Server 6.5	\$10650 10650	Oracle, Informix o Sybase database	\$50,000.00
Procesamiento de Transacciones	Transaction Server	\$2000 2000	Oracle Web Application Server	\$995.00
Proxy Server	Proxy Server	\$4995 4995	Netscape Proxy Server	\$995.00
Costo Total		\$26,995.00		\$53,285

*Costos expresados en M/N al mes de noviembre de 1998. -Fuente:Resumen "Microsoft en la Administración pública", septiembre de 1998, ciudad de México.

7.4 Fórmula para calcular los costos de implantación de Aplicaciones (Cliente/Servidor)

Una fórmula para poder calcular los costos puede ser la siguiente: (esta fórmula es aplicable cuando se parte desde nada para la implantación de aplicaciones Cliente/Servidor)

Costo de Implantación:

$$C_{EC} + C_{IR} + C_{ES} + C_{IO} + C_A + C_{IA} + C_{MA} = CT_{ICS}$$

C_{EC} = Costo de equipo Cliente (PC), este costo debe de multiplicarse por el número de clientes que se decida comprar, esta decisión dependerá de la capacidad de la aplicación Cliente/Servidor.

C_{IR} = Costo de infraestructura de red, son todos los elementos de hardware y software que permiten el enlace básico, así como la comunicación básica.

C_{ES} = Costo del equipo servidor (Servidor).

C_A = Costo de la Aplicación. Comprende el costo de desarrollo total. Cabe mencionar que el desarrollo de una aplicación Cliente/Servidor crece en complejidad en la medida que debe de controlar/negociar con más clientes y negociar con más servidores, por lo tanto también su costo de desarrollo total aumentará. Aquí se incluyen el costo de las diferentes herramientas de desarrollo y también herramientas auxiliares.

C_{IO} = Costo de la Infraestructura Operativa. Es el costo del sistema operativo de los clientes y del servidor, así como el DBMS y las herramientas auxiliares.

C_{IA} = Costo de la Instalación de la Aplicación. Serán los gastos que implique contratar personal para instalar y configurar en cada cliente la aplicación.

C_{MA} = Costo del mantenimiento de la aplicación. Son los gastos que producirá después de su instalación, ya sea adecuaciones o ampliaciones.

CT_{ICS} = Costo Total que produce la Implantación Cliente/Servidor.

7.5 Control de Costos de Administración de Aplicaciones Cliente/Servidor.

Los costos que podemos someter a control mediante una adecuada administración son los siguientes:

- Costo de los servidores.
- Costo de las estaciones de trabajo.
- Costo de la codificación.
- Costo de la capacitación (Es una inversión).
- Costos de downsizing (Si se van a reducir las aplicaciones de una minicomputadora o una computadora central a Workstations).

7.6 Control de los Costos de Ejecución de Aplicaciones Cliente/Servidor

La plataforma operativa básica para el desarrollo de aplicaciones Cliente/Servidor es Windows NT, los costos finales de la implantación de una aplicación Cliente/Servidor en un ambiente Cliente/Servidor, también dependerá del sistema operativo que se elija.

Windows NT Workstation ayuda a reducir gastos de soporte

La elección de un sistema operativo de escritorio confiable reduce considerablemente los gastos de soporte. Los usuarios de Windows NT Workstation 4.0, de todos los niveles, llaman al departamento de asistencia 38% veces menos en promedio que los usuarios de Windows 95. Como podemos apreciar en la siguiente gráfica. Con respecto a Windows NT Server el nivel de necesidad de asistencia será muy similar a Windows NT Workstation.

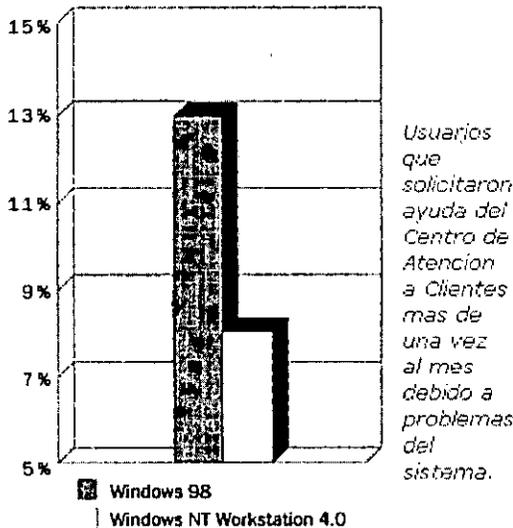


Fig. 7.1 -Fuente:Microsoft-

De acuerdo con un estudio independiente realizado por Technology Business Research, Windows NT Workstation 4.0 puede reducir los gastos de soporte en casi \$ 400,000 dólares al mes en las empresas más grandes. Asimismo, se descubrió que una organización común puede esperar un ahorro de \$483 dólares por usuario al año durante la vida útil del producto. El estudio atribuyó estos ahorros a la solidez, seguridad y facilidad de uso del software.

Microsoft dice que la familia Windows de sistemas operativos ofrece a los clientes lo mejor de ambos mundos: la flexibilidad de la computación distribuida con los beneficios administrativos de la computación centralizada.

Así mismo añade que el ambiente de 32 bits basado en Windows NT no sólo ofrece los TCO más bajos en la industria, sino también opciones, flexibilidad y compatibilidad. Esto se debe en gran medida a las millones de soluciones de hardware basadas en Windows y a las aplicaciones de software que están actualmente a la venta. Con el ambiente basado en Windows, los administradores tienen la opción de ejecutar las aplicaciones en forma local o completamente fuera del servidor, así como a través de la combinación de ambas formas. Con las nuevas tecnologías como el Zero Administration Kit, el Servidor de administración de sistemas e IntelliMirror en Windows NT 5.0, los administradores también tienen la capacidad de administrar equipos de escritorio desde una ubicación central. Windows NT ofrece los siguientes beneficios:

- La más alta flexibilidad de administración, desde equipos de escritorio completamente "bloqueados" o totalmente "abiertos". El más amplio conjunto de aplicaciones, dispositivos y hardware basados en estándares abiertos y bien definidos.
- Interfaz de usuario, aplicaciones y herramientas consistentes para usuarios finales, junto con herramientas universales de administración de clientes.
- Innovación continua en términos de administración remota y automática de usuarios, hardware, software e información de estado.

Esto permite que las organizaciones protejan las inversiones realizadas en la arquitectura Windows, en términos de capacitación, habilidades, desarrollo de aplicaciones y experiencia en Helpdesk, al tiempo que toman medidas importantes para disminuir los costos totales de propiedad. La arquitectura Windows es lo suficientemente flexible para proporcionar una solución de cliente dirigida a cada tipo de usuario, incluyendo usuarios orientados a tareas de una sola función, empleados con conocimiento, usuarios móviles y usuarios avanzados de high-end.

Y Microsoft finaliza diciendo que las características de administración disponibles en Windows NT Workstation 4.0 hacen que su administración sea mucho más fácil y menos costosa que la de cualquier otro producto de escritorio Windows. Eso significa que usted obtiene lo mejor de ambas partes: mayor potencia y confiabilidad, así como los costos totales de propiedad más bajos.

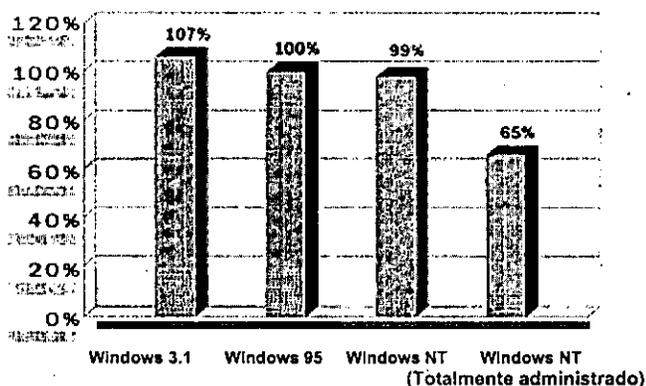


Fig. 7.2 Porcentaje de administración que requieren diferentes productos Windows. Fuente:Microsoft.

CONCLUSIÓN

El impacto positivo en reducir los TCO que una organización es capaz de lograr al implementar un ambiente de PCs bien administrado y basado en Windows, puede ser enorme y las reducciones en los TCO pueden ir de 25 a 35%, dependiendo de la configuración que se utilice. Un ambiente de PC bien administrado y basado en Windows ofrece los TCO más bajos disponibles, sin importar si está implementando plataformas de productividad personal o reemplazos de terminal.

7.6.1 Costo de la red.

Actualmente el objetivo de la red ha dejado de ser la conectividad (aunque es el fin fundamental) y se ha convertido en el medio necesario para traslado de grandes cantidades de datos para aplicaciones distribuidas operativas. Partiendo de este punto una forma para evaluar el costo de la red es sobre la base de la capacidad de trasladar datos, podemos decir que el costo evaluado en base a nodos es irrelevante, dado que es casi imposible determinar el ancho de banda que una LAN de acceso compartido puede proporcionar. Una forma prudente de medir el costo es por megabit utilizable, el cual determina el ancho de banda que se puede garantizar para un cliente o un servidor. Al utilizar una tecnología de conmutación, cada estación final cuenta con un ancho de banda dedicado. Es decir, con capacidad de red utilizable para llevar a cabo transacciones de datos. Sobre la base de costos por megabit, la conmutación resulta más económica que las redes locales de acceso compartido tradicionales.

Al migrar a una plataforma de aplicaciones Cliente/Servidor obtenemos las siguientes reducciones en costos.

- Reducción en los costos de procesamiento.

Mediante la adquisición de equipos de escritorio de gran capacidad de procesamiento, en lugar de comprar un gran equipo mainframe, de esta manera los costos se abaten en un rango de 10% a 20%³⁸ y se obtiene la misma capacidad de procesamiento.

- Reducción en los costos de desarrollo de aplicaciones

Mediante la utilización de sistemas operativos más poderosos tales como UNIX o Windows NT, que provee herramientas de desarrollo gráfica, capacidades de multitarea y capacidad de desempeño de varios procesos al mismo tiempo.

Se abaten gastos en las empresas generando una mayor productividad, desarrollando y liberando sus aplicaciones en menor tiempo, así a través del uso de tecnología abierta estándar

³⁸ Según estimación de grupo Scanda, empresa en México dedicada a consultoría y a la implementación de soluciones Cliente/Servidor.

las empresas se mantienen competitivas a la vez que se colocan en un punto donde es posible una migración posterior y adoptan tecnologías mucho más rápido y a menores costos.

7.7 Relación Costo-Beneficio de las Aplicaciones Cliente/Servidor

Aunque el avance tecnológico provoca que el hardware constantemente reduzca de precio esto aunado a los sueldos de las personas que operan este hardware, hace que los costos totales de operación aumenten considerablemente, las constantes necesidades de nueva información dentro de las organizaciones hacen que los ámbitos de cómputo tengan un constante movimiento económico en todas sus esferas. Por ello también aumenta la preocupación por encontrar una optimización en las relaciones de costo/beneficio.

Al igual que para otras cosas la medida de la efectividad de una aplicación Cliente/Servidor se mide sobre la base de dos beneficios obtenidos durante la vida útil de esta, los cuales se pueden dividir en **beneficios tangibles o directos** y los **beneficios intangibles o indirectos**.

Los beneficios tangibles. Se ven como ahorros para la organización y esto se debe fundamentalmente al código optimizado y a la mayor eficiencia de los procesos de la aplicación.

Los beneficios intangibles. Se consideran aquellos beneficios que no son provocados directamente por la aplicación por ejemplo mayor productividad de los usuarios, mejor control de calidad en la organización, respuesta más rápida a las preguntas de los clientes.

En un análisis se debe de comparar la efectividad de los beneficios directos e indirectos que ofrece la aplicación con las limitaciones de los recursos. Así este análisis determinará si la aplicación Cliente/Servidor aportará beneficios que superen a los costos, Normalmente se analizan algunas alternativas y una simple comparación nos dará la respuesta de cual es el mejor.

En el análisis costo-beneficio fundamentalmente se totalizan los costos de desarrollo e implantación de la aplicación y se comparan con los beneficios que resultan de utilizar la aplicación.

Sin embargo. La evaluación de los beneficios que aporta una aplicación es complicada, porque se pueden mostrar los ahorros tangibles en determinadas operaciones como son: un aumento en la productividad, aumento en la calidad de las tareas desempeñadas, utilización más eficiente de la red etc., sin apreciar una relación proporcional en los costos ya sea hacia abajo o hacia arriba.

En cambio no ocurre así con los casos en que el uso de la aplicación instalada es voluntario ya que aquí se facilita enormemente el análisis ya que su beneficio se evalúa en base al uso de la aplicación.

7.8 El Futuro de las Aplicaciones Cliente/Servidor.

La próxima generación de aplicaciones Cliente/servidor de misión crítica ya está próxima y pretenden consumir una parte sustancial de la capacidad de la red y por su parte los usuarios esperan mayor rendimiento y flexibilidad de las aplicaciones Cliente/Servidor de la próxima generación.

Actualmente las redes están siendo rediseñadas y en el futuro continuarán siéndolo a fin de que los ruteadores ya no se interpongan entre el Cliente y el Servidor, y retarden el tiempo de respuesta de las aplicaciones Cliente/Servidor ya que la mayoría de los ruteadores no fueron diseñados para soportar corrientes de transacciones Cliente/Servidor de escasa demora y alto rendimiento. Los ruteadores están asumiendo nuevamente su función original la de aislar redes de redes mayores y la interconexión de redes diferentes (Por ejemplo una red Ethernet y una red Token ring).

En 1989 un consorcio dedicado a la distribución de objetos llamado OMG, ha venido especificando en forma constante la arquitectura de software abierto, en el cual los componentes de objetos creados por distintos fabricantes pueden operar sobre distintas redes y sobre diferentes sistemas operativos. Actualmente son miembros 440 compañías de OMG y probablemente el software de objetos abiertos llegue a ser la fuente de todo el software medio Cliente/Servidor.

Según Gartner Group, una destacada firma de analistas, ha pronosticado que Windows NT o Windows 2000 será utilizado por el 95% de instituciones con más de 1000 empleados para el año 2000. Y esta cifra se prevé que llegará a un 99% para el año 2002.

A lo largo de los años, las aplicaciones empresariales más grandes han evolucionado de plataformas propietarias tipo AS/400 (con un muy alto costo de dispositivos) a plataforma Unix, y más recientemente están migrando hacia Windows NT por cuestión de costos, el desarrollo de aplicaciones para 10 o 14 versiones de UNIX es definitivamente mucho más costoso que el desarrollo enfocado en una sola plataforma.

	1994	1997	2000
interia	AS/400	AS/400	NT
jd edwards	AS/400	NT	NT
jba international	AS/400	NT, OS/2	NT
marcam	AS/400	NT, UNIX	NT
ssa	AS/400	NT	NT

La tabla anterior muestra una investigación efectuada por Gartner Group sobre cual es la plataforma primaria de desarrollo de empresas líderes en desarrollo de aplicaciones

empresariales. -Fuente: Resumen "Microsoft en la Administración pública", septiembre de 1998, ciudad de México.

También empresas como Oracle corp. , Informix y SAP han también dada a conocer que Windows NT Server es una plataforma primaria de sistema operativo para el desarrollo de sus sistemas.

CONCLUSIONES

Conocer cuales son sus requerimientos, las condiciones de trabajo necesarias, cuáles son las técnicas adecuadas recomendables para obtener una mejor eficiencia y menor costo así como la gama de herramientas de monitoreo, herramientas y ambientes de desarrollo que permiten el desarrollo de aplicaciones Cliente/Servidor sobre sistemas operativos de Microsoft de 32 bits será un requisito indispensable que debe de conocer fuertemente todo aquel profesional que se dedique al desarrollo. Y en el futuro seguramente seguirá siendo requisito indispensable.

Las redes de computadoras son actualmente una necesidad en cualquier organización. Como mencione las redes LAN son siempre el primer paso formal que dan las organizaciones pequeñas hacia una distribución total de sus recursos de cómputo y de información dentro sí mismas. Y esta es una tendencia que está en constante crecimiento. La necesidad de emplear aplicaciones Cliente/Servidor es un siguiente paso natural que realizan éstas organizaciones. Y esta necesidad es una solicitud que generalmente cae en primera instancia en el responsable o los responsables de sistemas y el cual debe de estar preparado para saber como actuar contando con las bases teóricas suficientes y posteriormente para decidir y determinar que personal interno o externo a la organización pueden de auxiliarlo.

Microsoft **NO** es la única compañía que desarrolla sistemas operativos, ni tampoco los sistemas operativos de 32 bits que construye Microsoft son los mejores que existen. Pero si son los más populares en la actualidad sobre todo en lo que respecta a PCs, generalmente por razones de mercadotecnia o por facilidad de uso los encontramos casi en cualquier parte. Si bien muchos usuarios destacan las deficiencias, sus innumerables parches y sus errores, que muchos otros usuarios viven en carne propia en los momentos más inoportunos; no pueden hacer más que resignarse y buscar una esperanza de solución en una gran lista de fallas en su sitio web; a través de su correspondiente parche o workaround como ellos llaman elegantemente a este último. (si es que lo hay) Esto debido a que no pueden tan fácilmente aislarse del resto de la organización adoptando un sistema operativo más tolerante a fallos y/o más caro pero incompatible con el resto de los equipos con los que tiene contacto. Sin embargo. Ahora existen alternativas nuevas por cierto que están dando mucho y bien de que hablar, tal es el caso del sistema operativo **LINUX** que entre sus más destacadas características es que es gratuito, se tiene acceso directamente al código fuente, es mucho más estable, robusto y eficiente que cualquier sistema operativo de Microsoft, aunque falta difundir mucho más su uso, sus ventajas así como también construir un juego suficientemente completo de aplicaciones, herramientas y servicios de soporte entre otros que lo hagan plenamente atractivo, **LINUX** tiene la ventaja de tener como aliado a la red de redes mundial, la Internet, donde puede a través de la contribución de diferentes programadores en todo el mundo gratuitamente - y otros no tanto - poder ir proporcionado estas características faltantes y entre ellas seguramente crecerá lo concerniente a la implantación y desarrollo de Cliente/Servidor sobre **LINUX**, además por supuesto de seguir distribuyendo el código fuente para que siga mejorándose por las contribuciones de diferentes personas que toman el código fuente. No es de dudarse que será un gran opción para los usuarios de PCs por la gran competencia que le dará a los sistemas operativos Windows en la próxima década.

Para finalizar pronóstico que los sistemas operativos Windows en sus siguientes evoluciones estén dominando plenamente el mercado de las redes LAN alrededor de unos seis años más, ya que están surgiendo innovadores sistemas operativos que hacen mejor administración de

los recursos de cómputo, como el caso arriba mencionado. Alguien dijo que "todo lo que empieza debe de terminar o evolucionar" y los sistemas operativos Windows como ahora los conocemos no serán la excepción a esta regla. Pero antes de que eso suceda estos estarán conviviendo dentro de la misma red con uno o más sistemas operativos muy similares o superiores a los sistemas operativos Windows pero gratuitos. Así que mientras eso aún no suceda los profesionales continuarán dando soporte técnico y desarrollando sobre todo aplicaciones Cliente/Servidor de todo tipo para Windows.

APÉNDICE

Este apéndice muestra como ejemplo una aplicación Cliente/Servidor junto con sus respectivos documentos de desarrollo.

Se Utiliza MS SQL Server ver 7.0 como motor de bases de datos y Visual Basic ver 5.0 como herramienta de desarrollo.

- Requerimientos la aplicación.
 - Definición de hardware o software que requerirá la construcción aplicación.
 - Definición de la interfaz gráfica de usuario.
 - Definición de los ambientes operativos de la aplicación.
 - Alcances y limitaciones que tendrá la aplicación.
 - Definición previa de hardware y software que requerirá la aplicación final.
- Inventario de recursos y requerimientos para el proyecto.
- Bitácora de avance.
- Registro del nombre y funciones de cada procedimiento.
- Registro de dependencias de código de la aplicación.
- Registro de la ubicación y plataforma de cada base de datos.
- Calendario de pruebas parciales.
- Registro de pruebas parciales.
- Registro de pruebas parciales en las plataformas proyectadas.
- Mantenimiento inicial de la aplicación.
- Puesta en Marcha. (liberación)

Requerimientos de la aplicación

- **Definición de hardware o software que requerirá la construcción aplicación.**

Se trata de un diseño de prueba o de muestra. Que involucra dos elementos solamente el cliente y el servidor

Características del hardware y software del servidor:

Hardware:

- Procesador: Intel Pentium III a 450MHz.
- Memoria RAM: 128 MB de RAM.
- Disco Duro: IDE mejorado 10 GB.
- Tarjeta de red: de 3com 10 Mbps. RJ45
- Otras características: puertos USB (hot docking), monitor de 17"
Unidad de CD 24x, FDD 3.5

Software:

- Sistema Operativo: MS Windows NT Server 4.0
- Service Pack 4.0 para Windos NT 4.0

- Navegador: Internet Explorer 4.0
- RDBMS: MS SQL Server 7.0
- Protocolos instalados por default: Netbeui, TCP/IP
- ODBC: SQL Server 3.70.06.23

Configuración:

- Nombre del servidor: Server1
- Tipo: PDC (Primary Domain Controller)
- Definición de dirección IP fija: Si
- Definición de grupo de trabajo: Seplan
- Nombre de la base de datos: Libros
- Nombre del cliente y del usuario definidos en User Manager For Domains

Características del hardware y software del cliente:

Hardware:

- Procesador: Intel Pentium 133 MHz.
- Memoria RAM: 32 MB de RAM.
- Disco Duro: IDE 1 GB.
- Tarjeta de red: de 3com 10 Mbps. RJ45
- Otras características: monitor de 17", unidad de CD 12x, FDD 3.5

Software:

- Sistema Operativo: MS Windows NT Workstation 4.0
- Service Pack 4.0 para Windos NT 4.0
- Navegador: Internet Explorer 4.0
- Herramienta de desarrollo: MS Visual Basic 5.0
- Protocolos instalados por default: Netbeui, TCP/IP
- ODBC: SQL Server 3.0.23.1

Configuración:

- Nombre de la computadora cliente: Lázaro
- Definición de dirección IP fija: Si
- Definición de grupo de trabajo: Seplan

• **Definición de la interfaz gráfica de usuario.**

Definir el GUI de la aplicación es determinar los comandos y botones básicos con los que contara el GUI, así como vista y colores. Esto es considerado en ocasiones una aproximación ya que puede llegar a cambiar durante el desarrollo de la Aplicación y se actualizará o actualizarán antes de liberarla o de hacerla más funcional.

GUI para Altas

Altas (Productos) [Salir] [Ayuda]

Salir Ayuda

Escribe la clave de alto

Buscar

Cve. alt: 001 00 0000

Nombre Artículo

Modelo

Marca

Proveedor

Telefono

Direccion

Ciudad

CP

Cancelar Ejecuta Alta

GUI para Bajas

Bajas (Productos) [Salir] [Ayuda]

Salir Ayuda

Escribe la clave de bajo

Buscar

Cve. alt

Nombre Artículo

Modelo

Marca

Proveedor

Telefono

Direccion

Ciudad

CP

Cancelar Ejecuta Baja

GUI para Modificaciones

Modificaciones (Productos) [Salir] [Ayuda]

Salir Ayuda

Escribe el ambito de la modificacion

Buscar

Cve. alt

Nombre Artículo

Modelo

Marca

Proveedor

Telefono

Direccion

Ciudad

CP

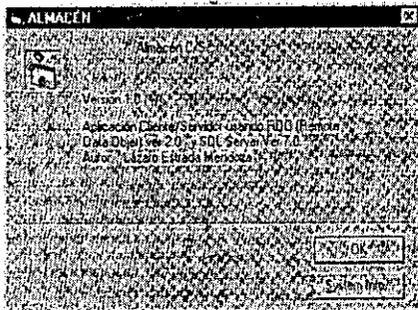
Cancelar Ejecuta Modificacion

GUI para Consultas

Código	Nombre	Modelo	Marca	Proveedor
177-77-7700	Monitor	AceView	Acer	AMD
177-77-7701	Mouse	T220	Generico	JGB
177-77-7703	NIC	Etherlink III	3COM	3COM
901-00-0006	Socinat	SK-II	BTC	JGB
901-00-0008	CD Rom	Aston 52x	Aston	JGB
901-00-0009	Teclado	Pass II	Generico	JGB
901-00-0010	Regulador	Reviva 500W	Vica	Vica
901-00-0012	FaxModem	Mot 28.8	Motorola	AMD
901-00-0013	Mouse	534	Acer	AMD
901-00-0014	NoBreak	IA-600W	APL	AMD

Buttons: Cancelar, Iniciar/Actualizar/Manejables

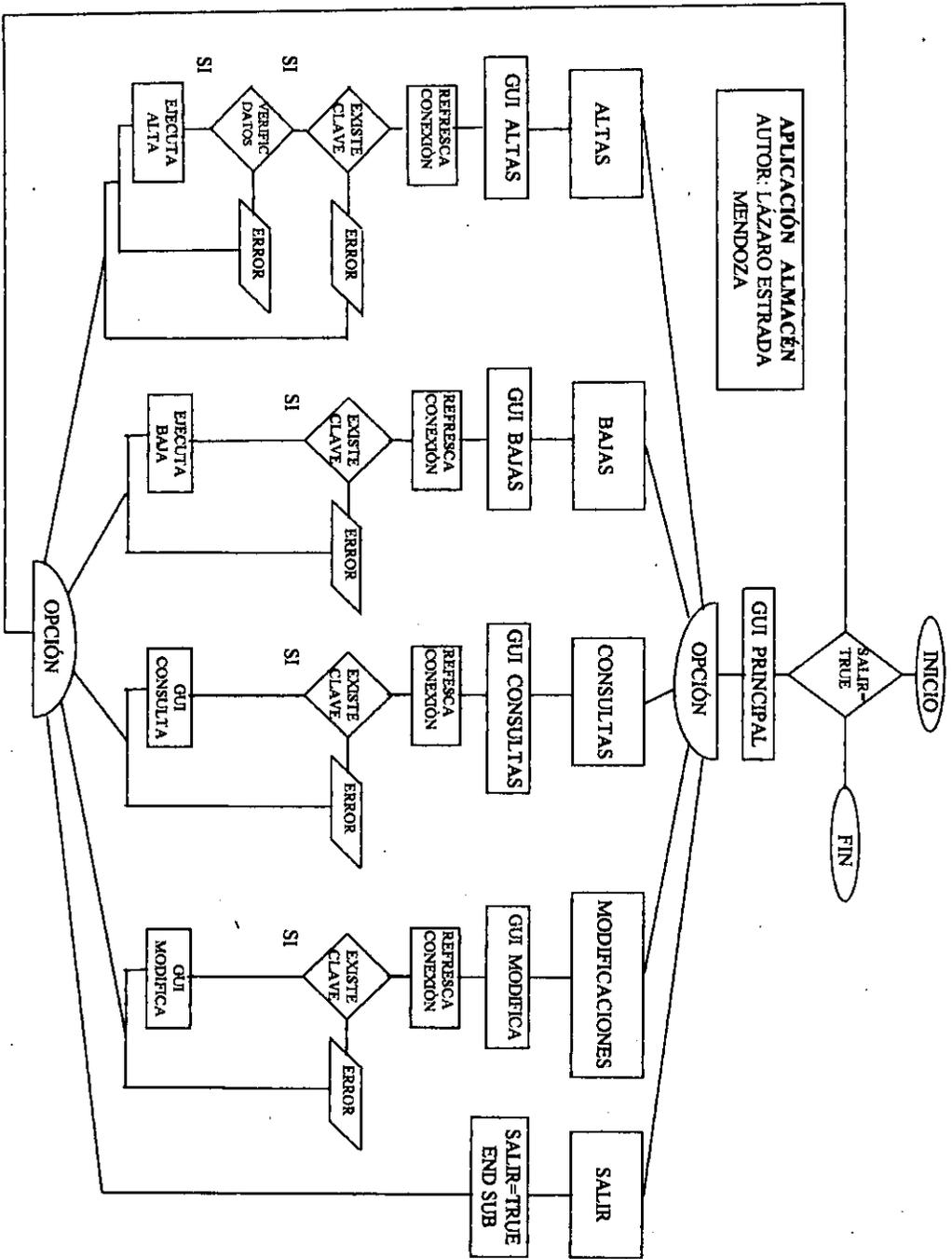
GUI para Ayuda e información de la Aplicación



■ Definición de los ambientes operativos de la aplicación.

Se hace una definición rígida de los sistemas operativos en los que trabajará adecuadamente la aplicación.

- En el servidor es Windows NT Server 4.0 o posterior.
- En el cliente se recomienda trabajar preferentemente sobre Windows NT workstation 4.0 aunque puede ser utilizado Windows 95 o Windows 98.



- **Alcances y limitaciones que tendrá la aplicación.**

La aplicación utiliza los componentes RDO y RDC a través de ODBC lo que permite que esta se conecta a otros motores de bases de datos solamente definiendo o trasladando la bases de datos a este. Esta limitado a plataformas Microsoft y servidores de Bases de Datos SQL server postiores a la version 6.0.

- **Definición previa de hardware y software que requerirá la aplicación final**

En este caso los requerimientos serán los mismos que para la construcción de la aplicación

Inventario de recursos y requerimientos para el proyecto.

Debido a que esta es una aplicación de muestra esa sección no se desarrolla

Bitácora de avance.

Ago/99

Semana 2: Preparación y entrega de la solicitud de presupuesto para el desarrollo del Proyecto.

Semana 3: Rechazo de la de la solicitud de presupuesto

Semana 4: Replanteamiento de la solicitud de presupuesto, solicitando movimientos de partidas para reunir el presupuesto del proyecto.

Nov/99

Semana 1: Autorización del presupuesto para el desarrollo del proyecto.

Semana 2: Solicitud de equipo

Semana 3: Adquisición de equipo de cómputo necesario para el proyecto.

Semana 4: Adquisición de software necesario.

Dic/99

Semana 1: Inicia la Reclutación de personal.

Semana 1: Se improvisa una red de dos computadoras.

Semana 2: Se instala y se configura SO en el servidor, se configura y se instala SQL server versión 7, se instala Service Pack 4.0.

Semana 3: Se instala y se configura el SO del cliente, se instala y configura la herramienta de Desarrollo.

Semana 3: Se prueba los enlaces básicos del cliente con el servidor, se instala y configura ODBC.

Ene/00

Semana 1: Se crea una aplicación de prueba para verificar el correcto enlace a nivel de ODBC.

Semana 2: Finaliza la reclutación de personal para el proyecto

Calendario de pruebas en los desarrollos parciales.

Feb/00

Semana 1: Diseño y Prueba del modulo altas.

Semana 1: Diseño y Prueba del modulo bajas.

Semana 2: Diseño y Prueba del modulo modificación.

Semana 2: Diseño y Prueba del modulo consulta.

Registro del nombre y funciones de cada procedimiento.

Todas las funciones y procedimientos tienen como campo llave clave del Producto

ALTAS.- Realiza adiciones a la tabla llamada Producto.

BAJAS.- Realiza eliminaciones de registros de la tabla producto.

CAMBIOS.- Realiza modificaciones a la tabla producto.

CONSULTAS.- Realiza desplegados de información con la opción de poder realizar desde aquí Mantenimiento a la tabla productos.

Registro de dependencias de código de la aplicación.

La fundamental dependencia de código la podemos solamente encontrar en en la Opción "Conectar al servidor de Bases de Datos RDBMS" que realiza la conexión al Motor de bases de datos SQL ver 7.0 en la forma principal del ejemplo y en cada modulo esta conexión es regenerada.

Registro de la ubicación y plataforma de cada base de datos.

Ubicación de la base datos Soluciones en el servidor: Acer

Plataforma: MS SQL Server Ver. 7.0, sobre Windows NT Server 4.0

Estructura de la tabla "producto"

Column Name	Datatype	Length	Precision	Scale	Allow Nulls
Cve_art	varchar	11	0	0	<input type="checkbox"/>
Nombre	varchar	15	0	0	<input checked="" type="checkbox"/>
Modelo	varchar	15	0	0	<input checked="" type="checkbox"/>
Marca	varchar	20	0	0	<input checked="" type="checkbox"/>
Proveedor	varchar	20	0	0	<input checked="" type="checkbox"/>
telefono	char	10	0	0	<input checked="" type="checkbox"/>
Direccion	varchar	35	0	0	<input checked="" type="checkbox"/>
Ciudad	varchar	15	0	0	<input checked="" type="checkbox"/>
CP	char	10	0	0	<input checked="" type="checkbox"/>

Registro de pruebas parciales.

Se realizan pruebas a nivel de conexión y procesos commit con la tabla "producto" de la base de datos "soluciones" . Con cada una de las interfaces de la aplicación ... Resultando satisfactorias.

Registro de pruebas parciales en las plataformas proyectadas.

Pruebas en clientes con Window95

- No se registraron problemas relevantes en la configuración del ODBC ver 3.0
- Sobre Windows95 se ejecuto la aplicación corriendo cada uno de sus modulos satisfactoriamente.

Pruebas para clientes Windows NT WorkStation

- En Windows NT workstation ver 4.0
- Es necesario primero instalar Service Pack 4 o 5
- Requiere de tiempo para configurar el controlador de ODBC original.. Es necesario instalar una versión más actualizada, que esta disponible en la página www de Microsoft.
- Se efecuto la ejecución con satisfacción de todos los modulos de la aplicación

Mantenimiento inicial de la aplicación.

En la aplicación de ejemplo por ser pequeña solo nos limitamos a labores de instalación.

Corrección de las fallas detectadas en la fase de pruebas.

Puesta en Marcha. (liberación)

Instalación de la aplicación ya sea a través del personal de desarrollo o por medios de programas wizards distribuidos en discos flexibles.

Presentación del sistema. Se trata de una aplicación cliente servidor tipo [PA]~[D]. Para el control que productos que entran y salen de un almacén.

- **Personal que participo en las diferentes etapas de programación.**
Lázaro Estrada Mendoza
- **Planificación provisional y real de la programación.**
 1. Realización y comprobación del modelo Conexión
 2. Realizacion y comprobación del modelo de Altas.
 3. Realizacion y comprobación del modelo de Bajas.
 4. Realizacion y comprobación del modelo de Modifica.
 5. Realizacion y comprobación del modelo de Consulta.

- **Archivos utilizados por la aplicación.**

FrmPrincipal, frmAltas, frmBajas, frmConsultan, frmModifica, frmAbout, frmLogin

- **Localización de las bases de datos, tipos de DBMSs y versiones utilizados**

Nombre de la base de datos: Soluciones

Ubicación: Microsoft SQL Servers, SQL Server Group, ACER (Windows NT), Databases, Soluciones.

Nota: En el diseño de la tabla "producto" se contemplo colocar "(UNKNOWN)" en los campos que permitieran la introducción de NULL, esto para evitar el "run Time 64" en el modulo de consultas y en el modulo de modificaciones.

- **Configuración actual de la red (topología, fallas actuales, nombres de maquinas etc.)**

La aplicación fue desarrollada y probada en una red aislada para evitar problemas de tráfico y de fallas de red.

Topología: Ethernet, velocidad 10Mbps, cableado UTP nivel 4.

Tajetas de red: Marca 3com Eterlink III. 10 baseT

Fallas actuales: Ninguna.

Nombres de los equipos: Acer (servidor) y Lazaro (cliente).

- **Documentar las agrupación de rutinas en unidades funcionales.**

Altas, Bajas, Consultas Cambios y Ayudas

- **Documentar la ubicación de almacenamiento de los subprogramas y secuencias de instrucciones.**

- **Documentar las variables utilizadas.**

-

Cajas de texto: Inician con "TXT" +"nombre de la caja"+"Modulo del que depende"

Botones de comando: Inician con "Cmd"+"nombre de la rutina"+"Modulo del que depende".

Frames: Inician con el prefijo "Frame"+"Nombre del frame" +"Modulo del que depende".

Formas: Inician con el prefijo "Frm"+"Nombre del frame" +"Modulo del que depende".

- **Realizar una lista de los programas compilados.**
- **Realizar una lista de la ejecución de los programas.**
- **Incluir el o los diagramas de flujo de la aplicación.**

Código Fuente de la Aplicación de Ejemplo

Modulo Principal

VERSION 5.00

Begin VB.Form frmPrincipal

```

Caption          = "PRODUCTOS ( GUI Para Almacen)"
ClientHeight    = 5400
ClientLeft      = 165
ClientTop       = 735
ClientWidth     = 6960
LinkTopic       = "Form1"
ScaleHeight     = 5400
ScaleWidth      = 6960
StartupPosition = 3 'Windows Default
Begin VB.CommandButton Command1
    Caption      = "Command1"
    Height       = 375
    Left         = 1920
    TabIndex     = 5
    Top          = 840
    Width        = 1215
End
Begin VB.Frame Framecmd
    Height       = 3975
    Left         = 5520
    TabIndex     = 0
    Top          = 360
    Width        = 1335
    Begin VB.CommandButton cmdModifica
        Caption   = "Modifica"
        Height    = 495
        Left      = 240
        TabIndex  = 4
        Top       = 3000
        Width     = 975
    End
    Begin VB.CommandButton cmdConsulta
        Caption   = "Consulta"
        Height    = 495
        Left      = 240
        TabIndex  = 3
        Top       = 2160
        Width     = 975
    End
    Begin VB.CommandButton cmdBajas
        Caption   = "Bajas"
        Height    = 495
        Left      = 240
        TabIndex  = 2
        Top       = 1320
        Width     = 975
    End
    Begin VB.CommandButton cmdaltas
        Caption   = "Altas"
        Height    = 495
        Left      = 240
        TabIndex  = 1
        Top       = 480
        Width     = 975
    End
End
Begin VB.Menu menuConectarop1
    Caption      = "&Conectar"

```

```

Begin VB.Menu menuConectar
  Caption      = "Conectar al Servidor de base de Datos (RDBMS)"
End
Begin VB.Menu menuConectarop2
  Caption      = "Desconectar al Servidor de base de Datos
(RDBMS)"
End
End
Begin VB.Menu menuayuda
  Caption      = "&Ayuda"
  Begin VB.Menu menuayuda_Op1
    Caption     = "Ayuda "
  End
  Begin VB.Menu menuayudaop2
    Caption     = "Acerca de..."
  End
End
End
Begin VB.Menu k
  Caption      = ""
End
Begin VB.Menu b
  Caption      = ""
End
Begin VB.Menu a
  Caption      = ""
End
Begin VB.Menu d
  Caption      = ""
End
Begin VB.Menu e
  Caption      = ""
End
Begin VB.Menu j
  Caption      = ""
End
Begin VB.Menu i
  Caption      = ""
End
Begin VB.Menu h
  Caption      = ""
End
Begin VB.Menu g
  Caption      = ""
End
Begin VB.Menu f
  Caption      = ""
End
Begin VB.Menu menuSalirPrincipal
  Caption      = "S&alir"
End
End
Attribute VB_Name = "frmPrincipal"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

```

```
Public cn As rdoConnection
Public rs As rdoResultset
Public strConnect As String
```

```
Private Sub cmdAltas_Click()
Load frmAltas
frmAltas.Show
End Sub
```

```
Private Sub cmdBajas_Click()
Load frmBajas
frmBajas.Show
End Sub
```

```
Private Sub cmdConsulta_Click()
Load FrmConsulta
FrmConsulta.Show
End Sub
```

```
Private Sub cmdModifica_Click()
Load frmModifica
frmModifica.Show
End Sub
```

```
Private Sub Command1_Click()

Set cn = New rdoConnection
strConnect = "Driver={SQL
Server};Server=SERVER;Database=soluciones;Uid=lazaro;Pwd=;"
With cn
.CursorDriver = rdUseServer
.Connect = strConnect
.EstablishConnection
End With
End Sub
```

```
Private Sub menuayudaop2_Click()
Load frmAbout
frmAbout.Show
End Sub
```

```
Private Sub menuConectar_Click()
Load frmLogin
frmLogin.Show
End Sub
```

```
Private Sub menuSalirPrincipal_Click()
Unload frmPrincipal
```

End Sub

Modulo Altas

VERSION 5.00

Begin VB.Form frmAltas

 Caption = "Altas (Productos)"
 ClientHeight = 6225
 ClientLeft = 60
 ClientTop = 630
 ClientWidth = 5400
 LinkTopic = "Form1"
 ScaleHeight = 6225
 ScaleWidth = 5400
 StartPosition = 3 'Windows Default

Begin VB.CommandButton cmdCancelA

 Caption = "Cancelar"
 Height = 495
 Left = 360
 TabIndex = 23

Begin VB.Menu menuAyudaAltas

 Caption = "A&yuda"

End

End

Attribute VB_Name = "frmAltas"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

 Public cn As rdoConnection

 Public rs As rdoResultset

Public strConnect As String

Private Sub cmdCancelA_Click()

 Unload frmAltas

End Sub

 Private Sub Command1_Click()

 'Dim StrConnect As String
 Dim cadena As String
 Dim errorCuerda As String
 Dim respuesta As Integer
 Dim titlebox As String
 Dim txtcve As String
 Dim NumDerror As Integer
 Dim Msg
 Dim Msgcve As String
 Dim txtcveA, txtNombreA, txtModeloA, txtMarcaA As String

```
Dim txtProveedorA, txttelefonoA, txtDireccionA, txtCiudadA, txtCPA As  
String
```

```
Set cn = New rdoConnection  
strConnect = "Driver={SQL  
Server};Server=SERVER;Database=soluciones;Uid=lazaro;Pwd="'  
With cn  
    .CursorDriver = rdUseServer  
    .Connect = strConnect  
    .EstablishConnection  
End With
```

```
txtcveA = "" + (txtcve_art1 + "-" + txtcve_art2 + "-" + txtcve_art3) + ""  
txtNombreA = "" + txtNombre + ""  
txtModeloA = "" + txtModelo + ""  
txtMarcaA = "" + txtMarca + ""  
txtProveedorA = "" + txtProveedor + ""  
txttelefonoA = "" + txtTelefono + ""  
txtDireccionA = "" + txtDireccion + ""  
txtCiudadA = "" + txtCiudad + ""  
txtCPA = "" + txtCP + ""
```

```
If (txtcve_art1 = "") Or (txtcve_art2 = "") Or (txtcve_art3 = "") Then  
    Msgcve = " El campo para clave de articulo no puede ser vacio"  
    MsgBox Msgcve, vbYes, " A T E N C I O N "
```

```
Else
```

```
    On Error Resume Next
```

```
    cadena = "insert into producto (cve_art,Nombre,  
    modelo,Marca,Proveedor,telefono,direccion,ciudad,cp) values ("'  
        & txtcveA + "," + txtNombreA + "," + txtModeloA + ","'  
        & txtMarcaA + "," + txtProveedorA + "," + txttelefonoA + ","'  
        & txtDireccionA + "," + txtCiudadA + "," + txtCPA + ")"
```

```
    Set rs = cn.OpenResultset(cadena, 2, rdConcurValues)
```

```
If Err.Number <> 0 Then
```

```
    Msg = "Error # " & Str(Err.Number) & " "  
        & "El producto no pudo ser dado de alta debido a que no  
concuerdan los tipos de los campos o porque ya existe la clave"
```

```
    ' & Err.Source & Chr(13) & Err.Description
```

```
    MsgBox Msg, , " A T E N C I O N - Error", Err.HelpFile, Err.HelpContext
```

```
End If
```

```
End If
```

```
End Sub
```

```
Function TipErrorw()
```

```
TipError:
```

```
    errorCuerda = "El producto no pudo ser dado de alta debido a que no  
concuerdan los tipos o porque ya existe la clave"
```

```
    titlebox = " A T E N C I O N "
```

```
    respuesta = MsgBox(errorCuerda, vbYes, titlebox)
```

```
End Function
```

Modulo Bajas

VERSION 5.00

Begin VB.Form frmBajas

```
Caption      = "Bajas (Productos)"
ClientHeight = 6525
ClientLeft   = 60
ClientTop    = 630
ClientWidth  = 5595
LinkTopic    = "Form1"
ScaleHeight  = 6525
ScaleWidth   = 5595
```

```
Height      = 255
Left        = 360
TabIndex    = 11
Top         = 360
Width       = 975
```

End

Begin VB.Label Label2

```
Caption      = "Nombre Artículo"
Height       = 375
Left         = 120
TabIndex     = 10
Top          = 960
Width        = 1335
```

Private Sub Command2_Click()

Dim cadenaborra As String

Dim textoBorra As String

Dim i As Integer

Dim varvalue As Variant

Dim renglon As Integer

Dim Encontrado As Boolean

Dim Msg As String

'Restablece Resultset para esta forma

```
Set rs = cn.OpenResultset("Select * from producto", rdOpenDynamic,
rdConcurValues)
```

'inicializa cajas de texto para Campos

txtcve_artBaj = ""

txtNombreBaj = ""

txtModeloBaj = ""

txtMarcaBaj = ""

txtProveedorBaj = ""

txtTelefonoBaj = ""

txtDireccionBaj = ""

txtCiudadBaj = ""

txtCPBaj = ""

'Deshabilita el boton de ejecución de la modificación

Do Until rs.EOF

For i = 0 To rs.rdoColumns.Count - 1

varvalue = rs.rdoColumns(i).Value

If txtBorracve = varvalue Then

```

        txtcve_artBaj = rs(0)
        txtNombreBaj = rs(1)
        txtModeloBaj = rs(2)
        txtMarcaBaj = rs(3)
        txtProveedorBaj = rs(4)
        txtTelefonoBaj = rs(5)
        txtDireccionBaj = rs(6)
        txtCiudadBaj = rs(7)
        txtCPBaj = rs(8)
        Encontrado = True
        Exit Do

    End If

Next i
rs.MoveNext
Loop

    If Encontrado = False Then
        Msg = " Registro NO encontrado " & Chr(13) &
            " Verifique Mayusculas/minúsculas del Teclado"
        MsgBox Msg, vbYes, " - A T E N C I O N - "

    End If

'Stop

If Encontrado Then
    errorCuerda = "Observe la información del Registro... ¿Desea Borrar este
Registro?"
    titlebox = "                -A T E N C I O N- "
    respuesta = MsgBox(errorCuerda, vbYesNo, titlebox)

    If respuesta = 6 Then

        textoBorra = "'" + txtBorracve + "'"
        cadenaborra = "delete from producto where cve_art=" + textoBorra
        'Print cadenaborra
        Set rs = cn.OpenResultset(cadenaborra, 2, rdConcurValues)
        End If
    End Sub

Private Sub Form_Load()
Set cn = New rdoConnection
    strConnect = "Driver={SQL
Server};Server=SERVER;Database=soluciones;Uid=lazaro;Pwd=;"
    With cn
        .CursorDriver = rdUseServer
        .Connect = strConnect
        .EstablishConnection
    End With
End Sub

Private Sub menuSalirBajas_Click()
    Unload frmBajas
End Sub

```

Modulo Modifica

```
VERSION 5.00
Begin VB.Form frmModifica
    Caption           = "Modificaciones (Productos)"
    ClientHeight      = 6630
    ClientLeft        = 60
    ClientTop         = 630
    ClientWidth       = 5925
    LinkTopic         = "Form1"
    ScaleHeight       = 6630
    ScaleWidth        = 5925
    StartupPosition   = 3 'Windows Default
    Begin VB.CommandButton cmdCancel
        Caption        = "Cancelar"
        Height         = 375
        Left           = 960
        TabIndex       = 23
        Top            = 6240
        Width          = 1215
    End
End

Begin VB.Menu menuSalirModifica
    Caption           = "S&alir"
End
Begin VB.Menu menuAyudaModifica
    Caption           = "A&yuda"
End
End
Attribute VB_Name = "frmModifica"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
    Public cn As rdoConnection
    Public rs As rdoResultset
    Public strConnect As String

Public Sub CmdBusca_Click()
    Dim i As Integer
    Dim varvalue As Variant
    Dim renglon As Integer
    Dim Encontrado As Boolean
    Dim Msg As String
    'Restablece Resultset para esta forma

    Set rs = cn.OpenResultset("Select * from producto", rdOpenDynamic,
rdConcurValues)

'inicializa cajas de texto para Campos
    txtcve_artMod = ""
    txtNombreMod = ""
    txtModeloMod = ""
```

```

        txtMarcaMod = ""
        txtProveedorMod = ""
        txtTelefonoMod = ""
        txtDireccionMod = ""
        txtCiudadMod = ""
        txtCPMod = ""
'Deshabilita el boton de ejecución de la modificación
cmdModifica.Enabled = False
'Stop
Do Until rs.EOF
    For i = 0 To rs.rdcColumns.Count - 1
        varvalue = rs.rdcColumns(i).Value

        If txtBusca = varvalue Then
            txtcve_artMod = rs(0)
            txtNombreMod = rs(1)
            txtModeloMod = rs(2)
            txtMarcaMod = rs(3)
            txtProveedorMod = rs(4)
            txtTelefonoMod = rs(5)
            txtDireccionMod = rs(6)
            txtCiudadMod = rs(7)
            txtCPMod = rs(8)
            Encontrado = True
            Exit Do

        End If

    Next i
    rs.MoveNext
Loop
    If Encontrado = False Then
        Msg = " Registro NO encontrado " & Chr(13) &
            " Verifique Mayusculas/minúsculas del Teclado"
        MsgBox Msg, vbYes, " - A T E N C I O N - "
    Else
        cmdModifica.Enabled = True
    End If

End Sub

Private Sub cmdCancel_Click()
    Unload frmModifica
End Sub

Private Sub cmdModifica_Click()

    rs.Edit

    rs(0) = txtcve_artMod
    rs(1) = txtNombreMod
    rs(2) = txtModeloMod
    rs(3) = txtMarcaMod
    rs(4) = txtProveedorMod
    rs(5) = txtTelefonoMod
    rs(6) = txtDireccionMod
    rs(7) = txtCiudadMod

```

```

        rs(8) = txtCPMod
        On Error Resume Next
        rs.Update
'On Error Resume Next
If Err.Number <> 0 Then
    Msg = "Error # " & Str(Err.Number) & " "
        & "El producto no pudo ser MODIFICADO debido a que no concuerda
el tipo de campo para la clave"
        '& Err.Source & Chr(13) & Err.Description
    MsgBox Msg, , " A T E N C I O N - Error", Err.HelpFile, Err.HelpContext
End If

End Sub

Private Sub Command2_Click()

End Sub

Private Sub Form_Load()
Set cn = New rdoConnection
    strConnect = "Driver={SQL
Server};Server=SERVER;Database=soluciones;Uid=lazaro;Pwd=;"
    With cn
        .CursorDriver = rdUseServer
        .Connect = strConnect
        .EstablishConnection
    End With

Private Sub menuSalirModifica_Click()
    Unload frmModifica
End Sub

```

Modulo Consultas

```

MaxRows          = 0
ErrorThreshold   = -1
BatchSize        = 15
BackColor        = -2147483643
ForeColor        = -2147483640
Enabled          = -1 'True
ReadOnly         = 0 'False
Appearance       = -1 'True
DataSourceName   = "soluciones"
RecordSource     = "Select * from producto"
UserName         = ""
Password         = ""
Connect          = ""
LogMessages      = ""
Caption          = "MSRDC1"
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

```

```

        Name           = "MS Sans Serif"
        Size           = 8.25
        Charset        = 0
        Weight         = 400
        Underline      = 0 'False
        Italic         = 0 'False
        Strikethrough  = 0 'False
    EndProperty
End
Begin VB.Menu menuSalirConsulta
    Caption           = "S&alir"
End
Begin VB.Menu menuAyudaConsulta
    Caption           = "A&yuda Mantenimiento"
End
End
Attribute VB_Name = "FrmConsulta"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub cmdCancelar_Click()
Unload FrmConsulta
End Sub

Private Sub CmdHabilita_Click()
cmdHabilita.Caption = "Habilitado"
DBgridConsulta.AllowAddNew = True
DBgridConsulta.AllowArrows = True
DBgridConsulta.AllowDelete = True
DBgridConsulta.AllowUpdate = True
menuAyudaConsulta.Enabled = True

End Sub

Private Sub Form_Load()
menuAyudaConsulta.Enabled = False
End Sub

Private Sub menuAyudaConsulta_Click()
MsgBox ("Una vez habilitado el mantenimiento:" _
& Chr(13) & " Utilice el * para Agregar un nuevo regsitro. " _
& "Marque un reglon(registrto) con el Boton izquierdo " _
& "del ratón y Utilice la tecla <DEL> para suprimir todo el registro." _
& "Para Modificar coloque en el campo deseado realice el cambio y cambie "

& "el cursor a otra posicion con el ratón o el teclado para acentuar la
modificación")
End Sub

Private Sub menuSalirConsulta_Click()
Unload FrmConsulta
End Sub

```

GLOSARIO

ActiveX.- Lenguaje desarrollado por Microsoft para la elaboración de aplicaciones exportables a la red y capaces de operar sobre cualquier plataforma a través, normalmente, de navegadores WWW entre otras ventajas, permite dar dinamismo a las páginas web.

Adaptador.- Es un elemento por lo general físico que permite acondicionar otro para su óptimo uso.

Algoritmo.- Es una secuencia de pasos lógicos a seguir en la solución de un problema.

Aplicación.- Programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Suele ser una solución informática para la automatización de ciertas tareas complicadas como por ejemplo la contabilidad o la gestión de un almacén, generalmente ofrecen gran potencia ya que son diseñadas exclusivamente para resolver un problema específico.

API.- Siglas de "Interfaz de Programación de Aplicaciones". Es un conjunto de rutinas que utiliza una aplicación para solicitar o efectuar servicios de nivel inferior ejecutados por el sistema operativo.

APIs.- Plural de API

Arquitectura Cliente/Servidor.- Es la combinación adecuada de software y hardware para el proceso distribuido, en el que la comunicación se establece de uno a varios, y en el que una máquina puede ser tanto servidor como cliente.

ATM.- Siglas de "Asynchronous Transfer Mode". Tecnología de punta de transmisión y conmutación a grandes velocidades, por medio del movimiento asíncrono de paquetes con rapidez. Opera en los niveles 1 y 2 del modelo OSI. También se conoce como cell relay, se considera como siguiente generación de arquitectura de redes.

Base de Datos.- Es un juego de registros interrelacionados almacenados sobre un dispositivo que está diseñado para permitir acceso de múltiples aplicaciones. Una base de datos debe de tener mínima redundancia de datos y debe de permitir el cambio y el crecimiento de esta.

Benchmark.- Comúnmente se utiliza este término cuando se habla de puntos de referencia para medir el rendimiento.

Bloquear.- Acción que realiza una aplicación sobre una base de datos para mantener exclusividad en el acceso a la base de datos, a la tabla o al registro. Esta acción se realiza cuando se pretende escribir o modificar información.

Browser.- El Browser es una aplicación que despliega una ventana en la cual el usuario puede escoger de una lista de elementos.

Buffer.- Area de almacenamiento que conserva información temporalmente, los buffers en una aplicación, son áreas reservadas que reciben información y la conservan para su procesamiento.

CASE.- Siglas de "Computer Aided Software Engineering". La utilización de software para ayudar en la elaboración, designación, documentación y algunas otras áreas de desarrollo de programas.

Capa de Transporte de una red LAN.- Es la capa numero cuatro de referencia OSI. Es la responsable de la comunicación confiable entre los nodos de una red. Realiza los controles de flujo y de errores usando generalmente circuitos virtuales para asegurar entrega confiable de datos.

Clase.- En Programación Orientada a Objetos es una categoría generalizada que describe a un grupo específico de elementos llamados objetos y que pueden existir dentro de este. Una clase es una herramienta descriptiva usada dentro de un programa para definir un conjunto de atributos o un conjunto de servicios (acciones disponibles para otras partes del programa).

Código proxy.- Apoderar. Entidad que por motivos de eficiencia, esencialmente ocupa el lugar de otra. Objeto específico de la interfaz que empaqueta parámetros para esa interfaz con el fin de prepararlo para una llamada remota. Un proxy se ejecuta en el espacio de direcciones del emisor y se comunica con una etiqueta correspondiente en el espacio de direcciones del receptor.

Componente.- Cualquier software compatible con Automatización (anteriormente denominada Automatización OLE), que significa que puede utilizarse mediante programa en una solución personalizada. Esto incluye controles ActiveX (archivos .ocx), documentos ActiveX y componentes de código ActiveX.

Componente ActiveX.- Archivo físico (por ejemplo, .exe, .dll u .ocx) que contiene clases, que son definiciones de objetos. Puede usar estos objetos en las aplicaciones que escriba con Visual Basic. Vea también términos de ActiveX.

Componente de código.- Archivo .exe o .dll que proporciona objetos creados a partir de una de las clases proporcionadas por el componente. Antes se denominaba servidor OLE y servidor de Automatización OLE.

Compilar.- Es el proceso por el cual se detectan los errores de sintaxis y semántica en un programa fuente, este proceso no genera código ejecutable por la computadora (código objeto) hasta que el código fuente esta libre de todo error.

Commit.- Es la escritura física al disco de los datos, o registros de la base de datos que se encuentran en memoria.

Conectividad.- Estado que permite la transferencia de señales eléctricas desde un origen hasta un destino.

Conmutador.- Es una técnica de telecomunicaciones que dinámicamente establece una conexión física antes de un intercambio de información, presentando la conexión siguiendo el intercambio.

Controlador.- Al realizar una operación, un controlador proporciona la capacidad de direccionamiento a estructuras creadas por las APIs de ODBC usadas para almacenar parámetros, errores y argumentos devueltos. Con la API de ODBC puede tener controladores de entorno, de conexión y de instrucción.

Correlación.- relación recíproca o mutua entre dos o más cosas.

CSMA/CD.- Siglas de "Carrier Sense Multiple Access/Collision Detection" Técnica de algoritmo que es utilizada para enviar señales dentro de una red local. Cuando una tarjeta de red detecta solo la portadora, empieza a transmitir, pero debe seguir escuchando por si ocurre alguna colisión. De ser así, requiere hacer una retransmisión

CSA.- Siglas de "Cliente/Servidor Architecture". Arquitectura Cliente/Servidor. Modelo de trabajo donde el proceso se distribuye entre dos entidades: la que solicita(cliente) y la que da el servicio (servidor).

Cursor.- Hace un seguimiento de la posición del controlador dentro del conjunto de resultados. El cursor se denomina así porque indica la posición actual en el conjunto de resultados, al igual que el cursor en una pantalla indica la posición actual. Los cursores permiten al usuario desplazarse y actualizar un conjunto de resultados con menos limitaciones que en el modo examinar.

C/S.- Abreviatura de Cliente/Servidor.

Datos distribuidos.- Un conjunto de información que se encuentra dispersa en un conjunto de depósitos de datos que a través de mecanismos Cliente/Servidor aparecen ante el usuario como un único conjunto.

DB2.- Manejador de bases de datos de IBM para ambientes MVS (mainframe). Utiliza SQL, y define en sí mismo un dialecto estándar.

DBMS.- Siglas de "Data Base Manager System". Es un mecanismo que permite el acceso transparente a los datos, distribución de los mismos y control de integridad.

Diagrama de Flujo.- Es la representación gráfica del flujo de un programa de computadora.

De Facto.- Estándar del mercado, de la industria, resultado del uso común.

Debugger.- Ver depurador.

De Jure.- Estándar de documento, resultado del estudio de un grupo de expertos con validez nacional o internacional.

Depurador.- Técnica que permite detectar los errores de una aplicación cuando esta se ejecuta, se auxilia de herramientas de software propias para ello, que por lo regular forman parte del juego de herramientas de las herramientas de desarrollo.

Dominio.- Es una declaración abstracta que se realiza a un servidor Windows NT en su calidad de PDC (Primary Domain Controller) o BDC (Backup Domain Controller), para agrupar a un conjunto de equipos workstations dentro de una red.

Driver.- O controlador. Es un conjunto de rutinas de software que se utilizan para controlar el intercambio de información entre un dispositivo y el CPU.

Escritorio.- Es el área de la pantalla del monitor, donde se encuentran los iconos de acceso directo de las aplicaciones, se dice escritorio por que es similar a un escritorio físico visto desde arriba, donde los iconos pasan ser los documentos o accesorios que se tienen encima de este.

Estándar.- Especificación que debe utilizar un sistema para cumplir con las características que exige el mercado si es que se quiere ser compatible y lograr la comunicación. Existen estándares DE JURE y DE FACTO.

Escalabilidad.- Es la cualidad que posee un elemento de hardware o de software para poder ser mejorado, enriquecido o actualizado mediante la adición o reemplazo de elementos nuevos de hardware o software, con el propósito de desempeñar funciones superiores a las que venía desempeñando.

Ethernet.- Es el estándar de tarjetas de red más conocido y sólido. Define una velocidad de transmisión utilizando de 10Mbps/segundo, utilizando el la técnica CSMA/CD.

FDDI.- Siglas de "Fiber Distributed Data Interface". El estándar para transmisión de datos en redes locales utilizando fibra óptica, a una velocidad de 100 mbps. Utiliza un doble anillo en una topología similar a Token-Ring, incluso en la definición del frame. Igualmente utiliza un protocolo de Token-Passing para el control de la RED.

Fiabilidad.- Probabilidad del perfecto funcionamiento de un dispositivo en determinadas circunstancias y durante un cierto período de tiempo.

Groupware.- Término genérico con el cual se define el software cuyo principal objetivo es automatizar la relación, entre un grupos de personas.

GUI.- Siglas de "Graphical User Interface". Enlace de comunicación o interfaz entre un usuario y el sistema operativo de una computadora. Generalmente utiliza pantallas con base de iconos que representan las funciones disponibles para el usuario.

Hardware.- Equipo físico. Todos los componentes electrónicos y mecanismos de una red, como computadoras personales, tarjetas de red y cables.

Hot docking.- Son dispositivos de conexión y desconexión en caliente. Es decir, dispositivos que pueden ser conectados o desconectados sin tener que apagar el equipo, por ejemplo, los dispositivos conectados a los puertos USB.

HTML.- Siglas de "Hyper Text Markup Language" (Lenguaje de Marcado de Hipertexto) Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

IBM.- Siglas de International Bussines Machines Corp.

IBM LAN Server.- Sistema de archivos distribuido derivado de LAN Manager, desarrollado y manejado por IBM.

Indizar.- Es la acción de realizar un archivo alterno sintetizado de datos partiendo de una base de datos que puede ser grabado en disco o puede permanecer en memoria, donde residen la ubicación de cierta información dentro de la base de dato.

IT.- Siglas de Tecnología de Información.

Interactua.- Atributo que describe la habilidad de un programa y un usuario o múltiples programas para conectarse con cada uno de los otros durante la ejecución.

Internet.- Enlace entre redes formando la red más grandes del mundo que concentra actualmente los trabajos de estandarización de la familia de protocolos TCP/IP.

Internetwork.- Dos o más redes conectadas por un bridge interno o externo.

Interoperabilidad.- Proceso donde las computadoras pueden operar interactuado con otras a través de la red sin conversión de datos o intervención humana.

Intranet.- Red propia de una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet, en particular el protocolo TCP/IP. Puede tratarse de una red aislada. Es decir. No conectada a Internet. Pero que utiliza navegadores (Netscape, Internet Explorer) para acceder a la información de la organización.

Implementación.- Es un término utilizado en la fase final de desarrollo de un proyecto, significar activar, poner en marcha una aplicación.

Mbps.- (Megabits por segundo) Unidad de medida de la capacidad de transmisión por una línea de telecomunicación. Cada megabit está formado por un millón de bits.

Middleware.- Es aquel software destinado a unir una aplicación con un servidor de manera lógica.

Motor de base de datos.- Parte del sistema de base de datos que recupera y almacena los datos en bases de datos del usuario y del sistema.

ISAM.- es un acrónimo de Indexed Sequential Access Method, es un procedimiento para almacenar y extraer datos desde un archivo de datos en disco. Cuando los diseñadores establecen el formato del archivo un juego de índices es creado el cual describe donde los registros de los archivos son localizados sobre el disco. Esto propicia un acceso rápido a la extracción de datos. Y elimina la necesidad de leer los datos desde el inicio para localizar la información deseada. Los índices de los archivos de datos pueden estar almacenados como parte del archivo de datos o como un archivo índice separado.

ISO.- Siglas de "International Standard Organization". Institución Internacional que se encarga de especificar estándares en diferentes áreas.

Kernel.- Parte del Sistema Operativo que interactúa directamente con el hardware.

LAN.- Siglas de "Local Area Network". Es la abreviación más común al hablar de Redes de Area Local

LAN Manager.- Es uno de los pioneros de lo sistemas operativos para redes LAN de Microsoft, utilizado para enlazar equipos con Windows 3.0 y con MS-DOS.

Laptops.- Es una computadora tipo PC portátil, ligeramente más voluminosa que una computadora PC portátil tipo cuaderno (NoteBook), es precisamente la antecesora de las hoy llamadas NoteBook.

Lógica de la red.- Es aquella distribución y clasificación no física de los elementos constitutivos de la red por medio de software para ser utilizados como se tratará de elementos físicos.

LU.- Siglas de "Logical Unit (Unidad Lógica) es una técnica creada por IBM en el que un puerto de software se establece para llevar a cabo una sesión.

LU 6.2.- Logical Unit 6.2. Unidad lógica que gobierna que gobierna las comunicaciones SNA entre nodos equivalentes (peer-to-peer). Maneja comunicaciones en general entre programas en un entorno de procesamiento distribuido.

Mainframe.- Es una computadora mayor que contiene el CPU (Unidad Central de Proceso) comúnmente operan con longitudes de palabra de 32 bits o más tienen gran capacidad de memoria y son utilizados en aquellas organizaciones donde grandes volúmenes de datos son almacenados y procesados. El término fue acuñado desde los días donde las computadoras ocupaban mucho espacio, ahí sí era realmente un mainframe y este tenía varios "frames" secundarios.

Método.- Procedimiento que opera sobre un objeto.

Monousuario.- Es el término que se utiliza para designar que un producto está restringido a ser utilizado por solo un usuario.

Netware.- Sistema Operativo de red, desarrollado por la compañía Novell tiene diversas versiones y modalidades.

Novell.- Uno de los principales fabricantes de productos para redes locales. Desde 1988 se ha enfocado preponderantemente al mercado de sistemas operativos para red LAN, desligándose casi en su totalidad del hardware para redes locales.

Objeto.- En Programación Orientada a Objetos es una variable que comprende rutinas y datos y que son tratados como una entidad discreta. En vinculación e inserción de objetos (OLE), es un documento o parte del mismo que es pegado o incrustado guardando un vínculo en otro documento mediante el clipboard.

OS/2.- Sistema operativo desarrollado conjuntamente por IBM y Microsoft para la línea de computadoras personales PS/2 (Personal System 2).

OSF.- Siglas de "Open Software Foundation". Organización de proveedores de soluciones UNIX, encargada de estandarizar este mercado.

PC.- Abreviaturas de "Personal Computer", Computadora Personal.

Peer to Peer.- Es aquella comunicación que se establece cuando dos computadoras pueden iniciar una conversación y no requiere de permiso de otra.

Portabilidad.- Propiedad que tiene el software de ser usado en computadoras de diferentes fabricantes.

POSIX.- Siglas de "Portable Operative System Inteface UNIX". Es una propuesta de una interfaz universal para UNIX, de tal manera que los programas de aplicación puedan correr en cualquier equipo, mejorando así la interoperabilidad de sistemas.

PPP.- PPP (Protocolo Punto a Punto) Definido en RFC 1661 proporciona un método para transmitir paquetes a través de enlaces seriales punto a punto.

Programa Objeto.- Es el código ejecutable por una computadora que es generado luego de ser sometido a un proceso de compilación o interpretación.

Proceso:- Se define así a un programa que se ejecute en modo real en una computadora.

Proceso Cliente.- Es aquel proceso capaz de solicitar un servicio.

Proceso Servidor.- Es aquel proceso capaz de proporcionar un servicio DBMS (Data Base Manager System)

Propietario.- Es todo lo contrario a lo estándar, es aquella tecnología que es definida por un fabricante y usado solo en equipos de ese fabricante.

Protector de pantalla.- Es un pequeño programa de seguridad que viene incluido con Windows que se dispara en un tiempo preestablecido y coloca una imagen sobre la pantalla cuando el usuario deja de por unos minutos de usar la computadora.

Protocolo.- Protocolo conjunto de reglas convencionales que regulan el intercambio de información en un sistema de comunicación.

RAS.- Siglas de "Remote Access System". Técnica utilizada por Windows NT para poder conectar dos equipos uno denominado cliente y otro servidor de manera remota utilizando una línea telefónica. Es implementada como un servicio.

RAM.- "Random Access Memory" Memoria de Acceso Aleatorio. Es un tipo de memoria de computadora que es capaz de almacenar datos en forma temporal, siempre y cuando sus celdas estén activas. Es decir. Que la circuitería este alimentada de energía eléctrica y a falta de esta la información almacenada se pierde.

Red Local.- Conjunto de computadoras enlazadas por algún cable, y en distancias relativamente cercanas (por ejemplo dentro de un mismo edificio) también conocida como simplemente LAN por sus siglas en ingles (Local Area Network).

RDA.- Siglas de "Desarrollo Rápido de Aplicaciones". Técnica utilizada para el desarrollo de aplicaciones Cliente/Servidor.

RPC.- Siglas de "Remote Procedure Call". El proceso utilizado en ambientes UNIX con TCP/IP para implementar un proceso específico en un nodo local o remoto.

Ruteador.- Dispositivo que toma un paquete (Nivel 3 del Modelo OSI) y lo envía del punto A al punto B. Después de analizar cual es el camino óptimo para llegar a su destino. Esto se logra gracias a la información que cada ruteador almacena sobre todos los nodos de la red.

Ruteo.- Acción que realiza un ruteador. Véase ruteador.

Servidor.- Dispositivo de hardware o rutina de software que provee uno o más servicios definidos a una población de entidades usuarias, tales como nodos en la red.

Sistema Operativo de red.- Un Sistema Operativo de red es aquel que trata de manera conjunta todos los dispositivos y periféricos de la red, coordina las funciones de todos los dispositivos periféricos de la red y proporciona seguridad y acceso a los datos y a los periféricos.

Software.- Programas y rutinas que indican a la computadora que hacer y en que momento realizarlo.

SNA.- "System Network Architecture": Arquitectura de redes de sistemas. Arquitectura grande, compleja y con múltiples características, desarrollada en la década de los 70s por IBM.

SQL-Server.- Servidor de bases de datos desarrollado por Microsoft y Sybase.

SQL-Windows.- Front-End muy poderoso desarrollado por Grupta Technologies. Es una herramienta tipo 4GL, para generar programas bajo Windows, que acceden a algún DBMS (por ejemplo SQL-Base, SQL-Server, Oracle).

SQL.- Siglas de Structure Query Language. Es el lenguaje de consulta y de acceso a bases de datos más común en la actualidad, definido como estándar por IBM, ANSI y OSI.

TCO.- Siglas de "Total Cost Owership".

TCP/IP.- Juego de protocolos creados en los 70s por Vince Cerf, profesor de Stanford por encargo del Pentágono. El objetivo era crear protocolos independientes del hardware. Hoy en día son los protocolos que la mayor conectividad entre los más diversos equipos.

Topología.- Descripción de las conexiones físicas de una red. Generalmente se conoce con este nombre la forma en que se dispone el cable de una red LAN.

WAN.- Siglas de "Wide Area Network". Se llama así a la red que se extiende sobre distancias muy grandes y que generalmente depende de líneas de comunicación dedicadas para su funcionamiento correcto.

Web.- (malla, telaraña, web) Servidor de información WWW. Se utiliza también para definir el universo WWW en su conjunto.

Web Master.- (administrador de Web) Persona que se encarga de la gestión y mantenimiento de un servidor web, fundamentalmente desde el punto de vista técnico; no hay que confundirlo con el editor de web.

Workstation.- Es comúnmente conocido como una computadora que hace las funciones de un cliente o estación de trabajo dependiente de un servidor.

Bibliografía

Título: Visual Basic Manual del Programador

Autor: Microsoft

Editorial: Microsoft Prést

Título: Análisis Informáticos Implantación de Sistemas Tomo 5

Autor: Autor José Luis Morales Pascual

Editorial: ECC Barcelona España)

Título: Fundamentos de redes Cap. 5

Autor: Microsoft MSCE (Microsoft System Certified Engineer)

Editorial: Microsoft Press

Título: Encyclopedia of Software Engineering

Autor: John J. Marciniak

Volumen I

Editorial : Editor-in -chief

Título: Manging Client/Server Environtments

Autor: John McConnell

Editorial : Prentice Hall

Título: Distributed Windows System A practical Guide to Open Windows

Autor: Allan Davison,

Editorial: Addison_Wesley

Título: Object Oriented Database

Autor : Setrag Khoshafian

Editorial: Willey Professional Computing

Título: Client/Server and Open System

Autor : Rand Dixon

Editorial: John Wiley & Son Inc

Título: A guide to Developing Client/Server SQL Applications

Autor : Setrag Khoshafian

Editorial:

Título: Client/Server Databases

Autor : Joe Salemi

(Segunda edición)

Editorial: Ziff Davis

Título: Microsoft Windows NT Network Administration

Autor : Microsoft

Editorial: Microsoft Press

Título: NT Server Management and Control
Autor : Kenneth L. Spencer
Editorial: Prentice Hall

Título: Visual Basic 4
Autor : Brad Shanman, Conrad Scott
Editorial: SAMS Publishing

Título: Microsoft Windows NT Server Networking Guide
Autor : Microsoft
Editorial: Microsoft Press

Título: Microsoft Training And Certification
Supporting Microsoft Windows NT 4.0 Core Technologies
Student Workbook
Autor : Microsoft
Editorial: Microsoft Press

Título: Guía de Programación 80386
Autor : Lance Leventhal
Editorial: Macrobit Rama

Revista Byte México
Año 7 No. 72, 1994
Artículo : "Sistemas Operativos Avanzados"

Revista Byte México
Año 8 No. 1995
Artículo : "Computación Intergalactica"

Revista Byte México
Año 7 No. 78, 1994
Sección Redes
Artículo : "Cómo ligar el flujo de trabajo con los sistemas heredados"
Sección Reseña hardware y software
Artículo : "Enlazando Equipos de desarrollo"

Revista Byte México
Año 9 No. 104, 1996
Artículo : "Calidad en el Desarrollo"

Revista Byte México
Año 9 No. 114, 1997
Artículo : "Nacen las NUIs"

Revista Byte México
Año 7 No. 77, 1994

Artículo : "Cómputo Distribuido"

Revista: Database advisor

Artículo: Client/Server Advisor /junio 1992 pagina 90

Revista: Windows NT Magazine América Latina

año 2

-CD Microsoft Mastering Visual Basic 5
PRODUCIDO POR MICROSOFT

- CD Service Pack 4 Para Microsoft Windows en Español
PRODUCIDO POR MICROSOFT

CD Microsoft Visual Basic versión 5.0
(Libros en pantalla)
PRODUCIDO POR MICROSOFT

CD Microsoft Solutions Development Kit
PRODUCIDO POR MICROSOFT