

43
24.



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"CAMPUS ARAGON"**

**EL MODELO CLIENTE/SERVIDOR DEL
PROCESAMIENTO DISTRIBUIDO**

T E S I S
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A :
J E S U S M O R O N O R T E G A

ASESOR: LIC. ISRAEL JUAREZ ORTEGA



SAN JUAN DE ARAGON, ESTADO DE MEXICO

1997

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A la memoria de mi hermano Miguel Angel y de mi amigo Daniel
Siempre estarán presentes en mi vida y en mi mente

A mis padres

Por el gran esfuerzo y sacrificio que han hecho por mi y por toda la confianza y cariño que siempre me han demostrado.

A mis hermanos

Por todo el apoyo y la confianza que me brindaron para que pudiera llegar y culminar una carrera profesional, éste triunfo es de ustedes.

A mi esposa Norma

Por permitir compartir tu vida y ser feliz conmigo ya que gracias a tu apoyo, cariño y comprensión me has ayudado a alcanzar otro de los momentos más gratos de mi vida.

Al Lic. Israel Juárez Ortega

**Por la asesoría y consejos que me dio para que pudiera realizar
éste trabajo de tesis.**

A Uriel

**Por que gracias a tu gran ayuda y dedicación has sido parte
importante para que elaborara mi trabajo de tesis.**

A todos mis familiares y amigos

Al departamento de Soporte Informático

Por todo el apoyo y la confianza que me han dado

ÍNDICE

	Tema	Pág.
	INTRODUCCIÓN	1
I.-	INTRODUCCIÓN AL PROCESAMIENTO DISTRIBUIDO Y AL MODELO CLIENTE/SERVIDOR	4
	I.1 Conceptos básicos.....	4
	I.2 Procesamiento centralizado.....	14
	I.3 Procesamiento cooperativo.....	16
	I.4 Procesamiento distribuido.....	17
	I.5 El Rightsizing.....	21
	I.6 El Downsizing.....	23
	I.7 Surgimiento del modelo Cliente/Servidor.....	23
	I.8 Definición del modelo Cliente/Servidor.....	25
	I.9 Tipos de cliente.....	28
	I.10 Tipos de servidor.....	29
II.-	EL MODELO CLIENTE/SERVIDOR	32
	II.1 Alternativas de solución Cliente/Servidor.....	32
	<i>Monitores de Procesamiento de Transacciones.....</i>	<i>32</i>
	<i>Funciones Distribuidas.....</i>	<i>34</i>
	<i>Acceso a Datos Remotos.....</i>	<i>35</i>
	<i>Presentación Remota.....</i>	<i>35</i>
	<i>Distribución de datos.....</i>	<i>36</i>
	II.2 Aumento de la capacidad y eficiencia de plataformas.....	37
	II.3 Localización de la información y de los programas de aplicación.....	37
	II.4 Comunicación entre el cliente y el servidor.....	38
	<i>División de procesos internos para cliente y servidor.....</i>	<i>39</i>
	<i>Llamadas a Procedimientos Remotos (RPC).....</i>	<i>40</i>
	<i>Conversaciones.....</i>	<i>41</i>
	<i>Sistemas de paso de mensajes.....</i>	<i>41</i>
	II.5 Beneficios del modelo Cliente/Servidor.....	42

III.- EL AMBIENTE CLIENTE/SERVIDOR	50
III.1 Plataformas para cliente y servidor.....	50
III.2 Protocolos de comunicación para una red Cliente/Servidor.....	51
III.3 Sistemas operativos de red y servidor para el modelo Cliente/Servidor.....	58
Novell Netware.....	58
Unix.....	59
Open VMS.....	60
WINDOWS NT.....	61
III.4 Dispositivos de red para facilitar el procesamiento distribuido.....	62
Repetidores.....	62
Puentes.....	62
Ruteadores.....	63
Gateways.....	64
Concentradores.....	64
III.5 Objetos distribuidos.....	65
III.6 Infraestructura de comunicaciones en el modelo Cliente/Servidor.....	68
El cableado estructurado.....	69
FDDI.....	69
La Ethernet rápida.....	70
El Frame Relay.....	70
ATM.....	71
III.7 Algunos riesgos y desventajas del ambiente Cliente/Servidor.....	71
 IV.- DISTRIBUCIÓN DE DATOS EN SISTEMAS CLIENTE/SERVIDOR	 74
IV.1 Tecnología de bases de datos distribuidas.....	74
Distribución de datos.....	75
Bases de datos lógicas.....	78
El compromiso en dos fases (two-phase commit).....	79
Ventajas y desventajas de las bases de datos distribuidas.	80

IV.2	Tecnología de bases de datos replicadas.....	81
	<i>Replicación de datos.....</i>	82
	<i>Replicación sincrónica.....</i>	84
	<i>Replicación asincrónica.....</i>	85
	<i>Replicación en cascada.....</i>	86
	<i>Consolidación de información remota.....</i>	87
	<i>Ventajas y desventajas de las bases de datos replicadas...</i>	88
IV.3	Comparación de bases de datos distribuidas y replicadas..	89
V.-	MÉTODOS Y HERRAMIENTAS DE DESARROLLO PARA SISTEMAS CLIENTE/SERVIDOR	91
V.1	Metodologías para el desarrollo de aplicaciones Cliente/Servidor.....	91
	<i>Diseño estructurado.....</i>	91
	<i>Diseño de prototipos.....</i>	93
	<i>Técnicas Orientadas a Objetos.....</i>	95
	<i>Modelado de Datos.....</i>	97
	<i>Diagramas de Entidad-Relación.....</i>	98
	<i>Normalización de bases de datos.....</i>	100
V.2	Uso de herramientas CASE para el desarrollo de aplicaciones Cliente/Servidor.....	102
V.3	Ejemplos de herramientas sql para el desarrollo de aplicaciones Cliente/Servidor.....	103
	<i>Herramientas Front end.....</i>	103
	<i>Herramientas Back end.....</i>	101
VI.-	ESTRATEGIAS CLIENTE/SERVIDOR DE ALGUNOS FABRICANTES	117
VI.1	Cliente/Servidor en IBM.....	117
VI.2	Cliente/Servidor en Digital.....	120
VI.3	Cliente/Servidor en Tandem.....	123
VI.4	Cliente/Servidor en Hewlett Packard.....	125
VII.-	EJEMPLO DE UNA APLICACIÓN CLIENTE/SERVIDOR	128
	CONCLUSIONES.....	136
	BIBLIOGRAFÍA.....	139

INTRODUCCIÓN

Conforme el manejo de la información se hace más complejo respondiendo a las necesidades crecientes de las empresas auxiliándose de la computadora como una herramienta importante en la toma de decisiones, se han seguido desarrollando nuevas tecnologías en computación para facilitar cada vez con mayor eficiencia la información disponible en una empresa y al mismo tiempo optimizar sus recursos.

Con la introducción de las redes de computadoras se dio un paso muy importante en el almacenamiento, procesamiento y transferencia de información entre lo que es una computadora central y terminales dependientes. Pero como cada vez las necesidades son más complejas y se pretende facilitar y optimizar con mayor eficiencia el procesamiento de información, el procesamiento centralizado comienza a bajar de rendimiento porque el software del equipo se maneja en la computadora central, mismo que es compartido con las terminales que no procesan información en forma independiente y hacen el tráfico de transferencia y procesamiento de información muy lento por atender a tantas terminales al mismo tiempo provocando que la velocidad de respuesta disminuya.

El modelo Cliente/Servidor surgió como una alternativa de procesamiento distribuido de la información en las redes de computadoras para contrarrestar las desventajas que ha presentado el procesamiento centralizado. Con este modelo las terminales se convierten en clientes capaces de procesar información de manera independiente y en ellas se almacena parte de los programas de aplicación, mientras que en el servidor se almacena y

administra la base de datos, además de algunos procesos que auxilian a los clientes. Por tal razón el modelo Cliente/Servidor es un tipo de procesamiento en el que la ejecución de una tarea se divide entre el cliente y el servidor explotando al máximo las características de cada uno.

El modelo Cliente/Servidor es una tendencia a adoptar por un gran número de empresas debido su velocidad de procesamiento de información e integración de diferentes tecnologías de hardware y software que anteriormente no habían sido manejadas con seguridad en un solo ambiente.

El objetivo principal de éste trabajo es proporcionar un marco de referencia que explique el modelo Cliente/Servidor de manera general como son: conceptos básicos, elementos necesarios, beneficios y algunos riesgos y desventajas a los que se puedan enfrentar las empresas que adopten el modelo Cliente/Servidor, también dar a conocer cuales son los recursos con que deben contar las empresas para integrarlo dentro de su estructura informática, herramientas para el desarrollo de aplicaciones y propiedades de cada una de ellas, características importantes de algunos sistemas operativos de red para el modelo Cliente/Servidor, qué sistemas son compatibles y los que no lo son cómo se pueden interconectar.

La organización de éste trabajo comienza con un capítulo introductorio al modelo Cliente/Servidor en el que se definen algunos conceptos indispensables y cómo surge Cliente/Servidor. El capítulo II describe de manera general lo que es Cliente/Servidor, algunas alternativas y beneficios de éste modelo. El capítulo III trata sobre ambiente Cliente/Servidor explicando las características de las plataformas para cliente y servidor.

protocolos de comunicación, sistemas operativos de red y para servidor, infraestructura de comunicaciones, objetos distribuidos y algunos riesgos y desventajas que traería la implantación o migración a éste modelo. El capítulo IV trata sobre las tecnologías para distribución de información en sistemas Cliente/Servidor como son bases de datos replicadas, bases de datos distribuidas, ventajas, desventajas y una comparación entre ellas. El capítulo V describe los métodos para el desarrollo de aplicaciones tratando desde los tradicionales hasta la tecnología de objetos, además se incluye la descripción de algunas herramientas para el desarrollo de aplicaciones Cliente/Servidor. El capítulo VI esta dedicado a algunas compañías importantes en el área de la informática que son fabricantes de hardware y de algunos productos que contribuyen a la implantación del modelo Cliente/Servidor. El capítulo VII es la presentación de un ejemplo de una aplicación Cliente/Servidor.

I.- INTRODUCCIÓN AL PROCESAMIENTO DISTRIBUIDO Y AL MODELO CLIENTE/SERVIDOR

El surgimiento del modelo Cliente/Servidor es el resultado de una evolución del procesamiento de la información desde el centralizado, cooperativo y distribuido, hasta el rightsizing* y downsizing**. En el presente capítulo se explican éstos y otros conceptos que son fundamentales para el buen funcionamiento y entendimiento del modelo Cliente/Servidor, algunos de ellos se describen a manera de introducción y en los capítulos siguientes se explicarán en forma más detallada.

1.1 Conceptos básicos

1.1.1 Cliente

Es el elemento conectado a una red de computadoras capaz de procesar información en forma independiente, este requiere de la acción de un servidor al cual le hace sus peticiones para aprovechar los recursos de la red. Un cliente puede ser desde una PC o Macintosh, hasta una estación de trabajo u otra red.

1.1.2 Servidor

Es el elemento al que acuden todos los clientes conectados a la red para la compartición de los recursos disponibles en ella, se encarga de almacenar la información y de administrar la red. De acuerdo a los recursos suministrados hay diferentes tipos de servidores: de bases de datos, de archivos,

* Reevaluación de los recursos de cómputo

** Reestructuración de los recursos de cómputo

comunicaciones, impresión y aplicación, pero también se clasifican por su funcionamiento en dedicados o no dedicados.

En la **figura 1.1** se muestra la conexión entre un cliente y un servidor en la red, donde el cliente la utiliza para hacer peticiones al servidor.



Fig. 1.1 Modelo básico Cliente/Servidor presentado por DEC en 1991.¹

1.1.3 Aplicación

Es un programa o conjunto de programas generados por métodos y herramientas de desarrollo para automatizar actividades que comúnmente se repiten en una empresa. En el modelo Cliente/Servidor una parte de la aplicación se instala en los clientes y otra en el servidor con la finalidad de distribuirse el procesamiento de información.

1.1.4 Transparencia

La Transparencia la proporciona el sistema operativo de la red y se da cuando un cliente solicita un servicio sin importarle la localización física de quien se lo está proporcionando, es decir, trabaja como si lo que él necesitara estuviera en su computadora.

¹ Digital Equipment Corporation, Integrated Desktop Computing, USA 1991, p. 1-2.

1.1.5 Interfaz Gráfica de Usuario, GUI

La Interfaz Gráfica de Usuario (GUI, Graphical User Interface) es el medio más práctico para que el usuario interactúe con la computadora mediante menús desplegables en ambientes gráficos fáciles de manejar, ocultan la complejidad del uso de instrucciones y comandos. El usar una GUI proporciona un ambiente de trabajo cómodo, amigable, rápido, intuitivo y muy atractivo. Por medio de la GUI los procesos son transparentes para el usuario debido a que ésta interfaz se encuentra en el cliente. Además, con una GUI se pueden desarrollar aplicaciones que tengan características al ambiente de desarrollo.

1.1.6 Front end

Es la interfaz entre el usuario y el cliente. Aquí se introducen y validan los datos a procesar para después enviarlos al servidor. Se caracteriza por tener una GUI donde se despliegan menús, datos y resultados. Por medio del Front end el cliente hace todas sus peticiones al servidor. Constituye una parte de la aplicación instalada en el cliente dentro de un ambiente de procesamiento distribuido.

1.1.7 Back end

Forma la parte oculta del proceso de la información: son las instrucciones que se ejecutan en el servidor, tales como abrir y cerrar archivos, ordenaciones, indexaciones e instrucciones de manipulación de base de datos. En el Back end las operaciones con los archivos se realizan con mayor proximidad. También se da la administración y seguridad de la base de datos

en él almacenada. Constituye la otra parte de la aplicación instalada en el servidor dentro de un ambiente de procesamiento distribuido.

1.1.8 Plataforma

Es una combinación de software y hardware de sistema operativo y sirve como base de servicios de soporte de aplicaciones de red. Mediante una plataforma se pueden conjuntar diferentes arquitecturas para proceso cooperativo, una PC además de su sistema operativo puede ejecutar otros sistemas operativos más robustos y viceversa.

En lo que se refiere al hardware, la plataforma puede estar basada en un procesador Intel, Motorola o por el conjunto de instrucciones que maneje dicho procesador, ya sea RISC* o CISC** y aunado a ello el fabricante del equipo como pueden ser IBM, DEC, TANDEM, COMPAQ y HP entre otros. Desde el punto de vista software, también se considera al sistema operativo en sí como una plataforma.

1.1.9 Middleware

El Middleware es un producto de software utilizado en el modelo Cliente/Servidor para el desarrollo de aplicaciones y tiene una interfaz de comunicación que oculta la complejidad de la interacción del Front end con el Back end en sistemas con protocolos diferentes, haciéndolo de esta forma transparente. También con el Middleware es posible la traducción entre diferentes Interfaces de Programación de Aplicaciones (API).

* Complex Instruction Set Computer.

** Reduced Instruction Set Computer.

Por ejemplo, en las versiones recientes de Novell Netware y Windows NT se han desarrollado productos middleware para optimizar el modelo Cliente/Servidor y tener independencia entre aplicaciones, integración de diferentes plataformas y además del manejo de datos distribuidos.

En la **figura 1.2** se muestra como los clientes hacen sus peticiones en una red con diferentes protocolos y sistemas operativos con la finalidad de que el usuario utilice aplicaciones aún corriendo en diferentes plataformas.

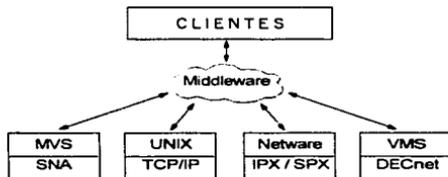


Fig. 1.2 Interconexión de diferentes plataformas con los clientes a través del middleware.

El middleware se puede clasificar en tres tipos: Llamadas a Procedimientos Remotos (RPC, Remote Procedure Calls), Sistemas con paso de mensajes y las Conversaciones. Algunos ejemplos de middleware son: ODBC (Open Data Base Connectivity) Conectividad Abierta de Bases de Datos de Microsoft; Arquitectura de Base de datos Relacionales Distribuidas (DRDA, Distributed Relational Database Architecture) de IBM; SqlNet de Oracle.

1.1.10 Interfaz de Programación de Aplicaciones. API

La Interfaz de Programación de Aplicaciones (API, Application Programming Interface) permite establecer comunicación entre aplicaciones que se ejecutan en sistemas heterogéneos. "Las API's son el formato de lenguaje y mensajería que define cómo interactúan los programas con las funciones de otros programas, con sistemas de comunicación o controladores de hardware"². Los formatos ya están definidos en el producto (Sistema Operativo o en el Middleware) incluyendo rutinas que permiten a los programadores desarrollar módulos para un sistema específico de manera transparente y que estos puedan trabajar en diferentes plataformas (computadoras con sistemas operativos diferentes). Las API's pueden clasificarse en tres tipos como se muestra en la **figura 1.3**.



Fig. 1.3 Clasificación de las API's.

Las API's de **Front end** están formadas por el conjunto de bibliotecas que permiten al usuario establecer una sesión transparente ocultándole la complejidad de instrucciones que están detrás del proceso que tienen que realizar, estas pueden ser herramientas gráficas. Las de **servicios de red** trabajan en los niveles más bajos para establecer comunicación entre sistemas con diferentes protocolos conectados a la red dentro de un ambiente distribuido. Por último las API's de **administración de bases de datos**

² Sheldon, Tom. Enciclopedia de redes, Osborne McGraw Hill, México 1995, p. 495.

enlazan el programa de aplicación del cliente con el sistema administrador de bases de datos del servidor. Existen API's diseñadas para trabajar en sistemas homogéneos, pero para el modelo Cliente/Servidor es necesario utilizar API's que trabajen en diferentes plataformas y en un ambiente distribuido.

1.1.11 Transacción

Es considerada como una unidad discreta de trabajo que implica sentencias ejecutadas en una o múltiples bases de datos, por lo regular se utiliza un Lenguaje de Consulta Estructurada (SQL, Structured Query Language) para implementar las transacciones de algunos negocios. Una transacción implica la modificación de una base de datos en un ambiente de procesamiento centralizado o distribuido. La ejecución de las transacciones puede ser *en línea* (en tiempo real) o bien *por lotes* (se almacenan por un tiempo para ejecutarse después). Esta serie de sentencias SQL se ejecutan todas o ninguna, lo que permite garantizar la integridad de base de datos. Existen algunas aplicaciones Cliente/Servidor que utilizan el procesamiento de transacciones en sistemas distribuidos, como es el caso del Procesamiento de Transacciones En-Línea (On-Line Transaction Processing, OLTP).

1.1.12 Procesamiento de Transacciones en Línea. OLTP

El Procesamiento de Transacciones en Línea (OLTP, On-Line Processing Transaction) consiste en la recolección y procesamiento de datos para una transacción y envía los cambios a la base de datos en forma inmediata, de esta manera los resultados de un OLTP están disponibles al instante en la base de datos una vez terminada la transacción. Un ejemplo clásico del OLTP son las transacciones bancarias y los sistemas de reservación en líneas aéreas.

1.1.13 Interoperabilidad

“La interoperabilidad es la capacidad de que diferentes sistemas de computadoras, redes, sistemas operativos y aplicaciones trabajen juntas y compartan información”³, esto significa que es una integración eficiente de equipo de diversos proveedores en un ambiente de funcionalidad, incluyendo interfaces comunes y una similitud en comunicaciones y aplicaciones para que los diferentes usuarios conectados a la red puedan acceder a la información y recursos de los distintos sistemas. Aquí se involucran diferentes clientes DOS, Windows, OS/2, Macintosh y plataformas Unix, VAX e IBM. La interoperabilidad se puede dar gracias al uso de algunos dispositivos de red como los puentes, gateways y ruteadores para reducir el tráfico en la red, interconexión de sistemas de diferentes arquitecturas y para el encaminamiento inteligente de los paquetes de información en la red respectivamente, entre otras.

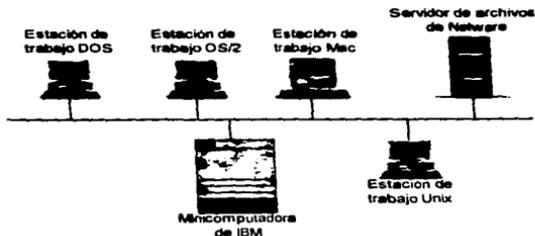


Fig. 1.4 Red que conecta de forma transparente sistemas heterogéneos, *Conectividad*.⁴

³ Sheldon, Tom. Enciclopedia de redes, Osborne McGraw Hill, México 1995, p. 512.

⁴ Bradley F. Shimmin - Sheldon. Guía de interoperabilidad, Osborne McGraw Hill, México 1995, p. 24.

En la **figura 1.4** los clientes DOS, OS/2, Macintosh y Unix pueden acceder a los recursos de los servidores IBM o Novell como si fueran una sola arquitectura.

1.1.14 Sistemas Abiertos

Sistemas Abiertos, es el enfoque adoptado por los proveedores de hardware, software y dispositivos de comunicación para producir sistemas que se ajustan a los estándares internacionales de comunicación abierta con la finalidad de interactuar en un ambiente de funcionalidad. "Son sistemas de computación, de comunicaciones o ambos, cuyas especificaciones son ampliamente disponibles, aceptadas y estandarizadas"⁵, este se enfoca bajo un sistema de normas que se utilizan en diferentes niveles: de comunicación, sistemas operativos, aplicación y usuario, estas se muestran en la **figura 1.5**.

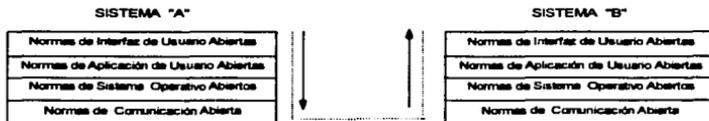


Fig. 1.5. Normas para Sistemas Abiertos

Las *Normas de Comunicación Abierta* están encaminadas a uniformizar el proceso de comunicación entre los sistemas de cómputo, por ejemplo el modelo de referencia OSI (Open System Interconnection-Interconexión de Sistemas Abiertos); Las *Normas de Sistemas Operativos Abiertas* son usadas para que los fabricantes de sistemas operativos las tomen como referencia para diseñar sus productos, tal es el caso Unix; Ejemplo de *Las Normas de Interfaz de*

⁵ Deitel, H. M., Sistemas Operativos, Addison Wesley Iberoamericana, USA 1993, p. 13

Usuario Abiertas es el X-Window System y por último las *Normas de Aplicación de Usuario Abiertas* adoptadas por la X/Open y la Organización de Software Abierto (OSF-Open Software Fundation).

1.1.15 Conectividad

La conectividad es la especificación utilizada para facilitar el empleo de software o hardware a los usuarios de un sistema de cómputo con múltiples clientes con diferentes formatos de archivos o con plataformas heterogéneas. En un sistema Cliente/Servidor debe existir la conectividad entre las aplicaciones y arquitecturas de computadoras que lo conforman.

1.1.16 Conectividad de Bases de Datos

El empleo de diferentes bases de datos en una misma empresa es conveniente cuando la naturaleza de cierto tipo de aplicaciones requiere su desarrollo en una base de datos en particular para tener un mejor rendimiento, pero cuando es necesaria la comunicación entre estas aplicaciones se debe contar con sistemas abiertos, disponibilidad de software para trabajo en red y conectividad de diferentes compañías de bases de datos. Esto es lo que se busca con la conectividad de bases de datos en las que se pretende un mejor rendimiento, portabilidad y flexibilidad.

Un factor importante para este tipo de bases de datos es el Lenguaje de Consulta Estructurado (SQL, Structured Query Language) utilizado como estándar para las diferentes compañías. Microsoft ha desarrollado un producto que funciona como middleware denominado ODBC (Conectividad Abierta de Bases de Datos) en el que un cliente que cumple con un ODBC puede

interactuar con cualquier administrador de base de datos de servidor que también cuente con ODBC, **Figura 1.6**.



Fig. 1.6. Acceso de diferentes clientes a una misma base de datos utilizando ODBC.

1.2 Procesamiento centralizado

El modelo del procesamiento centralizado comenzó a utilizarse para proceso de datos en las empresas debido a que el CPU, la memoria y dispositivos de almacenamiento eran tan caros que era prácticamente imposible que todos los usuarios de uno o varios departamentos tuvieran estos recursos individualmente, por lo que se adoptó un modelo centralizado, en el que todos los recursos de cómputo están localizados en un mainframe al que se conectan todos los usuarios en forma de terminales que no procesan información (tontas), en este modelo se concentran: todos los programas de aplicación, administración y almacenamiento de la información, además del procesamiento de datos de todos los usuarios. En la **figura 1.7** se muestra un ejemplo del procesamiento centralizado en el que todo el poder de cómputo está localizado y controlado por la computadora central.



Fig. 1.7. Modelo del procesamiento centralizado.*

Entre las ventajas que ofrece el procesamiento centralizado están:

- Capacidad para administrar grandes bases de datos desde un solo lugar.
- La consolidación de todos los datos que maneja el sistema en la computadora central facilita la eliminación de datos redundantes.
- Se tienen altos niveles de seguridad en el acceso a la información.
- Facilidad para respaldar la información.
- Ejecución de procesamiento de transacciones en unidades de tiempo pequeñas.

Así mismo comenzaron a surgir algunos problemas en el modelo del procesamiento centralizado provocando que éste comenzara a bajar de rendimiento para algunas aplicaciones. El inconveniente principal es el que los usuarios solo son terminales que no procesan información de manera independiente y para que uno de ellos pueda trabajar depende de la disponibilidad del servidor, el número de usuarios y de qué tantas peticiones se le hagan en un intervalo dado. Otro factor decisivo para que el procesamiento centralizado comenzara a bajar de rendimiento es lo indispensable de utilizar aplicaciones en plataformas de más bajo costo para que los usuarios puedan

* Day, Michael. Downsizing to Netware. NRP, U.S.A. 1992, p.29.

pasar de un ambiente de PC o de estación de trabajo al de la red y viceversa. También se comenzó a tener la necesidad de la distribución del procesamiento entre varias computadoras (procesamiento en paralelo como sucede en las bases de datos replicadas y distribuidas) para reducir el tiempo requerido al realizar una tarea, lo cual era muy difícil de darse.

Además en el procesamiento centralizado, el servidor de la red a parte de tener almacenada la información y los programas de aplicación, realiza todos los procesos correspondientes a cada usuario y se encarga de mantener los niveles de seguridad en la red. Esto implica una carga de trabajo pesada cuando se tiene una gran cantidad de usuarios conectados y al hacer demasiadas peticiones de información y procesamiento de la misma provoca el tráfico en la red, reduciendo la velocidad de procesamiento del servidor para cada usuario. En el procesamiento centralizado se trabaja por lo regular bajo un mismo ambiente de hardware y software, pero si la empresa requiere utilizar otro tipo de ambientes de diferentes proveedores en algunos casos hay problemas de compatibilidad de hardware y portabilidad de aplicaciones o no hay mucha flexibilidad. Estos beneficios los tienen las computadoras pequeñas además resulta mas económico actualizar o reemplazarlas aunque para un ambiente distribuido se tiene que emplear software más complejo.

1.3 Procesamiento cooperativo

El proceso cooperativo implica una interacción entre el sistema operativo de la red y los sistemas operativos de los clientes pudiendo ser también otros servidores que atiendan diversas aplicaciones, por lo tanto, se puede manejar

un sistema operativo de red como Netware, Unix o VMS con clientes DOS, OS/2, Macintosh y Windows. En la figura 1.8 se representa la forma en la que se podrían interconectar los sistemas operativos de los clientes con el sistema operativo del servidor.

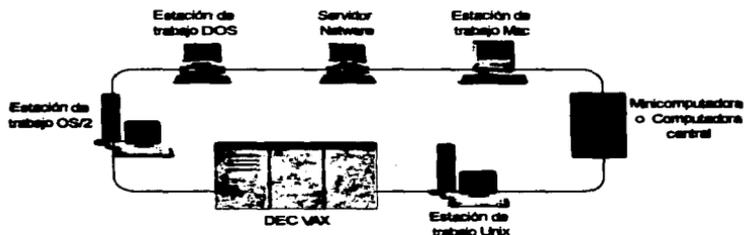


Fig. 1.8. Conexión de diferentes sistemas operativos para procesamiento cooperativo

Este tipo de procesamiento se aplica a sistemas informáticos que están distribuidos, donde más de una computadora comparte el proceso de una aplicación o tarea. Un sistema de esta magnitud debe permitir que los clientes se comuniquen entre sí.

1.4 Procesamiento distribuido

Con la aparición de las microcomputadoras se dio un gran impulso en las redes de área local (Lan) debido a su bajo costo y velocidad de procesamiento. Esto permitió que el procesamiento de la información evolucionara y surgiera un nuevo modelo: el de procesamiento distribuido,

donde los datos no se localizan en un solo servidor, sino en varios servidores que están ubicados en diferentes zonas geográficas conectados con enlaces de redes de área extensa (Wan).

Cuando se comenzó a utilizar la microcomputadora fue necesario incrementar la velocidad de los medios de comunicación porque cada una almacena sus propios datos, corre sus propios programas y procesos, independientemente de las demás microcomputadoras de toda la organización, pero aún así debe existir comunicación entre ellas. Estas computadoras tienen la capacidad de compartir datos en una Lan y se introdujo la alternativa del procesamiento distribuido en el que la combinación de muchas microcomputadoras de bajo costo trabajando en conjunto, resulta mucho más barato que comprar un mainframe. En el procesamiento distribuido los recursos de cómputo están situados donde son utilizados y los datos se almacenan donde son generados o procesados, **figura 1.9.**

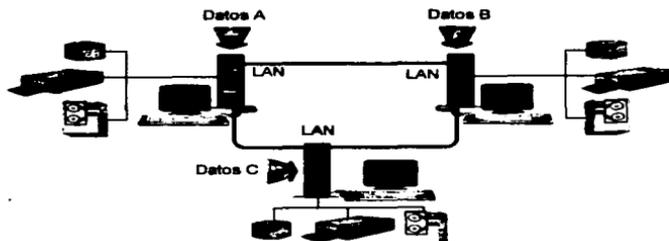


Fig. 1.9. Los recursos de cómputo, generación, almacenamiento y procesamiento de datos, son los elementos importantes del procesamiento distribuido.

Partiendo de estas características se determina que hay una distribución de procesamiento de una misma aplicación entre diferentes servidores para aumentar la productividad de la empresa, también se aplica cuando se tienen diferentes datos ubicados en lugares independientes pero que tienen la necesidad de compartir información entre sí y al mismo tiempo cada zona descentralizada debe ejercer una administración y control de sus propios datos, figura 1.10.

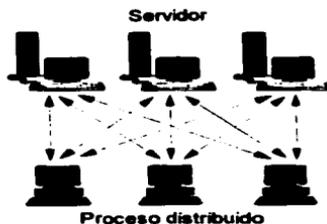


Fig. 1.10. Procesamiento dividido entre varios servidores a los que pueden acceder los diferentes clientes.

Existe una diferencia muy grande entre el modelo del procesamiento centralizado y el distribuido ya que en el primero muchos usuarios accesan a una sola computadora central, mientras que en el otro muchos usuarios accesan a muchas computadoras tal como se muestra en la figura 1.11, donde las computadoras pueden acceder a otras, hay un dispositivo de información centralizada, además de que puede heredar los datos. Los usuarios remotos pueden acceder datos específicos, que están disponibles en los sistemas por etapas, reduciendo de esta manera la carga de trabajo de los

sistemas a los que invoca toda la empresa. Puede existir comunicación entre usuarios de una zona común o diferente y cuando son de la misma se pueden formar grupos de trabajo. Cuando se aplica apropiadamente el procesamiento distribuido se incrementa la productividad de toda la organización obteniendo comunicaciones mejores y efectivas para incrementar el acceso a los datos existentes, para introducir mejores métodos en la obtención de datos y poder acceder a los recursos de cómputo por toda la empresa.

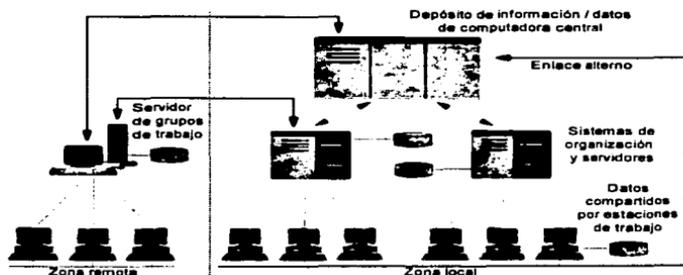


Fig. 1.11. Métodos de acceso en un entorno de procesamiento distribuido.⁷

Algunas de las ventajas que presenta el modelo del procesamiento distribuido son la escalabilidad, modularidad y redundancia.

1.4.1 Escalabilidad.

Es la acción de agregar usuarios a una red con el fin de incrementar la capacidad de la misma para ejecutar aplicaciones distribuidas, obteniéndose de

⁷ Sheldon, Tom. Enciclopedia de redes, Osborne McGraw Hill, México 1995, p.458.

esta manera más poder de procesamiento, de memoria, liberación de carga de trabajo, entre otros.

1.4.2 Modularidad.

“Es la división de una aplicación en un Front end y en un Back end”⁸. Consiste en la división de funciones en módulos, de tal manera que si se particiona la aplicación en uno o varios Frontends y en un Back end, se tiene la posibilidad de trabajar con varios Front end ya sea bajo ambiente DOS, Windows, Unix o Macintosh, la modularidad permite a estos tipos de clientes acceder a los servicios del mismo Back end.

1.4.3 Redundancia.

Esta ventaja proporciona protección ante una caída del sistema o fallas en el hardware. En una Lan los datos redundantes se pueden direccionar (rutear) hacia otro segmento de la red. Para implementar la redundancia en una Lan es necesario el uso de puentes para abrir la comunicación entre dos o más segmentos diferentes, o el empleo de ruteadores para el control de tráfico. Ambos dispositivos se encargan de que los datos lleguen a sus destinos correctos.

1.5 El Rightsizing

El constante desarrollo de microcomputadoras y el diseño de Redes de Área Local (Lan) han sentado las bases para implementar nuevas tecnologías

⁸ Day, Michael. Downsizing to Netware, NRP, U.S.A. 1992, p. 37.

para el procesamiento de la información, una de ellas es el Rightsizing, que es un proceso de **reevaluación** de los recursos de los sistemas de información para determinar como pueden satisfacer los requerimientos actuales de la empresa y después involucrarlos para maximizar el costo-beneficio y mantener la ventaja competitiva. Esta reevaluación de recursos considera desde un mainframe, minicomputadoras y una red de PC's, el tipo de aplicación y la organización de la información, hasta la distribución de procesos y datos en diferentes computadoras. En la **figura 1.12** se describe el rightsizing.

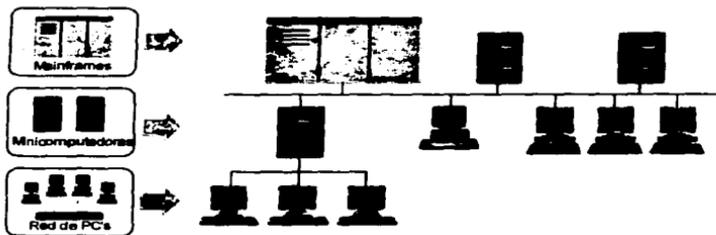


Fig. 1.12. El rightsizing describe el movimiento de los sistemas de información desde una computadora central (mainframe) y redes Lan de PC's independientes a sistemas distribuidos en empresas extensas⁹.

En esta figura se observa que el rightsizing y procesamiento distribuido se dan simultáneamente al haber una distribución y proceso de datos entre diferentes computadoras sobre una misma tarea. Al mismo tiempo se maneja el concepto de sistemas abiertos haciendo referencia a las diferentes normas de

⁹ Sun. Rightsizing, reengineering information systems through client/server technology. U.S.A. 1994. p.3.

comunicación, de sistemas operativos y de aplicación e interfaz de usuario. Lo que indica que el rightsizing sentó las bases para el procesamiento distribuido y el surgimiento del modelo Cliente/Servidor.

1.6 El Downsizing

Algunos autores definen este concepto como “reestructuración” en el que las empresas reemplazan sus computadoras centrales y minicomputadoras por servidores y estaciones de trabajo basados en redes de área local, en este proceso se transfieren los datos y programas a las microcomputadoras conectadas a la red e involucra además un rediseño de los métodos que el personal utiliza para manejar y dar mantenimiento a esos datos y programas.

“El downsizing es más que una transferencia de datos, distribución de aplicaciones y un proveedor de procedimientos hacia una nueva plataforma, este involucra una diferencia conceptual en el salto de un procesamiento centralizado a distribuido”¹⁰, donde en el procesamiento distribuido todos los recursos están localizados en diferentes lugares distribuidos físicamente.

Aunque la realización de un downsizing dependerá totalmente de las necesidades de la empresa y naturaleza de las aplicaciones.

1.7 Surgimiento del modelo Cliente/Servidor

El procesamiento de la información ha evolucionado de tal manera que se han ido presentando algunas formas y fenómenos particulares para el

¹⁰ Day, Michael. Downsizing to Netware, NRP, U.S.A. 1992, p. 13.

tratamiento de la misma hasta llegar al modelo Cliente/Servidor, en la **figura 1.13** se describe como es que se origina éste modelo.

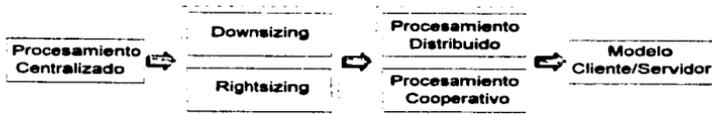


Fig. 1.13. Antecedentes del Modelo Cliente/Servidor

El primer modelo de procesamiento es el centralizado basado en mainframe y es el que procesa la información de todos los usuarios, posteriormente surge el downsizing que es una reestructuración de los sistemas de cómputo donde se reemplazan los mainframes y minicomputadoras por servidores y estaciones de trabajo basados en redes de área local y, el rightsizing en el que se reevalúan los recursos de cómputo para mover los sistemas de información a través de mainframes y redes de área local basadas en PC's independientes a sistemas distribuidos.

Con ello surge el modelo distribuido en el que se dividen el proceso y los datos entre varios servidores distribuidos en diferentes zonas geográficas para dar servicio a varios clientes que ya pueden trabajar independientemente y son PC's o estaciones de trabajo. El proceso cooperativo permite que los sistemas operativos de los clientes puedan interactuar con el de la red. Aquí surge el modelo Cliente/Servidor donde una aplicación se divide en un Front end y un Back end para aprovechar de manera óptima la capacidad de cada elemento.

1.8 Definición del modelo Cliente/Servidor

El concepto del modelo Cliente/Servidor no es nuevo, se dio a finales de la década de los 80's y fue adoptado por Microsoft y Sybase para describir la tecnología de un servidor SQL. Actualmente éste concepto se maneja a diario en el ámbito de las redes de computadoras desde los desarrolladores de aplicaciones para red hasta el personal de comunicaciones. Antes de definir éste modelo se presentarán algunas definiciones de lo que es Cliente/Servidor.

"La computación Cliente/Servidor es la relación entre dos procesos que son cooperativos en la ejecución de una misma tarea. El cliente requiere que alguna función se procese, el servidor procesa esta función, la función se refiere a un servicio"¹¹.

"El modelo Cliente/Servidor es una formalización del procesamiento distribuido en el que una computadora actúa como un servidor de recursos y otras computadoras actúan como los clientes de los recursos"¹².

Estas definiciones tienen la esencia de lo que es Cliente/Servidor, la distribución de procesamiento de una misma tarea entre un cliente y un servidor, pero aún faltan algunos aspectos a considerar dentro de ellas, para este caso se considera una aplicación de base de datos.

Cliente/Servidor es un modelo del procesamiento distribuido en el que la ejecución de una tarea se divide entre un sistema cliente y un servidor para

¹¹ Digital Equipment Corporation. Guide to building Client/Server solutions. U. S. A. 1993. p. 1-2

¹² Day, Michael. Downsizing to Netware. NRP. U.S.A. 1992. p.44

utilizar de la mejor manera la capacidad de cada uno. El cliente generalmente puede ser una PC o una estación de trabajo con una interfaz gráfica de usuario, el servidor puede ser desde otra PC, estación de trabajo, una minicomputadora hasta un mainframe. En el modelo Cliente/Servidor una parte de la aplicación se instala en el cliente (o clientes) ya que tiene la capacidad de procesar información de manera independiente y esto constituye la parte frontal o Front end de ésta aplicación, mientras que la otra parte se instala en el servidor y son los procesos más pesados de la aplicación que son transparentes para el cliente. Esto forma la parte oculta o Back end.

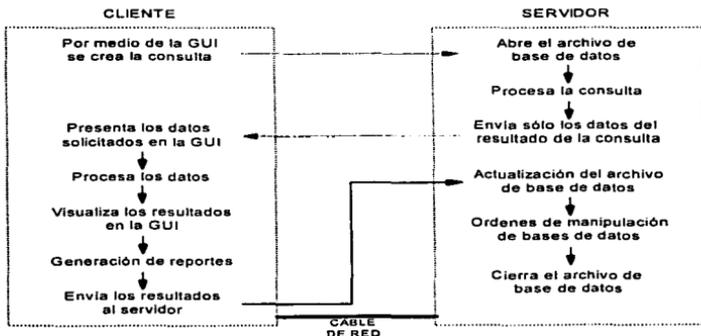


Fig. 1.14. Procesos que se realizan en el modelo Cliente/Servidor.

En la **figura 1.14** se muestran los procesos que se realizan tanto en el cliente como en el servidor, aquí el cliente genera las restricciones para que el servidor solo le envíe la información necesaria para que se visualicen en el Front end y pueda realizar sus procesos independientes, los resultados se

muestran en pantalla para el usuario y pueda generar algunos reportes y enviar el resultado al servidor para que se realicen las operaciones de manipulación de base de datos. De esta manera se reduce considerablemente el tráfico en la red. También se observa que en el modelo Cliente/Servidor los usuarios no tienen un acceso directo a los archivos de base de datos para realizar consultas y modificaciones, ya que las operaciones más próximas con los archivos se realizan en el Back end. Esto implica que se tiene un almacenamiento centralizado en el servidor con lo cual se pueden manejar controles de seguridad para restringir y supervisar el acceso a la información.

La finalidad de distribuir el proceso de una aplicación consiste en asignar las operaciones que mejor se ejecuten en cada parte, es decir, en el cliente se aprovecha la facilidad del manejo de un sistema de escritorio (PC una estación de trabajo), la familiaridad con el usuario, velocidad de proceso, compatibilidad de hardware y software, escalabilidad, resolución e interfaz gráfica, sistema operativo, entre otros. En el servidor se manejan grandes volúmenes de información, mayor cantidad de memoria, facilidad para el manejo del sistema operativo de la red para la administración de recursos y niveles de seguridad, manejo de las comunicaciones y ejecución de tareas pesadas.

Hasta el momento sólo se ha definido el modelo Cliente/Servidor, los beneficios, desventajas, arquitectura, plataformas, datos distribuidos, y herramientas de desarrollo se describirán en los capítulos siguientes.

1.9 Tipos de cliente

Los clientes conectados a la red pueden variar dependiendo del tipo de aplicación y de la naturaleza de la tarea a realizar.

1.9.1 Computadoras Personales.

Cuando se ha seleccionado un sistema operativo DOS, OS/2 o Windows como un estándar para la empresa, se tiene flexibilidad y facilidad para correr aplicaciones en la misma PC dando por consiguiente una mayor portabilidad y compatibilidad de software con otras. La ventaja principal de utilizar una PC como cliente es que por medio de ellas se pueden expandir las Redes de Área Local a un bajo costo y debido a que tanto las mismas PC como sus aplicaciones pueden emigrar hacia las nuevas tecnologías para incrementar su rendimiento.

1.9.2 Macintosh.

A diferencia de las computadoras compatibles con IBM que utilizan tecnología Intel, las Macintosh de Apple Computer emplean tecnología Motorola para sus aplicaciones. Estas computadoras son las pioneras en utilizar la Interfaz Gráfica en el sistema operativo ya que en su arquitectura esta involucrada una GUI haciéndola más eficiente para el manejo de gráficos que otras computadoras. Este tipo de computadoras ya están más difundidas porque Apple ya ha introducido en el mercado productos para que sean más abiertas y ahora se pueden considerar como clientes de muchos sistemas operativos de red, así mismo tienen la posibilidad de ser considerados para ambientes distribuidos.

1.9.3 Estación de trabajo.

Se utiliza como cliente cuando se maneja un sistema operativo de red como Unix o VMS en donde las aplicaciones requieren de pantallas grandes con un manejo de gráficos de alta resolución como son aplicaciones científicas y de ingeniería con procesadores de alto rendimiento. Ejemplos de estas computadoras son equipos IBM, Digital, Sun, HP, etc. Recientemente se comienza a utilizar el término estación de trabajo a todas las computadoras que están conectadas a las redes.

1.9.4 Una red como cliente.

Cuando se tienen aplicaciones distribuidas en diferentes lugares geográficos tales como bases de datos distribuidas ó replicadas se puede utilizar un servidor que actúe como cliente haciendo peticiones al servidor principal ya sea para solicitar información de otra zona o para actualizar las bases de datos, de esta manera se tiene servidores de red conectados a otra red.

1.10 Tipos de servidor

Dependiendo de la naturaleza de las aplicaciones que se comparten en red es como se tendrá un determinado tipo de servidor, o bien puede ser que un solo servidor desempeñe diferentes funciones.

1.10.1 De base de datos.

Es un depósito central de información al que accesan muchos usuarios mediante el uso de comandos de SQL porque en este tipo de servidor los datos

se almacenan en forma estructurada o como información orientada a objetos. Regularmente en un sistema Cliente/Servidor los usuarios hacen sus peticiones al servidor de base de datos mediante un Front end, mientras que la mayor parte del procesamiento la hace el mismo servidor de base de datos (Back end).

1.10.2 De archivos.

Su función es proporcionar los servicios de acceso a disco (almacenamiento y recuperación de archivos), niveles de seguridad para controlar los derechos de acceso y compartición de los archivos, lo cual significa que ofrece dos tipos de servicios: para usuarios de la red y para los administradores de la misma. Debe proporcionar sistemas de protección a los datos como las copias de seguridad, discos espejo, control de la fuente de alimentación ininterrumpida. Además de almacenar los datos de los usuarios también almacena sus programas para el manejo de los mismos.

1.10.3 De impresión.

Cuando se tienen varios dispositivos de impresión conectados a la red (graficadores, fax's y sobre todo impresoras) se puede utilizar un servidor para compartirlos con los usuarios de la red usando funciones de administración de trabajos por medio de sistemas de colas de impresión.

1.10.4 De comunicaciones.

Administra las conexiones entre los nodos de una red, conexiones con puntos remotos (cuando se tiene software de acceso remoto) o bien la conexión con las computadoras centrales. Es usado cuando se requiere de la

transferencia o acceso a archivos en lugares remotos a través de enlaces de telecomunicaciones. Para efectos de integración, traduce la información en varios formatos para que los usuarios se puedan comunicar con los de otros sistemas (conversión de formatos de datos, protocolos y señales de cable). También ofrecen servicios de correo electrónico para depositar y recoger correo en otras Redes de Área Local.

1.10.5 De aplicación.

Proporcionan servicios para algunos sistemas que necesitan compartirse entre varios clientes que utilizan SQL en una red (DOS, OS/2, ULTRIX, Macintosh, VMS). También se emplean para integrar datos de varios departamentos, además de almacenar las aplicaciones se pueden administrar desde un solo sistema. Este servidor puede ejecutar una parte del proceso de la aplicación o solamente proporcionar el acceso de los usuarios a las aplicaciones. En el modelo Cliente/Servidor las aplicaciones del servidor se ejecutan en el Back end del proceso distribuido.

II.- DESCRIPCIÓN DEL MODELO CLIENTE / SERVIDOR

En este capítulo se explicarán algunas alternativas para implementar sistemas Cliente/Servidor además de los procesos de comunicación, beneficios y consideraciones para la distribución de tareas y datos entre los clientes y el servidor.

II.1 Alternativas Cliente/Servidor

La finalidad del modelo Cliente/Servidor es integrar ambientes complejos de múltiples plataformas, protocolos y redes para que se mantenga la simplicidad como si fuera un solo sistema. Para ello hay algunas alternativas Cliente/Servidor dependiendo de la naturaleza de la aplicación, entre ellas están: los monitores de procesamiento de transacciones, funciones distribuidas, acceso a datos remotos, presentación remota y la distribución de datos.

II.1.1 Monitores de Procesamiento de Transacciones

Existen bases de datos que se consideran de misión crítica y emplean sentencias SQL para ejecutar transacciones en ambientes distribuidos y deben estar sincronizadas para mantener la integridad de la información (arquitectura Cliente/Servidor de tres niveles). Los manejadores de estas bases de datos deben tener Monitores de Procesamiento de Transacciones (MPT) para administrar los procesos y dirigir los programas particionando las aplicaciones complejas en varios módulos (transacciones). "La transacción es el contrato

que conecta a la estación cliente con uno o varios servidores. Es la unidad fundamental de recuperación, consistencia y concurrencia en un sistema Cliente/Servidor¹, esto implica que la transacción se inicia pero se termina o suspende en su totalidad ya que no se puede realizar o abortar parte de ellas (transacciones planas), esto asegura la integridad de la base de datos.



Fig. 2.1. Monitoreo de procesamiento de transacciones. (1)Estado actual de los datos, (2)Precompromiso, (3)Preparados, (4)Realizar actualizaciones, (5)Actualizaciones terminadas, (6)Se completa la transacción, (3.a) y (5.a) Abortar transacción.

En la figura 2.1 se tienen varios sistemas de bases de datos (SBD) que están involucrados en la transacción y en primera instancia mantienen el estado actual de la base de datos y el monitor de procesamiento de transacciones (MPT) manda una orden de precompromiso a los SBD y ellos contestan que están preparados, al ocurrir esta respuesta el MPT ya envía la orden para que se efectúen actualizaciones en cada una de las bases de datos. Si los SBD contestan que se han realizado todos los movimientos

¹ Orfali, Harkeyand, Edwards, "Computación intergaláctica cliente/servidor", Byte, Julio 1995, p. 49.

satisfactoriamente el MPT finaliza la transacción. En caso de que no estén preparados o que no se completen todas las actualizaciones en los SBD, el MPT aborta la transacción. A todo este proceso se le denomina compromiso en dos fases (two phase commit), la primera consiste en preparar a los SBD y la segunda los actualiza.

II.1.2 Funciones Distribuidas

En el modelo Cliente/Servidor las funciones distribuidas consisten en proporcionar parte de la lógica de la aplicación tanto en el cliente como en el servidor con el propósito de reducir la duplicación de funciones y dar transparencia al usuario (arquitectura de dos niveles). En la **figura 2.2** se observa una carga balanceada de la lógica de la aplicación entre el cliente y el servidor para tener un mejor rendimiento.



Fig. 2.2. Funciones distribuidas Cliente/Servidor.

Aquí el cliente se encarga de la presentación de la información y hace una solicitud de datos para que el servidor se los envíe y pueda trabajar con ellos independientemente, una vez que el cliente completa sus funciones regresa los datos al servidor para que éste los actualice.

II.1.3 Acceso a Datos Remotos

Otra alternativa para solución Cliente/Servidor es el acceso a datos remotos, donde se proporciona una conexión de los clientes hacia un servidor que esta en otra área. En la **figura 2.3** se muestra como la presentación y la lógica de la aplicación están incorporadas en el cliente para acceder a los datos almacenados en el servidor.



Fig. 2.3. Acceso a datos remotos Cliente/Servidor.

Todo el proceso se realiza en el cliente sin que se preocupe por la administración de datos ya que esta puede ser un recurso compartido. El acceso a datos remotos permite que las aplicaciones independientes trabajen con datos comunes o recursos remotos.

II.1.4 Presentación Remota

En una presentación remota Cliente/Servidor la lógica y datos de la aplicación se localizan en el servidor, mientras que la presentación esta en el cliente. **Figura 2.4**



Fig. 2.4. Presentación remota Cliente/Servidor.

Esta alternativa da flexibilidad de mantenimiento porque se tiene una administración centralizada y se pueden compartir aplicaciones para que diferentes usuarios las puedan acceder y ejecutar desde sus sistemas cliente.

II.1.5 Distribución de datos

Los mainframes de muchas compañías medianas o extensas almacenan todos los datos que maneja la organización en un sólo sitio lo cual proporciona altos niveles de control sobre la base de datos, incluyendo el control de concurrencia de usuarios, además del acceso, integridad y seguridad de los datos. Pero este acceso en un mainframe es problemático porque está limitado el número de usuarios que se pueden conectar al sistema, existe dificultad para consultar los datos almacenados en un solo lugar y un recurso centralizado compartido para muchos usuarios ocasiona tiempos de respuesta impredecibles. En la mayoría de los casos se requiere que el acceso a los datos y tiempo de respuesta del sistema sean lo más inmediato posible por lo que se adoptan tecnologías para la distribución de datos, esto es, que se diseñan aplicaciones utilizando bases de datos descentralizadas ya sean distribuidas, cuando se particiona y dispersa la base de datos por todos los sitios locales, o

replicadas cuando se hacen copias fieles de toda la base de datos y se instala en cada sitio local, según la naturaleza de estas aplicaciones.

II.2 Aumento de la capacidad y eficiencia de plataformas

Con el auge de las normas de sistemas abiertos y la conectividad entre productos de diferentes proveedores se ha permitido que distintas plataformas hardware y software puedan integrarse en un mismo ambiente y trabajar como si fuera un solo sistema. Por tal razón, para una migración a un modelo Cliente/Servidor se hace una reevaluación (rightsizing) de los recursos que se tienen para que sean aprovechados de la mejor manera posible y tener nuevas funcionalidades. En caso de que estos recursos no cumplan con las necesidades de la nueva aplicación se procede a realizar un downsizing para adquirir plataformas hardware que permitan a los sistemas cliente ejecutar además de su sistema operativo, otros más robustos sin que el Front end se vea afectado. De igual forma se deben emplear plataformas software que cumplan con las especificaciones necesarias para tener aplicaciones de mayor flexibilidad y portabilidad.

II.3 Localización de la información y de los programas de aplicación

Las alternativas Cliente/Servidor determinan la forma en que se distribuyen los datos y la aplicación entre los clientes y el servidor, además de acuerdo a como se organizan los datos en cada uno de los sitios en que son

requeridos implica un grado de dificultad para la administración, es decir, cuando se almacena la información en un solo lugar y todos los clientes accesan a ellos la administración y niveles de seguridad son más altos que cuando se tienen bases de datos distribuidas o replicadas, aunque para estas se utilizan algunas técnicas para la administración de la información.

En la siguiente tabla se resume la forma en que se distribuyen la información y los programas de aplicación en el modelo Cliente/Servidor.

Alternativa	Información	Programas de aplicación
Funciones Distribuidas	Localizada en el servidor.	Una parte en el cliente y otra en el servidor.
Acceso a datos remotos	Localizada en el servidor.	La mayor parte se localiza en el cliente.
Presentación remota	Localizada en el servidor.	La mayor parte se localiza en el servidor.
Procesamiento de Transacciones	Localizada en cada Sistema de Bases de Datos.	En el Monitor Transacciones, en los Sistemas de Bases de Datos y en los Clientes.
Bases de datos distribuidas	La porción de la base de datos correspondiente a cada sitio.	La parte correspondiente a cada sitio local.
Bases de datos replicadas	En cada sitio se tiene una copia de la base de datos.	La parte correspondiente a cada sitio local.

II.4 Comunicación entre el cliente y el servidor

La forma en que se comunican el cliente y el servidor depende de como estén organizados los procesos para cada uno, además si se tiene un ambiente multivendedor con protocolos diferentes es necesario utilizar un middleware

para que se puedan comunicar efectivamente, ya sean llamadas a procedimientos remotos, conversaciones o sistemas de paso de mensajes.

II.4.1 División de procesos internos para cliente y servidor

Existen algunos procesos internos que se ejecutan tanto en el cliente como en el servidor pero que son transparentes para el usuario y que forman parte de una misma tarea. Estos son el software de redirección del cliente, de soporte de red para ambos y para la administración en el servidor, que finalmente sirven para el mecanismo de comunicación en el modelo Cliente/Servidor, lo cual forma un aspecto muy importante por que el rendimiento de éste modelo depende de la forma en como se comuniquen ambas partes.

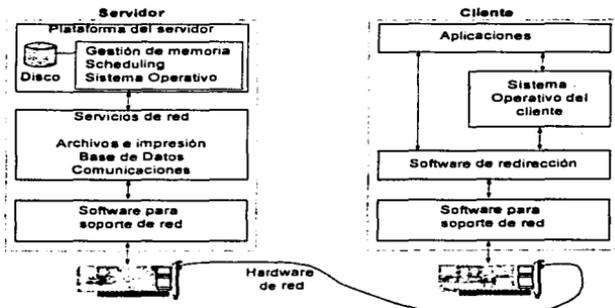


Fig. 2.5. División de procesos en el modelo Cliente/Servidor.²

² Sheldon, Tom. Enciclopedia de redes, Osborne-McGraw-Hill, México 1994, p. 449.

En la **figura 2.5** la aplicación instalada en el cliente interactúa con su mismo sistema operativo o directamente con el software de redirección para hacer sus peticiones a un servidor local o de red. El software de redirección se carga en el cliente al encender la computadora e interactúa con su mismo software para soporte de red y se comunica con el de soporte de red del servidor para poder acceder los servicios de red, estos pueden ser de archivos, impresión, bases de datos y comunicaciones. Otras funciones del servidor son la administración del sistema de archivo, de memoria y planificación de tareas de procesamiento (scheduling). También se observa como el software de soporte de red enlaza el hardware con el cable y el sistema operativo de red.

II.4.2 Llamadas a Procedimientos Remotos (RPC)

El modelo Cliente/Servidor utiliza las Llamadas a Procedimientos Remotos (RPC)* para que los procesos del cliente y el servidor se comuniquen, realicen peticiones y solicitudes de servicios, se invocan desde la GUI del cliente e integran las partes de la aplicación instaladas en cada elemento.



Fig. 2.6. Llamada a un Procedimiento Remoto.

* Remote Procedure Call

En la **figura 2.6** se observa el empleo de las RPC, por medio de la red "Un proceso de un sistema puede llamar a un procedimiento de un proceso de otro sistema"³, en este caso el proceso del cliente hace una petición al servidor para que le envíe datos o les realice algunas actualizaciones y solo le mande los resultados para que los pueda visualizar en la interfaz de usuario. Las RPC se ejecutan de manera secuencial, es decir, que el proceso que hace la llamada se detiene mientras que el procedimiento llamado en el sistema remoto retorna el resultado de la petición, posteriormente continúa la ejecución del procedimiento que hizo la llamada.

II.4.3 Conversaciones

Las conversaciones son un diálogo entre varias computadoras conectadas en red y son utilizadas en el modelo Cliente/Servidor cuando se tienen datos distribuidos en regiones geográficas distintas y es necesario realizar actualizaciones en las bases de datos en forma sincronizada.

II.4.4 Sistemas de paso de mensajes

El funcionamiento de los sistemas de paso de mensajes consiste en la formulación de un mensaje y su almacenamiento en un repositorio para que después se envíe al destino en donde se podría almacenar y procesar posteriormente. Estos sistemas son utilizados en aplicaciones que no trabajan en tiempo real puesto que permiten el intercambio de ordenes e información en modo de almacenamiento y reenvío asíncrono.

³ Deitel, H.M., *Sistemas Operativos*, Addison -Wesley-Iberoamericana, USA 1993, p. 127.

Su aplicación es práctica cuando se tienen conexiones de área extensa a bajas velocidades de transferencia o en su defecto, cuando el sistema de comunicaciones no es muy seguro.

1.5 Beneficios del modelo Cliente/Servidor

El modelo Cliente/Servidor retoma algunas características del procesamiento centralizado a diferencia de que cada elemento puede realizar funciones diferentes sobre una misma tarea, es decir, el cliente puede ejecutar algunas funciones de interfaz de usuario, generación de reportes y parte de la lógica de la aplicación, mientras que el servidor se comporta como un sistema administrador de bases de datos tradicional y realiza las funciones de definición y manipulación de datos, seguridad, respaldos y recuperación, control de concurrencia y administración de transacciones. A continuación se describen los beneficios del modelo Cliente/Servidor.

11.5.1 Costo

El costo de una red de área local basada en PC's ha propiciado que las diversas aplicaciones sean trasladadas a plataformas de más bajo costo (rightsizing y downsizing), teniendo diversas microcomputadoras conectadas a los servidores para una aplicación particular. Esto significa que resulta más barato tener una aplicación basada en un servidor y varios clientes que procesan su propias tareas, que el tener un sistema mainframe con terminales tontas.

11.5.2 Interoperabilidad

La interoperabilidad es la integración eficiente de equipo de diversos proveedores en un ambiente de funcionalidad, incluyendo interfaces comunes y una similitud en comunicaciones y aplicaciones para diferentes usuarios, es decir, que diferentes computadoras pueden trabajar juntas con facilidad. También se proporciona una plataforma sobre la cual varios proveedores pueden desarrollar aplicaciones.

11.5.3 Disponibilidad de recursos

Los recursos, sobre todo operaciones con equipo caro, se pueden localizar en un sitio donde puedan ser mejor administrados para que sean utilizados por el cliente autorizado desde cualquier parte de la red. Por lo general la administración de los recursos se hace en forma centralizada en el servidor con el objetivo de tenerlos disponibles para cada usuario autorizado.

11.5.4 Familiaridad

Algunas de las ventajas del rightsizing y downsizing es el movimiento de las aplicaciones de minicomputadoras y mainframes a sistemas de más bajo costo sobre todo en microcomputadoras que permiten a los usuarios trabajar con el software que ellos conocen en forma independiente, para que todos hagan un trabajo más rápido y productivo. Por esta razón la migración hacia un modelo Cliente/Servidor es más productivo para el usuario que actúa como cliente debido a que no le costará trabajo adaptarse a éste cambio.

II.5.5 Transparencia

Una de las características de mayor importancia de las aplicaciones Cliente/Servidor es el estilo de procesamiento como una localización de servidores en forma transparente, es decir, "Un cliente solicita un servicio de manera estándar sin importarle la localización del proveedor de este servicio"⁴. Esto significa que la localización de servidores no influye en el procesamiento de los clientes ni en las peticiones de los mismos, siempre y cuando se cuente con una buena infraestructura de comunicaciones por que en caso de que sea lenta la respuesta en la red no serán tan transparentes los procesos para el usuario.

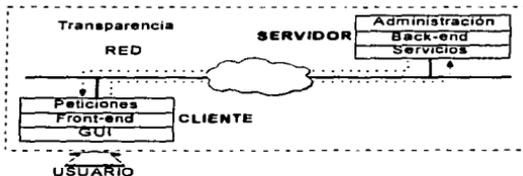


Fig. 2.7. Los servicios y procesos que proporciona el servidor son transparentes para el usuario.

En la **figura 2.7** se muestra como en el Front end, el usuario a través de la GUI hace sus peticiones al servidor y este último ejecuta algunos procesos (Back end) para satisfacer las solicitudes hechas por el cliente de manera transparente, como si todo el proceso se hiciera en la computadora del usuario.

⁴ DEC, Guide to Building Client/Server Solutions, January 1993, p.1-13.

Este beneficio del modelo Cliente/Servidor es proporcionado por el sistema operativo de la red debido a que oculta la ubicación de los recursos y servicios a las aplicaciones. La transparencia se puede dar en la ubicación de recursos y servicios, funciones de administración, de acceso asegurado y en comunicaciones.

II.5.6 Escalabilidad

Este beneficio permite a los usuarios pasar de un ambiente de trabajo desde su sistema cliente hacia el de la red. La escalabilidad se puede dar gracias a que el procesamiento de la aplicación y las funciones de bases de datos están claramente separadas en el modelo Cliente/Servidor, esto implica que existe la posibilidad de reemplazar al servidor de bases de datos Back end por un servidor idéntico que funcione en plataformas hardware y software más poderosas sin que el Front end se vea afectado.

“La escalabilidad es un simple concepto, pero tiene grandes beneficios”⁵, uno de ellos es que el Front end virtualmente no se altera y así el usuario no invierte en el software de aplicación ni en capacitación. El otro es que la escalabilidad ofrece un camino para el crecimiento de la red a bajo costo.

II.5.7 Administración centralizada

En el modelo Cliente/Servidor los administradores de sistemas pueden controlar la red a través de ambientes Netware, Unix, VMS, MS-DOS, OS/2, Macintosh o Windows y ejecutar software para el respaldo y actualización de

⁵ Distributed Computing, p. 59.

datos desde un solo lugar, además de permitir a las diferentes aplicaciones locales (clientes) el acceso a los dispositivos y recursos de la red. También en éste modelo se tiene una administración centralizada de información donde todas las bases de datos son manejadas y accesadas por un servidor central. Este es uno de los componentes claves del modelo Cliente/Servidor, sin embargo, todo el procesamiento de los datos esta centralizada en el sistema administrador de base de datos del servidor ya que es el responsable de controlar la concurrencia de datos, seguridad, integridad, respaldos, y recuperación de la misma forma que lo hacen los sistemas administradores de bases de datos en mainframes y minicomputadoras.

II.5.8 Seguridad (Autenticación y autorización)

La seguridad de la información es el factor que más preocupa a casi toda organización. En los sistemas distribuidos se corre el riesgo de que la información y las aplicaciones estén fuera de control debido a que la combinación de PC's, estaciones de trabajo y grandes equipos de cómputo de diferentes proveedores en un mismo ambiente puede poner en peligro la seguridad del cliente ó incompatibilidad entre las aplicaciones y el servidor. Por consiguiente se debe autenticar y autorizar a cada usuario que requiera de servicios o recursos. La autenticación es la capacidad que tiene el sistema servidor para identificar a cada usuario o aplicación para acceder a los recursos y servicios de la red, mientras que la autorización es la restricción que tienen los usuarios para acceder transparentemente a los recursos y servicios por medio del sistemas operativo de la red.

II.5.9 Reducción del tráfico en la red

Las aplicaciones Cliente/Servidor consisten en distribuir diversas tareas de proceso entre los clientes y el servidor. Dichas aplicaciones se conocen como Front end/Back end en donde la interfaz del cliente se separa de los comandos realizados en el servidor de manipulaciones de bases de datos y solo se transfieren los datos al Front end para mostrarlos al usuario. De esta manera se reduce el tráfico en la red porque el servidor solo envía lo que el cliente necesita para trabajar en forma independiente, liberando así la carga de trabajo.

II.5.10 Paralelismo

El trabajo en paralelo consiste en ejecutar aplicaciones al mismo tiempo en varios procesadores para reducir el tiempo requerido para realizar una tarea. Un ejemplo que ilustra las ventajas del procesamiento en paralelo en el modelo Cliente/Servidor es distribuir o replicar una base de datos para permitir su lectura o actualización en forma simultánea ya que de lo contrario si un usuario abre un archivo para leerlo otro usuario no podría hacer cambios en los datos de ese mismo archivo al instante. En cuanto más fina sea la granularidad de los datos (control de la información distribuida a nivel bloque, tabla, registro o campo) mayor será la información disponible para todos los usuarios y se incrementa el grado de paralelismo. Por su puesto que hay que utilizar herramientas sofisticadas (además no muy baratas) para poder controlar la información.

II.5.11 Flexibilidad

Este beneficio se da gracias a las herramientas SQL para el desarrollo de aplicaciones Cliente/Servidor donde se utilizan tecnologías como normalización de bases de datos, modelado de datos por medio de entidades y relaciones, manejo de objetos y herramientas CASE (Ingeniería de Software Asistida por Computadora) y así permitir un mejor mantenimiento de la aplicación para que su tiempo de vida se incremente y los ajustes se puedan hacer con una pequeña recodificación (en algunos casos ni siquiera se recodifica).

Con esto se pueden hacer versiones de las aplicaciones de manera más controladas.

II.5.12 Mayor rendimiento

El rendimiento es la integración de los beneficios explicados anteriormente, porque con el modelo Cliente/Servidor se trata de aprovechar al máximo a cada uno de los dos elementos, distribuyendo la parte de la aplicación que más se adecue a cada sistema, es decir, el servidor no se preocupa por la presentación de los datos, ni de la interfaz con el usuario, en cambio si se encarga de la administración de la base de datos y de algunos procesos que son muy pesados para que los realice un cliente, por esto es que se tiene más velocidad de procesamiento, menos tráfico en la red y una administración centralizada entre otras.

II.5.13 Independencia del Front end

En el modelo Cliente/Servidor es posible tener un servidor de base de datos de un proveedor específico que funcione como Back end y además diferentes Front ends que lo utilicen para extraer una pequeña parte de los datos y realizar las procesos en el cliente. En la **figura 2.8** se observa esta relación, donde se tiene un sistema administrador de base de datos Oracle, y como Front ends a interfaces gráficas de usuario de diferentes proveedores como son SQL Windows, Uniface, PowerBuilder y el mismo Oracle (Developers/2000). El uso de los diferentes Front ends se da cuando varios departamentos o grupos de trabajo de una misma empresa emplean las herramientas de desarrollo que más se ajustan a sus necesidades pero que requieren de la compartición e integración de datos, servicios y recursos administrados desde un solo lugar.



Fig. 2.8. Independencia del Front end en el modelo Cliente/Servidor

III.- EL AMBIENTE CLIENTE / SERVIDOR

El modelo Cliente/Servidor esta basado en un ambiente de funcionalidad que involucra la interacción de productos tanto de hardware, software y comunicaciones, es por eso que en este capítulo se describen algunos elementos indispensables del ambiente Cliente/Servidor tales como plataformas para cliente y para servidor, sistemas operativos de red y servidor, protocolos de comunicación, dispositivos de red para interconexión de sistemas de diferentes proveedores, el aspecto de las comunicaciones, el concepto de objetos distribuidos y algunos riesgos y desventajas que surgen al migrar o diseñar una aplicación Cliente/Servidor.

III.1 Plataformas para cliente y servidor

Una plataforma esta basada en el hardware y el sistema operativo para manejarla por lo que cada fabricante tiene su familia de productos que cumplen con características diferentes. En la arquitectura Cliente/Servidor se tienen plataformas tanto para el servidor como para los clientes.

Dependiendo del tipo de aplicación se puede tener una configuración Cliente/Servidor con plataformas que funcionen como servidores que pueden ser desde sistemas de escritorio, minicomputadoras hasta mainframes. Una plataforma para servidor deben de considerar: velocidad del procesador, tipo de bus, cantidad de memoria RAM, capacidad y velocidad acceso al disco duro. En la siguiente tabla se listan las plataformas para servidor identificando el fabricante del hardware y del sistema operativo que maneja.

Plataformas para servidor

Fabricante	Sistema Operativo
IBM	LanServer, OS/2, MV, AIX
Digital Equipment Corporation	VMS, ULTRIX
Hewlett Packard (HP)	HP-UX, Windows NT y 95, Novell Netware
Sun Microsystems	Unix, Solaris, Windows NT y 95, Novell Netware
Tandem	NonStop Kernel y NonStop-UX
Apple	System 7

La característica de las plataformas para cliente es que deben ser capaces de procesar y almacenar información de manera independiente, contar con Interfaz Gráfica de Usuario y ser compatibles, por lo regular son computadoras de escritorio que utilicen los sistemas operativos MS-DOS, OS/2, Windows, además de las Macintosh con su sistema operativo System 7.

III.2 Protocolos de comunicación para una red Cliente/Servidor

Las redes de computadoras necesitan del empleo de reglas para controlar y administrar la forma en que éstas se comunican, éste conjunto de reglas es llamado protocolo. El protocolo es un medio lógico que permite que los componentes de una red puedan comunicarse e intercambiar información. También con el protocolo se asegura la sincronía y dirección correcta de datos a los usuarios de la red.

III.2.1 El modelo de referencia OSI

El modelo de referencia OSI* (Interconexión de Sistemas Abiertos) fue presentado por la Organización Internacional de Normalización ISO** el cual

* Open Systems Interconnection

** International Organization for Standardization

se basa en siete capas o niveles que realizan una función en particular, para que los diferentes fabricantes de computadoras lo tomen como referencia al diseñar sus productos. De acuerdo a este modelo también se definen dos niveles de protocolos tal como se muestra en la **figura 3.1** y son los de red y alto nivel.

Los protocolos de red se encargan del proceso de la comunicación en la red donde los datos del emisor deben ser dirigidos y llegar a su destino correctamente y en sincronía. Los protocolos de alto nivel aseguran que los datos que se entregan sean reconocibles y tengan el formato apropiado.

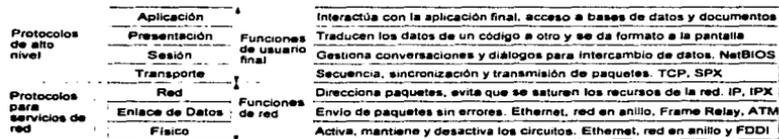


Figura 3.1. Siete niveles del modelo de referencia OSI. Los tres niveles inferiores son utilizados para servicios de red y los cuatro superiores son para protocolos de alto nivel.

III.2.2 TCP/IP (Transmission Control Protocol/Internet Protocol)

Es un conjunto de protocolos (tcp, ip, telnet, ftp, etc) estándar de los sistemas Unix que incluye uno de los protocolos de transporte. Este conjunto de protocolos esta orientado a la comunicación entre varios sistemas y de distintos vendedores. TCP asegura que los paquetes que forman el mensaje se entreguen en orden y de emisor a receptor (orientado a la conexión) sin error.

Las características del TCP son: control de flujo de paquetes para liberar la carga en la red y recuperación de errores; reconocimiento de paquetes recibidos que informa al emisor que el receptor obtuvo los datos; secuenciamiento de paquetes para verificar que estos lleguen en el momento enviado; código de paridad para detectar errores en la transmisión; retransmisión de los paquetes perdidos o erróneos.

El IP es un protocolo de comunicación sin conexión donde los paquetes se entregan desde su origen hasta su destino por el trayecto que la red tenga disponible o crea más eficiente. No hace ninguna comprobación de error, ni de control de flujo, reconocimiento ni secuenciamiento ya que algunos de estos se dejan al receptor. Proporciona un servicio de datagramas (paquetes independientes de información que contienen una dirección hacia uno o varios nodos). Soporta los servicios de FTP para recibir archivos de un cliente desde un servidor remoto, NFS que permiten a los usuarios acceder a discos independientemente de su localización física, Telnet que permite a los usuarios conectarse e iniciar una sesión con un servidor remoto, entre otros.

III.2.3 IPX (Internetwork Packet Exchange)

El IPX es el protocolo de conexión par a par propietario de Novell Netware, trabaja en el nivel de red del modelo OSI. Es un protocolo de datagramas que se transmiten en una comunicación no orientada a la conexión. Los paquetes de datos que intercambia son grupos de bits que contienen datos y una cabecera con la dirección fuente y destino, información del orden de los paquetes y gestión de errores. Los datos de un paquete pueden contener la solicitud de un servicio hecha por un cliente, una respuesta del servidor o bien

información que se transfiera de un lugar a otro. Se limita a enviar paquetes a una determinada dirección.

III.2.4 SPX (Sequenced Packet Exchange)

El SPX es un protocolo orientado a la conexión y trabaja en el nivel de transporte del modelo OSI. Al igual que el IPX es usado por Novell Netware. Este protocolo usa el intercambio secuencial de paquetes entre redes de Netware para enviar mensajes, se encarga del envío y orden del flujo de paquetes que componen el mensaje. Se usa para establecer una conexión permanente entre el cliente y el servidor para informar al instante el estado de un servidor.

III.2.5 DECnet

Es un conjunto de protocolos propietarios de DEC (Digital Equipment Corporation) y que están orientados a la integración de normas de Interconexión de Sistemas Abiertos (OSI) para implementar la Arquitectura de Red Digital (DNA, Digital Network Architecture) donde se puede dar conectividad entre sistemas DIGITAL y/o de diferentes proveedores. Los protocolos DECnet definen las especificaciones para redes Ethernet, Man, Wan y FDDI. También se puede usar en conjunto con el TCP/IP. Entre los servicios que proporciona DECnet están el DFS, Servicio de Archivos Distribuidos que permiten trabajar con los archivos por toda la red; DQS, Servicio de Colas Distribuidas para administrar y controlar los recursos de impresión de la red; RSM, Administrador de Sistemas Remotos para que un administrador gestione múltiples sistemas VAX y Ultrix distribuidos; DECdns,

Servicios de Nombres Distribuidos de DEC, para proporcionar nombres por toda la red en las aplicaciones distribuidas.

III.2.6 NetBIOS / NetBEUI

Estos dos protocolos fueron diseñados en conjunto por IBM y Microsoft para dar soporte a redes de área local. El Sistema Básico de Entrada y Salida en Red (NetBIOS, Network Basic Input Output System) es un protocolo que corresponde al nivel de sesión del modelo OSI y se encarga de establecer y mantener las sesiones de comunicación entre computadoras personales que pertenecen a una misma red de área local. Es utilizado en los entornos de LAN Manager y Windows 3.11 de Microsoft, LAN Server y OS/2 de IBM. El NetBIOS establece sesiones orientadas a la conexión además de nombres únicos para los nodos de la red para simplificar la referencia a otros sistemas. No es ruteable por lo que se tiene que empaquetar por otros protocolos para su distribución entre redes (DECnet, TCP/IP, IPX). La Interfaz Extendida de Usuarios de NetBIOS (NetBEUI, NetBIOS Extended User Interface) proporciona los servicios de transporte y encaminamiento de los datos en redes de área local. Es utilizado por Microsoft en Windows 3.11 y NT para trabajo en grupo además de su interconexión con sistemas IBM. NetBEUI le proporciona los servicios de transporte de datos al NetBIOS y les da un encaminamiento dinámico, además de ser el responsable de mensajes desde un extremo a otro de la red.

III.2.7 Conductos Nominados (Named Pipes)

Trabajan en el nivel de sesión tanto para NetWare como LAN Manager y proporcionan una interfaz entre procesos que se ejecutan entre distintas

computadoras, permitiendo el diálogo entre usuarios localizados en lugares apartados. Son más sofisticados y eficientes que el NetBIOS pero no permiten la denominación por nombres de nodos ni los servicios de datagramas sin conexión. Los conductos nominados también proporcionan servicios de conexión y desconexión, además de la administración de solicitudes de transporte de datos durante toda la sesión.

III.2.8 Apple talk

Es un grupo de protocolos que definen las conexiones entre el hardware de las computadoras Macintosh de Apple y sus periféricos con las redes de computadoras, además del sistema de cableado propio de Apple (LocalTalk) admite las redes Ethernet y Red en anillo con paso de testigo. Inicialmente fue diseñado para grupos de trabajo pequeños y locales por lo que su rendimiento no es apropiado para redes Lan grandes o Wan. La velocidad de transmisión es relativamente baja pero igualmente lo es el costo de instalación ya que no requiere de instalación de tarjetas de red ni de software para los sistemas cliente. Entre los protocolos que maneja AppleTalk están: Protocolo de Transacciones de AppleTalk (ATP), Protocolo de Asignación de Nombres (NBP), Protocolo de Eco de AppleTalk (AEP), Protocolo de Mantenimiento de la Tabla de Encaminamiento (RTMP), Protocolo de Sesión AppleTalk (ASP), Protocolo de Flujo de Datos de AppleTalk (ADSP), Protocolo de Interconexión de Zona (ZIP) y el Protocolo de Acceso a la Impresora (PAP).

De los protocolos mencionados anteriormente se puede decir que los más utilizados en la actualidad son TCP/IP debido a su facilidad para la detección y corrección de errores y su empleo en los sistemas que trabajan con plataforma

UNIX siendo éste el sistema operativo más popular en cuanto a la interconexión de sistemas abiertos. También otro de los sistemas operativos de red más utilizados son el Netware de Novell que utiliza los protocolos de comunicación IPX/SPX basados en el protocolo XNS, además del TCP/IP. Netware utiliza la tecnología de protocolos abiertos para trabajar con plataformas DOS, OS/2, Macintosh y UNIX desde la misma red.

OSI	SNA	DNA	TCP/IP	NetWare	AppleTalk	LAN Manager
Aplicación	Servicios de transacción	Aplicación DNA	FTP, Protocolo Sencillo de Transmisión de Correo (SMTP)	Protocolo principal de Netware	AppleShare	Bloques de mensajes del servidor
Presentación	Servicios de presentación	Control de sesión de la DNA	NFS, Servicio de Nombres de Dominio (DNS)		Protocolo de Clasificación AppleTalk (AFP)	
Sesión	Control del flujo de datos		Transporte	TELNET	Conductos nombrados y NetBIOS	ASP ADSP ZIP PAP
Transporte		Control de transmisión	TCP, Protocolo de Datagramas de Usuario (ARP)	SPX	ATP NBP AEP RTMP	NetBEUI
Red	Control de trayecto	Red	IP, Protocolo de resolución de dirección (ARP)	IPX	Protocolo de Distribución de Datagramas (DDP)	
Enlace de datos	Control de enlace de datos	Enlace de datos	Controladores LAN		Controladores LAN	Controladores LAN
			ODI	NDIS	LocalTalk EtherTalk TokenTalk	NDIS
Físico	Control físico	Físico		Físico	Físico	Físico

Figura 3.2 Conjuntos de protocolos comparados con el modelo de referencia OSI¹.

En la figura 3.2 se muestra una comparación de los protocolos con el modelo de referencia OSI con la finalidad de identificar sus similitudes y diferencias y tomándolo como base para la interoperabilidad entre redes que

¹ Sheldom, Reiss, Curdin. Guía de Interoperabilidad. Osborne McGraw Hill, México 1995, p. 98.

utilizan diferentes protocolos con la ayuda de algunos dispositivos de red como gateways y ruteadores para realizar la conversión de protocolos.

III.3 Sistemas operativos de red y servidor para el modelo Cliente/Servidor

Los sistemas operativos Cliente/Servidor están diseñados de tal forma que pueden acceder archivos de otros sistemas operativos. A continuación se describen algunos sistemas operativos de red para la computación Cliente/Servidor.

Novell Netware

El sistema operativo Novell Netware esta orientado a la estrategia de computación en red permitiendo la interconexión de cualquier tipo de sistema operativo y plataformas de trabajo ideales para procesamiento distribuido Cliente/Servidor donde las estaciones de trabajo comparten una tarea de procesamiento entre un cliente y un servidor. Su característica principal es que consta de una serie de módulos cargables (NML) y descargables para optimizar los recursos consumidos.

Netware utiliza una Tecnología de Protocolos Abiertos (OPT) como solución para utilizar varios protocolos en una red, para cubrir la necesidad de interconectar componentes de una red que funciona bajo sistemas operativos diferentes. Uno de los aspectos a cubrir es la Independencia de protocolo Cliente/Servidor que permite tener una interfaz de enlace de datos abierta para que diferentes protocolos como IPX/SPX, Apple Talk, TCP/IP, SNA y los modelos OSI puedan cargarse o liberarse en el servidor de la red cuando se

requiera. Siendo un servidor Netware los clientes soportados son los sistemas con DOS, Windows, los OS/2, Macintosh de Apple y con sistema operativo Unix a futuro.

UNIX

En la actualidad UNIX es el Sistema Operativo más utilizados a nivel mundial tanto en la industria como instituciones porque ofrece un ambiente de trabajo para el desarrollo de programas y el procesamiento de textos, permite la combinación de programas facilitando un enfoque modular, además porque es una de las tendencias de estandarización como sistema operativo abierto. "Unix es el sistema operativo que se ha instalado en todo tipo de computadoras desde las micro hasta las supercomputadoras y también es adoptado por una gran cantidad de fabricantes importantes de computadoras"² sacando al mercado variantes del sistema operativo Unix como son: Solaris de Sunsoft, Unixware de Novell, A/UX de Apple, AIX de IBM, HP-UX de HP, ULTRIX de Digital, NonStop-UX de Tandem, etc. Con él los usuarios de computadoras personales pueden usar un sistema operativo multiusuario y multitareas con almacenamiento virtual manejando microprocesadores de 32 y 64 bits.

Los Sistemas UNIX también permiten transportar las aplicaciones de muchos desarrolladores de aplicaciones (entre ellas Cliente/Servidor) y sistemas CASE avanzados. También ofrece la capacidad de interacción con el MS-DOS para que los usuarios puedan ejecutar la plataforma de aplicaciones para IBM PC además de la plataforma UNIX, donde en las últimas

² Deitel, H.M., Sistemas Operativos, Addison-Wesley Iberoamericana, USA 1993, p.14.

generaciones ya se ha adoptado el concepto de Interfaz Gráfica de Usuario (GUI) para facilitar su uso y manejo.

El sistema operativo UNIX está compuesto de una serie de sistemas distintos como lo son diferentes tipos de Shell y editores que lo componen, ofreciendo una variedad de alternativas para edición y ejecución de programas para la administración de recursos o para el desarrollo de aplicaciones. Para el modelo Cliente/Servidor también es el sistema operativo más utilizado debido al impacto que han tenido la UNIX International (UI) y la Organización de Software Abierto (OSF) para la conectividad entre diferentes sistemas operativos, por lo que UNIX se ha convertido en la plataforma preferida que representa el enlace entre diferentes redes de computadoras, macrocomputadoras, minis, estaciones de trabajo, computadoras personales y supercomputadoras, ya que los sistemas UNIX tienen la capacidad de unificar la industria en una Norma abarcando una gran cantidad de sistemas de cómputo y de comunicaciones. Así mismo el empleo de protocolos de comunicación para la interconectividad de sistemas se ha basado en los protocolos TCP/IP, IPX/SPX.

Open VMS

Open VMS es un sistema operativo multiusuario de propósito general que soporta las series de computadoras VAX y Alpha de Digital tanto en ambientes de desarrollo como en producción. Puede ejecutar una amplia variedad de aplicaciones incluyendo las de misión crítica, OLTP Cliente/Servidor, cómputo intensivo, I/O intensiva, procesamiento en tiempo real y la combinación de estos y otros ambientes. El rendimiento de este

sistema operativo depende del tipo de computadora, memoria física disponible y el número y tipo de discos y controladores de cinta en el sistema.

Open VMS puede trabajar como integrador de redes, en procesamiento distribuido, multiprocesamiento y tiene capacidad de manejo de ventanas. Contiene amplias características que permiten un uso fácil, incrementar la productividad de los programadores y facilita la administración del sistema. Este sistema operativo es un ambiente de software abierto que soporta los estándares clave tales como OSF/Motif, POSIX, XPG3 y el Ambiente de Computación Distribuida (DCE, Distributed Computing Environment).

WINDOWS NT

Es un sistema operativo para servidor de Microsoft de 32 bits el cual cuenta con un conjunto de servicios para trabajo en red para sustituir a LanManager.

La característica principal de Windows NT es su sistema de seguridad de archivos y protección contra fallas, puede ejecutar aplicaciones MS-DOS, Windows 3.x, POSIX y de OS/2 además de tener conectividad con Netware de Novell, Vines de Banyan, LanManager, UNIX, VMS y redes SNA. Se puede correr bajo los procesadores 80386, 80486, Pentium, MIPS R4000 de 64 bits de Dec. Combina la potencia de capacidad de ejecución del procesador, transferencia de datos y de acceso a memoria para obtener un buen rendimiento.

Windows NT permite procesamiento multitarea, protección de memoria, procesamiento simétrico (aprovechar múltiples procesadores), duplicación

automática en discos secundarios. Utiliza memoria virtual además de tener conectividad con NetBIOS, NetBEUI, TCP/IP, IPX, SPX. Maneja jerarquías de usuarios y de seguridad. Su interfaz gráfica es idéntica a la de Windows 3.x. las aplicaciones de 16 bits no se ejecutan con un buen rendimiento, así como las aplicaciones de Windows 3.x, no usa la tecnología de conectar y usar (plug and play).

III.4 Dispositivos de red para facilitar el procesamiento distribuido

La ampliación e interconexión de redes locales es un factor fundamental para el modelo Cliente/Servidor debido a que se pueden añadir clientes con diferentes plataformas y distintos protocolos.

III.4.1 Repetidores

Se utilizan en el nivel físico de los protocolos y son empleados en las redes Ethernet, Token Ring y ARCNET. En esencia es una pequeña caja con una conexión de entrada que recibe la señal del cable, ésta la amplifica para transmitirla por medio de la conexión de salida.

III.4.2 Puentes

Funcionan en el nivel de enlace y permiten conectar distintos tipos de redes (**figura 3.3.b**) y también por medio de ellos se puede dividir una red en dos ramas separadas para aumentar su rendimiento para evitar los cuellos de botella cuando hay demasiados clientes conectados a la red. Con ello una red se carga con un número de usuarios reducido (**figura 3.3.a**).



Figura 3.3.a. División de una red grande en otras dos separadas.

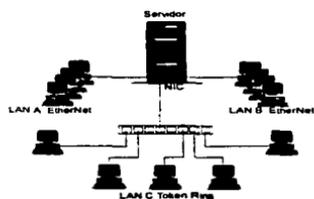


Figura 3.3.b. Conexión de distintos tipos de redes.

III.4.3 Ruteadores

Funcionan en el nivel de red y se utilizan para interconectar redes Lan y Wan y proporcionan control de tráfico y filtrado de funciones. Dirigen los paquetes por las rutas mas eficientes o económicas entre las redes interconectadas. Su función principal es conmutar los paquetes por medio de una tabla de ruteo en base a su dirección para enviarlo a su destino directamente o lo manda a otro ruteador para hacerlo llegar a su destino, figura 3.4.

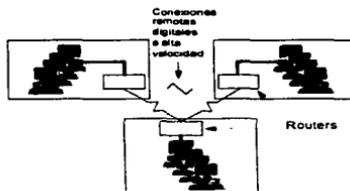


Figura 3.4. Configuración con tres Lan's separadas y ruteadores

III.4.4 Gateways

Conocidos como pasarelas, puertas de enlace ó compuertas funcionan en los niveles más altos de la jerarquía de protocolos, se utilizan cuando se tienen interconectados sistemas con diferentes protocolos, arquitecturas, o formatos de estructuras de datos. Esto significa que un gateway es un punto de conexión y un traductor entre dos tipos de protocolos ya que modifica los paquetes de información o su sintaxis para que el sistema destino lo puede entender. La ventaja es que al conectar un sistema a la Lan con el gateway, los usuarios de cualquier cliente pueden acceder al sistema, **figura 3.5.**

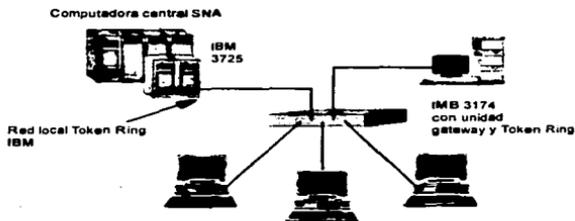


Figura 3.5. Conexión de un sistema central IBM SNA desde una red local IBM Token Ring. Los Gateways pueden conectar sistemas que utilizan distintos sistemas operativos y protocolos de comunicación.

III.4.5 Concentradores

Se utilizan para interconectar entre sí a las computadoras de una red bajo un esquema de cableado configurado en estrella. Es un bus que acepta módulos Ethernet, FDDI, Red en anillo con paso de testigo. Facilita la

administración de la red. Algunos ya traen incorporadas características de gestión, puenteo y encadenamiento.

Todos éstos dispositivos de red tienen un funcionamiento correspondiente a cada nivel del modelo de referencia OSI y sirve para la conexión de sistemas de diferentes proveedores, en la **figura 3.6** se muestra en que nivel trabaja cada uno de los dispositivos.



Figura 3.6. Relación de los dispositivos de red con el modelo OSI.

III.6 Objetos distribuidos

El procesamiento con objetos distribuidos es el resultado de la combinación de las tecnologías orientadas a objetos y Cliente/Servidor, permite a los objetos distribuirse a través de una red heterogénea donde todos los componentes pueden interoperar entre sí como si fueran una sola entidad. Los objetos interactúan utilizando el paso de mensajes que representan solicitudes de información o servicios, además los objetos pueden asumir la categoría de clientes y servidores. El elemento que liga físicamente a los

objetos es el Gestor de Peticiones de Objetos (ORB, Object Request Broker) que proporciona un camino para que los objetos localicen y activen a otros objetos en una red sin importar el procesador o lenguaje de programación utilizado para generar los objetos tanto para el cliente como para el servidor.

La tecnología de objetos distribuidos esta regida por normas y una de las organizaciones encargada de definir estándares requeridos para los sistemas de objetos distribuidos en ambientes heterogéneos es el Grupo de de Administración de Objetos (OMG, Object Management Group), quien desde 1989 ha especificado constantemente la arquitectura de un bus de software abierto en el que los objetos desarrollados por diversos fabricantes puedan interoperar a través de redes y sistemas operativos.

Posteriormente la OMG definió la Arquitectura para Administración de Objetos (OMA, Object Management Architecture) la cual incluye cuatro series de estándares: la Arquitectura Genérica de para Gestión de Peticiones de Objetos (CORBA, Common Object Request Broker Architecture), Servicio de Especificación de Objetos Comunes (COSS, Common Object Service Specification), Facilidades Comunes (CF, Common Facilities) y Objetos de Aplicación (AO, Application Objects). A finales de 1994, OMG aprobó una arquitectura estándar llamada CORBA 2.0 que define los servicios a ser suministrados por un ORB basadas en TCP/IP.

Las especificaciones de interfaz entre un componente y el bus de objetos (este bus proporciona un ORB que permite que los clientes invoquen métodos en objetos remotos) de la OMG están escritas en un Lenguaje de Definición de Interfaz (IDL, Interface Definition Language) que es

independiente a cualquier lenguaje de programación. Estos componentes especifican en IDL los tipos de servicios que proporcionan, los métodos, atributos, parámetros, administradores de errores y relaciones hereditarias con otros componentes, de esta forma el IDL enlaza al cliente con los componentes del servidor.

Con el IDL se pueden encapsular los datos y aplicaciones para distribuirlos y manejarlos como objetos. "ORB es el middleware que utiliza la computación con objetos distribuidos para interoperar en redes heterogéneas de objetos"³, con esto el ORB proporciona una manera para localizar, activar y conectar entre si de manera transparente a los usuarios y desarrolladores.

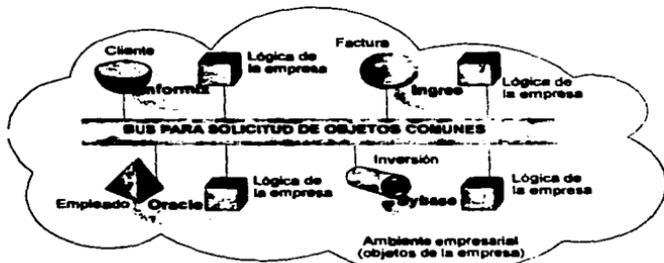


Figura 3.7. En el ambiente de objetos distribuidos los datos y aplicaciones se encapsulan en objetos mediante el IDL para que puedan navegar en toda la red en diferentes plataformas e interactuar con objetos de otros fabricantes.

³ Distributed Objects for business, <http://www.sun.com/sunworldonline/swol-04-1996/sow1-04-ooobook.html>, Abril 1996, pág. 18.

El concepto de computación con objetos distribuidos se puede ver como una simple red global de clientes y servidores heterogéneos o como objetos corporativos del negocio (**Figura 3.7**). El encapsulamiento de la aplicación en forma de objetos es una técnica para crear la interfaz orientada a objetos para acceder a una o más aplicaciones. En este ambiente los objetos solo conocen qué servicios son proporcionados por otros objetos (sus interfaces) pero no como se los proporcionan (su implementación). De esta manera los objetos aparecen para los usuarios y desarrolladores como objetos que son familiares para su negocio no como máquinas, redes ni lenguajes de programación, es decir, ellos no tienen que pensar en términos de tecnologías, solo en términos de objetos que son familiares para su negocio.

El procesamiento con objetos distribuidos es un ambiente muy complejo por lo tanto requiere una infraestructura tecnológica robusta y muy sofisticada.

III.7 Infraestructura de comunicaciones en el modelo Cliente/Servidor

Quizá el aspecto más importante del ambiente Cliente/Servidor es su infraestructura de comunicaciones debido a que la transparencia de procesos entre el cliente y el servidor dependen totalmente de ella, además de brindar una mejor seguridad, rendimiento y facilidad de crecimiento. También se requiere de la integración de herramientas y aplicaciones que intervienen en el ambiente Cliente/Servidor como si fueran una sola sin afectar las peticiones de los usuarios. Esta infraestructura de comunicaciones puede estar formada por una o varias tecnologías tales como: Cableado estructurado, FDDI, Ethernet rápida, Frame Relay y ATM.

III.7.1 El cableado estructurado

Es una infraestructura ideal para el modelo Cliente/Servidor por que es un sistema de cableado prefabricado que esta diseñado para dar soporte a las reconfiguraciones y crecimiento de la red, ya sea FDDI, Ethernet rápida, Frame Relay o ATM, de esta manera proporciona un sistema de cableado uniforme que admite productos y elementos de diferentes vendedores. Es bastante flexible para el crecimiento de la red. Esta basado en la norma EIA/TIA 586 (Electronic Industries Association/Telecommunications Industries Association).

III.7.2 FDDI

La Interfaz de Datos Distribuidos por Fibra (FDDI, Fiber Distributed Data Interface) es una norma de cable para fibra óptica, transmite datos a 100 Mbps en una topología en anillo doble que admite 500 nodos distribuidos a una distancia de 100 Km. El arreglo en anillo doble produce una redundancia en los sistemas de transmisión para proteger al sistema de los cortes o defectos en el cableado, tiene un diseño antigiratorio de anillo para mantener activas las conexiones si se rompe el anillo (autoreconfiguración con bucle de retorno). Los datos se transmiten mediante pulsos de luz a través de un medio de cristal puro o plástico. No presenta distorsión ni diafonía (ruido en la línea ocasionado por fuga de señal en el cable par trenzado) además de atenuación muy reducida. Todas las estaciones comparten el mismo conducto de comunicaciones y al agregar más clientes disminuye el ancho de banda, además es muy caro, pero transmite a velocidades muy altas, con seguridad, adaptabilidad y cobertura amplia.

III.7.3 Ethernet rápida

Surge para cubrir las limitaciones que se presentan en los sistemas de trabajo en red Ethernet, es flexible y con tolerancia a fallas. Soporta distintas topologías y normas de cableado (coaxial, par trenzado desde 185 a 500 m, fibra óptica hasta 4 km). Existen dos variaciones de la Ethernet rápida: Ethernet 100Base-X y Ethernet 100VG-AnyLAN. La primera trabaja en una topología en estrella con par trenzado y Acceso Múltiple con Detección de Portadora/Detección de Colisiones (CSMA/CD, Carrier Sense Multiple Access/Collision Detected) escalable y es compatible sobre cable de categoría 5 y ejecutarse en cable de 4 pares, alcanza una velocidad de 100 Mbps. La segunda elimina la sobrecarga de la red y admite más tráfico, ya no usa CSMA/CD ahora usa el método de acceso por prioridad de demanda dejando al concentrador la capacidad para tomar decisiones, estas prioridades dependen de lo que se transmita (en tiempo real y calidad). Utiliza cableado de 4 pares y tiene la capacidad de enviar y recibir simultáneamente hasta una cobertura de 200 m. Trabaja con gran seguridad debido a que el concentrador 100VG-AnyLAN transmite paquetes solo por puertos conectados a la dirección de destino del paquete (privacidad).

III.7.4 El Frame Relay (Retransmisión de Tramas)

Es un método de comunicación que trabaja en base a la conmutación de paquetes, admite paquetes de longitud variable permitiendo a las aplicaciones utilizar el tamaño de paquete adecuado para tener un rendimiento óptimo. Se puede usar en redes Lan y Wan sobre redes públicas o privadas. Utiliza la retransmisión de tramas para el transporte en red Wan y puede transmitir datos

hasta una velocidad de 140 Mbps con un ancho de banda entre 56 Kbps y 1.544 Mbps. Los ruteadores y puentes juegan un papel importante en la conexión de la red hacia las comunicaciones Frame Relay.

III.7.5 ATM

La tecnología de Modo de Transferencia Asíncrono (ATM, Asynchronous Transfer Mode) se utiliza en redes Lan y Wan, proporciona una alta velocidad de transmisión de varios tipos de datos (voz, datos, video en tiempo real, sonido con alta fidelidad, imágenes) aprovechando al máximo los cables de fibra óptica. El rango de alta velocidad ATM oscila entre 155Mbps a 622 Mbps. Los paquetes de datos que utiliza son pequeños y de tamaño constante lo cual permite reducir el tipo de retardo en los conmutadores de la red. Tiene una tasa de errores binarios de 10^{-12} lo cual aumenta la seguridad de transmisión en conexiones físicas libres de errores.

III.8 Algunos riesgos y desventajas del ambiente Cliente/Servidor

Definitivamente las aplicaciones Cliente/Servidor son mucho más complejas que las tradicionales basadas en computadoras centrales con terminales o sistemas standalone debido a que exigen un nivel de rendimiento más elevado de la red, por lo tanto son muy difíciles de administrar.

También son difíciles de construir, principalmente en la etapa de análisis y diseño donde se debe identificar perfectamente los procesos que se van a ejecutar como Back end y los Front ends (distribución de procesos), si no se realiza a conciencia, no se podrá explotar al máximo la capacidad tanto de los

clientes como del servidor y ocasionar una sobrecarga en la red. Cuando la aplicación es bastante crítica se incrementa la urgencia del proyecto y se trabaja bajo presión.

Una actualización del personal mal planeada puede ocasionar que el desarrollo, administración y mantenimiento de la aplicación Cliente/Servidor caiga en manos de unos cuantos especialistas o bien en consultores externos.

Estas aplicaciones son difíciles de depurar ya que un cambio puede implicar introducirse en productos de diferentes proveedores (sistemas operativos, protocolos, middleware, bases de datos) por lo que una depuración mal planeada provocará serios problemas y más aún cuando se tienen bases de datos distribuidas o replicadas.

Otro riesgo es el que la tecnología emergente ocasiona una curva de aprendizaje prolongada para los desarrolladores.

Aunque se dice que Cliente/Servidor aumenta los tiempos de respuesta y disminuye el tráfico en la red, el incremento a un gran número de usuarios que utilizan simultáneamente procesos en línea puede producir un fuerte impacto en el rendimiento en los tiempos de respuesta.

Otra desventaja son los ambientes heterogéneos que se vuelven complejos y difíciles de integrar.

Las herramientas para desarrollo de aplicaciones Cliente/Servidor, productos de conectividad, infraestructura de comunicaciones, capacitación del personal, la contratación de servicios de consultoría o desarrollo tienen un costo elevado, por lo que Cliente/Servidor es bastante caro.

Una alternativa para enfrentar los riesgos y desventajas que surgen en el ambiente Cliente/Servidor es el empleo de un plan para probar la calidad de la aplicación, haciendo una variedad de pruebas, simulación y monitoreo. Entre ellas están: la prueba de la GUI del cliente para verificar que acceden los datos y procesos del servidor de manera transparente; análisis e inspección del middleware para que los desarrolladores se aseguren de que sus aplicaciones se comuniquen con la base de datos; carga del cliente (verificar programas residentes, controladores cargados, varias aplicaciones en memoria) para sincronizar las transacciones distribuidas y recolección de resultados fuera de línea; la carga en la plataforma del servidor y verificar que los tiempos de respuesta en las transacciones sean aceptables; la carga en el ambiente Cliente/Servidor, que es una medición del rendimiento y funcionalidad del sistema completo validando tanto al cliente como al servidor identificando el hardware para el servidor, la red o los factores que ocasionan cuellos de botella; plataformas en las que se ejecuta la aplicación (mainframes, minicomputadoras, microcomputadoras con procesador pentium); y por último, ejecutar algunos procesos de prueba de administración de la aplicación.

IV.- DISTRIBUCIÓN DE DATOS EN SISTEMAS CLIENTE / SERVIDOR

Como se mencionó en el capítulo II una de las alternativas Cliente/Servidor es la distribución de datos que consiste endescentralizar la información de redes Wan y dispersarla en varios sitios geográficos diferentes con la finalidad de tener un acceso a los datos de la manera más conveniente para el usuario, ya sea que solo tenga la información necesaria para realizar sus procesos y después integrarla con los demás sitios o bien tener una copia exacta de información en la que se requiere que los movimientos de cada lugar sean en toda la base de datos. Es por esto que se utilizan dos técnicas para la distribución de datos: las bases de datos distribuidas y bases de datos replicadas. Su impacto en el modelo Cliente/Servidor es trascendental porque el procesamiento de las aplicaciones se distribuye entre varias computadoras cliente y uno o varios servidores, estas pueden estar en sitios geográficos diferentes y utilizar distintos sistemas operativos, plataformas hardware o software y fuentes de datos.

IV.1 Tecnología de bases de datos distribuidas

Cuando se tiene una base de datos bastante grande en una red Wan y en algunos sitios el manejo de cierta información es mucho más constante que en otro y además es necesario que el procesamiento de la información sea al instante, se divide la base de datos en fragmentos y se destina cada uno de

ellos al sitio correspondiente, teniendo de esta manera una base de datos distribuida.

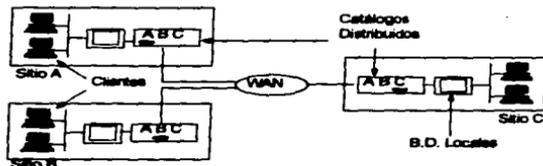


Fig. 4.1 Esquema de Base de Datos Distribuida¹.

En la **figura 4.1** se representa el esquema de bases de datos distribuida en donde los usuarios locales de cada sitio (A, B ó C) cuentan con su porción de la base de datos para tener un acceso inmediato, en caso de requerir información de otro sitio, el Sistema Manejador de Base de Datos Distribuidas (SMBDD) debe proporcionar transparencia para que el sistema funcione como si fuera un solo sitio, a menos de que hubieran problemas de retardo en la red. Este SMBDD es una colección de bases de datos independientes localizadas físicamente en diferentes servidores, enlazados por una Lan grande o por una Wan para que los usuarios y programas la vean como una sola base de datos lógica.

IV.1.1 Distribución de datos

El factor importante para crear una base de datos distribuida es que solamente de deben distribuir los datos por motivos de un mayor rendimiento

¹ García Kanecay, Juan, ¿Bases de datos distribuidas o replicadas?, PC Semanal, 6 de Dic. 1993, p 44.

y sobre todo reducción del uso de las comunicaciones en redes Wan y consecuentemente la obtención de una mayor confiabilidad. Así la velocidad del acceso a los datos aumenta considerablemente y se trabaja de manera transparente. En la **figura 4.2** se observa como se hace la distribución de datos para formar una base de datos distribuida donde este sistema resuelve las limitaciones en el número de usuarios que en el esquema centralizado puede manejar y se divide una gran base de datos en pequeños fragmentos

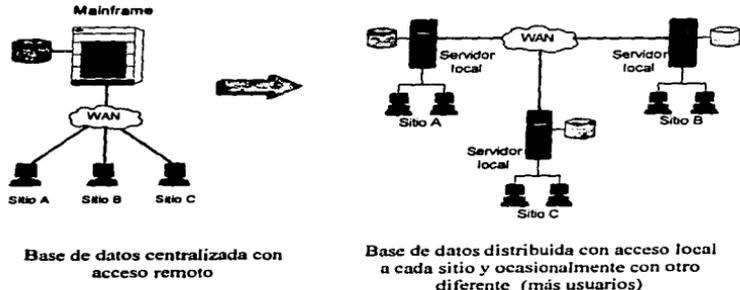


Fig. 4.2 Distribución de datos.

Aquí solo se utiliza la Wan cuando algún sitio requiere hacer consultas de datos de otros sitios (que será ocasionalmente) y no a cada instante como en el centralizado donde habría más problemas de tráfico en la red. La distribución de datos se puede hacer mediante una fragmentación de la base de datos en forma horizontal, vertical o mixta, utilizando sentencias *sql* para definir

expresiones que hagan la fragmentación pero unidas por las relaciones globales. En la **figura 4.3** se muestra un esquema de distribución de datos utilizando varios tipos de fragmentación.

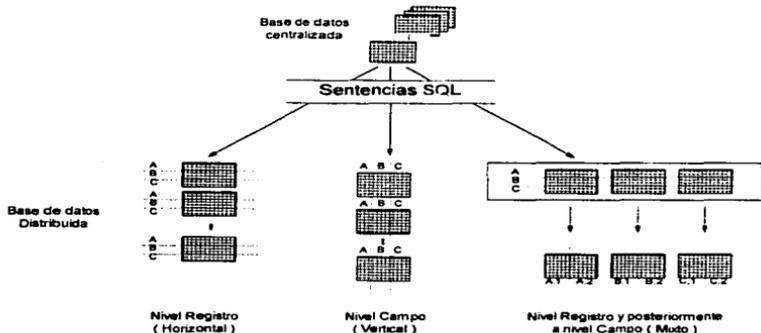


Fig. 4.3 Tipos de fragmentación para bases de datos distribuidas.

La fragmentación horizontal se hace agrupando la información común para cada sitio local (por registros) y se conserva la misma estructura de la base de datos en cada uno. En cambio la fragmentación vertical altera la estructura de la base de datos tomando los campos que se involucran en cada sitio local relacionados entre sí. La fragmentación mixta hace agrupaciones de los registros comunes para cada sitio y a su vez estos se vuelven a agrupar por campos para la información que maneje cada sub-sitio local. Una propiedad importante que deben guardar los fragmentos es que a partir de ellos se pueda

reconstruir la base de datos original sin provocar pérdida o duplicidad de información.

IV.1.2 Bases de datos lógicas

El concepto de las bases de datos lógicas tiene una estrecha relación con el de transparencia. Por lo que una base de datos lógica es el empleo de todas las porciones de bases de datos distribuidas como si fuera una sola, el usuario puede realizar consultas, actualizaciones, vistas y reportes sin que tenga conocimiento donde están localizados físicamente los datos (siempre y cuando se tenga una buena infraestructura de comunicaciones), tal como se ve en la figura 4.4.

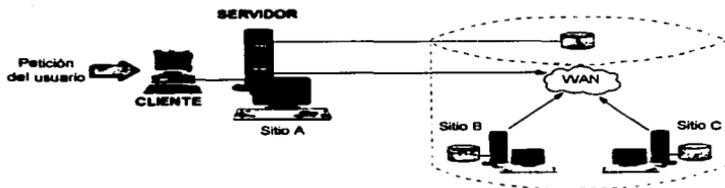


Fig. 4.4 En una base de datos distribuida el usuario ve los datos de cada sitio como una sola base de datos.

El funcionamiento de las bases de datos distribuidas no es fácil, requiere de una buena administración por parte del manejador de bases de datos, en el que se debe mantener la seguridad, integridad y consistencia de los datos así como la disponibilidad cuando sean requeridos por otros sitios locales. En las bases de datos distribuidas se tiene un compromiso de transparencia en dos

sentidos: del lugar y operativa. La transparencia del lugar es la que se acaba de explicar donde las aplicaciones accesan a la información en diferentes lugares sin preocuparse por su localización. La transparencia operativa consiste en realizar en un ambiente distribuido los procesos que se ejecutan en un entorno local.

IV.1.3 El compromiso en dos fases (two-phase commit)

En los esquemas de bases de datos distribuidas las aplicaciones que las manejan requieren de actualizar varias bases de datos y a su vez mantener una sincronización entre los sitios involucrados. El método que utilizan para mantener tal sincronización es el compromiso en dos fases (two phase commit). Se aplica para finalizar el proceso de una transacción y consiste en la actualización física de la información de las bases de datos en dos fases con el objetivo de evitar errores en caso de que se presenten problemas en la Wan.

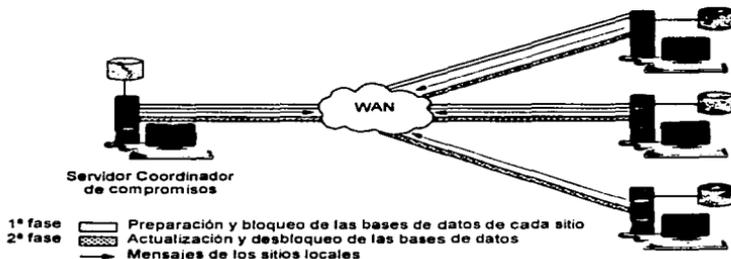


Fig. 4.5 Compromiso en dos fases (two phase commit), actualiza todas o ninguna de las bases de datos.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Para que las bases de datos puedan efectuar la sincronización en cada lugar es necesario un coordinador de compromisos para garantizar la integridad y seguridad en la base de datos, éste bien podría ser un monitor de procesamiento de transacciones, **figura 4.5.**

En esta figura se observa el funcionamiento del compromiso en dos fases, cuando el procesamiento de una transacción esta por finalizar el coordinador de compromisos manda mensajes de solicitud de compromiso a todos los servidores de bases de datos locales para que envíen su mensaje de que están preparados, en caso de que todos lo estén, cada servidor local bloquea la base de datos (puede ser toda la base, por tablas, bloques o páginas o por registros, dependiendo de la granularidad del sistema) y el coordinador vuelve a enviar mensajes de confirmación del compromiso a los servidores locales donde se efectuará la actualización de las bases de datos y desbloquearlas. Si por algún motivo en cualquiera de las dos fases por lo menos un servidor local no esta listo el coordinador envía mensajes de anulación de compromiso (roll back) lo que significa que la transacción se ejecuta completa actualizando todas las bases de datos o se aborta no actualizando ninguna.

IV.1.4 Ventajas y desventajas de las bases de datos distribuidas

La ventaja principal de las bases de datos distribuidas es que el usuario tiene acceso rápido a los datos que solo él necesita en su sitio local y en algunas ocasiones hará uso de los enlaces Wan para acceder datos de otros sitios. También el empleo del compromiso en dos fases da un grado de integridad y seguridad en los datos debido a la sincronización de los servidores de cada base de datos involucrados en la transacción. Otra ventaja es el que se

incrementa el número de usuarios que puedan procesar información independientemente de manera local. También las aplicaciones que utilizan las bases de datos distribuidas asocian la diferente información de cada sitio por medio de relaciones.

De las desventajas de las bases de datos distribuidas es la gran cantidad de mensajes que se envían entre el coordinador de compromisos y los servidores de bases de datos locales donde el tiempo de bloqueo de cada base de datos puede durar un tiempo considerable inhabilitando esta parte a cada sitio. En caso de no terminar la transacción en el compromiso en dos fases se tiene que generar de nuevo el proceso de actualización. Cada elemento del sistema distribuido está sujeto a fallas, "si el nodo coordinador del compromiso en dos fases falla, todos los datos participantes pueden permanecer bloqueados hasta que el nodo se restablezca"². Cuando existen pocos campos o registros que actualizar en una base de datos local se estaría desperdiciando un costo de procesamiento ya que influiría en los tiempos de respuesta para los demás sitios.

IV.2 Tecnología de bases de datos replicadas

Cuando se tiene una base de datos en una red Wan y es necesario que todos los procesos (consultas, ordenaciones, reportes, vistas, actualizaciones, etc.) se hagan al momento, se procede a realizar una réplica de la base de datos en cada sitio local, y de esta manera se tendrá una base de datos replicada. En

² Sun Microsystems Corporation, *Managing a Distributed Environment*, USA 1994, pág. 8.

la **figura 4.6** se representa el esquema de base de datos replicada de cada uno de los sitios de la red Wan donde en el sitio C esta localizada la base de datos primaria con el servidor de réplica, la base de datos de los sitios A y B son secundarias.

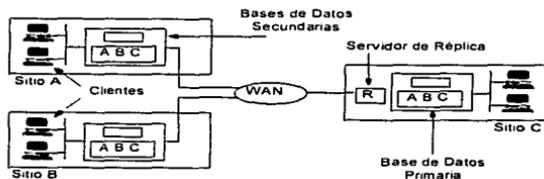


Fig. 4.6 Esquema de Base de Datos Replicada³.

La replicación se puede hacer en Sincronía donde las actualizaciones de las bases de datos se hacen al mismo tiempo, o de manera Asíncrona donde la actualización se hace a intervalos de tiempo.

IV.2.1 Replicación de datos

La replicación de datos es necesaria cuando se requiere que los usuarios accedan a toda la información localmente en lugar de hacerlo de manera remota utilizando conexiones a una red Wan.

Este esquema de bases de datos replicadas consta de una base de datos primaria la que por medio de su servidor es replicada a los sitios remotos y por lo tanto pasarán a ser bases de datos secundarias, **figura 4.7**.

³ García Kaneezay, Juan, ¿Bases de datos distribuidas o replicadas?, PC Semanal, 6 de Dic. 1993, p.44.

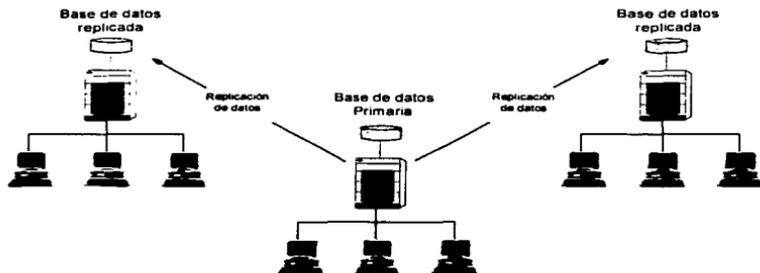


Fig. 4.7. En una base de datos replicada, las copias de los datos se ponen en los sitios remotos.

Aquí una transacción actualiza en primer instancia a la base de datos primaria, posteriormente la información es copiada o replicada a las bases de datos secundarias ya que para los usuarios locales es de solo lectura. Esta replica se puede realizar de manera inmediata (síncrona) o en intervalos de tiempo a lo largo de un día o en la noche (asíncrona).

En caso de que una base de datos secundaria no este disponible el proceso de réplica se efectúa en cuanto se restablezca sin que esto afecte al funcionamiento de las otras bases de datos secundarias como se muestra en la figura 4.8.

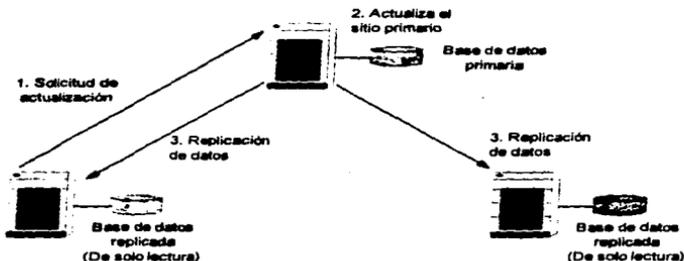


Fig. 4.8. Actualización sólo en el sitio primario y replicación en los secundarios.

IV.2.2 Replicación síncrona

Se utiliza cuando al momento de ejecutar una transacción la información se debe escribir simultáneamente en la base de datos primaria y en las secundarias. Para esto se utiliza el compromiso en dos fases explicado en la sección de bases de datos distribuidas en la que se requiere de una sincronización de los datos.

Tanto en las bases de datos distribuidas como en las replicadas se comporta de la misma forma: incluye bloqueo de datos en cada sitio y por consiguiente un retardo para acceder a la información, la actualización de todas o ninguna de las bases de datos en cada sitio, el bloqueo de todas las bases de datos participantes en una transacción en caso de que el nodo coordinador o supervisor del compromiso en dos fases falle.

IV.2.3 Replicación asíncrona

Cuando se tienen aplicaciones en las que el proceso de actualización en cada sitio no es tan crítico o que los usuarios remotos no necesitan la nueva información inmediatamente se procede a realizar una replicación asíncrona, en la que la actualización o replicación de cada sitio se efectuará en intervalos periódicos y en cada hora determinada cuando los enlaces Wan estén más libres. Aquí una falla en cualquiera de los servidores locales no afecta el funcionamiento de los demás y su replicación se puede hacer hasta que se restablezca.

Entre los métodos de actualización asíncrona en las bases de datos replicadas se encuentran los “snapshots” y los “triggers” (activadores o disparadores). La implementación de snapshots permite a la distribución asíncrona efectuar actualizaciones en las tablas individuales, subconjuntos de tablas o sobre colecciones de tablas de acuerdo a un horario o programa determinado. Pero los snapshots no mantienen la integridad de las transacciones y proporcionan copias de solo lectura que no pueden ser actualizadas por los sitios que las accesan. Con el uso de los snapshots mientras los datos se copian a múltiples sitios, las copias permanecen de solo lectura y no pueden realizar ningún cambio en las datos distribuidos.

Los triggers son otro mecanismo de actualización asíncrona donde al efectuarse un cambio en alguna parte de los datos se activa una alarma dentro de la base de datos y posteriormente activa un código específico de replicación en la base de datos la cual inicia el proceso de replicación. Este mecanismo protege la integridad referencial de los datos de la empresa.

IV.2.4 Replicación en cascada

Una de las variantes de las bases de datos replicadas es la replicación en cascada donde la base de datos primaria (en la que solamente se pueden efectuar las actualizaciones) replica los datos a un segundo sitio y este posteriormente se encarga de replicarlos en los demás, **figura 4.9**.

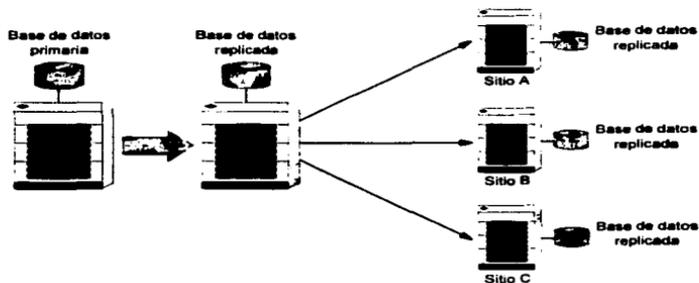


Fig. 4.9, Replicación en cascada.

En esta figura la base de datos primaria además de funcionar como un sistema de respaldo sede los privilegios a otro servidor de base de datos que esté más próximo a los sitios locales para reducir los costos de comunicaciones.

IV.2.5 Consolidación de información remota

Es un caso excepcional en las bases de datos replicadas donde los diferentes sitios manejan su propia información y por las necesidades de la empresa se requiere tener la consolidación de todos los sitios en un lugar central. Para esto cada sitio remoto funciona como un maestro o sitio primario, mientras que el sistema central que integra los datos funciona como un esclavo.

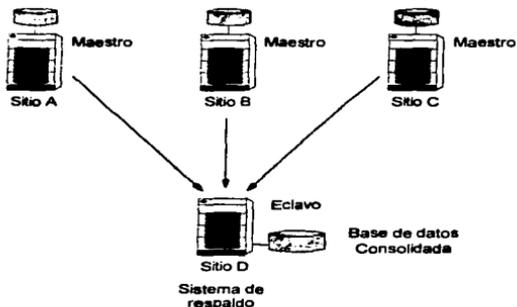


Fig. 4.10, Consolidación (integración) de información remota.

En la **figura 4.10** se puede observar que mientras cada sitio maestro tiene sus propios datos existe un sistema esclavo que entraría en funcionamiento en caso de un desastre. Además el sistema central se puede utilizar para el soporte de decisiones y reportes de integración.

IV.2.6 Ventajas y desventajas de las bases de datos replicadas

La ventaja principal de las bases de datos replicadas es la disponibilidad de cada uno de los servidores locales de manera independiente, es decir, si al momento de la replicación o actualización alguno falla, este proceso se realiza en los demás servidores sin que se vea afectado su funcionamiento mientras el otro se restablece.

Otra de las ventajas es que se puede seleccionar uno de los dos sistemas de replicación dependiendo de la naturaleza de las aplicaciones y las necesidades de la empresa: el sistema de replicación síncrona (inmediata) en la que se requiere una infraestructura de comunicaciones con un buen rendimiento, o asíncrona (a intervalos de tiempo determinados) en la que se aprovecha la capacidad del sistema de comunicaciones cuando no se tenga una carga pesada de tráfico en la red.

De las desventajas es la responsabilidad tan grande que tiene el servidor de la base de datos primaria debido a que todas las actualizaciones que hagan las bases de datos secundarias deben de realizarse primero en la base de datos primaria y posteriormente su replica a todas las secundarias, en caso de que el servidor de la base de datos primaria no este disponible no se podrá realizar la replicación en las secundarias.

Otra desventaja es cuando se tiene un sistema de replicación asíncrona si por algún motivo se requiere de información actualizada no se podrá obtener hasta que pase el intervalo de tiempo establecido.

IV.2 Comparación de bases de datos distribuidas y replicadas

El surgimiento tanto de las bases de datos distribuidas como el de las replicadas se originó de la misma necesidad, la distribución de información por diferentes sitios geográficos para realizar procesamiento distribuido y dar un acceso inmediato a la información que requieren los usuarios de cada lugar. Pero existe una característica fundamental que las hace diferentes, la estructura para cada una es distinta. Mientras que para una base de datos distribuida con fragmentación horizontal se altera la estructura de la base de datos particionandola en registros para cada sitio, en las replicadas se copia exactamente la misma estructura a cada uno con la diferencia de que una es primaria y las demás son secundarias, tal como se muestra en la figura 4.11.



Fig. 4.11 La estructura de la base de datos distribuida con fragmentación horizontal es diferente para cada sitio, mientras que en las replicadas es igual.

El proceso de actualización para los dos tipos de bases de datos es caro y complicado, están relacionadas totalmente con el rendimiento de las comunicaciones porque dependiendo de su velocidad y eficiencia será el grado de transparencia para el usuario en el procesamiento distribuido.

La actualización en ambos tipos de base de datos es diferente por que en las bases de datos distribuidas si alguna de ellas no se encuentra disponible la transacción no se efectúa y regresa al estado anterior de los datos, pero en las bases de datos replicadas primero se actualiza la base de datos primaria y posteriormente las secundarias, si alguna de ellas no esta lista actualiza a las demás sin afectar su funcionamiento.

Cuando es necesaria una sincronización entre las bases de datos en una actualización se utiliza el compromiso en dos fases, en la distribuida se actualizan los sitios implicados pero en la replicada ésta se efectúa en el sitio primario y posteriormente en todos los secundarios. Aunque también es necesario recalcar que se presentan las mismas ventajas y desventajas de la actualización sincrónica para ambos tipos de bases de datos.

El proceso de actualización en las bases de datos replicadas tiene otra variante, la replicación asíncrona la cual ayuda a contrarrestar los problemas de la replicación sincrónica debido a que se puede realizar en intervalos de tiempo preestablecidos utilizando triggers y/o snapshots.

V.- MÉTODOS Y HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES CLIENTE/SERVIDOR

Otro de los aspectos fundamentales del modelo Cliente/Servidor es el empleo de métodos y herramientas adecuadas para el desarrollo de aplicaciones, porque dependiendo de ellos es como se aprovecha la arquitectura y comunicaciones en sistemas Cliente/Servidor. En este capítulo se describen algunas metodologías tradicionales como son el análisis y diseño estructurado, desarrollo de prototipos, además del uso de herramientas CASE para modelado de datos, normalización de bases de datos y metodologías orientadas a objetos, así como el funcionamiento de herramientas front-end y back-end.

V.1 Metodologías para el desarrollo de aplicaciones Cliente/Servidor

V.1.1 Diseño estructurado

El diseño estructurado es uno de los métodos tradicionales más usados para el desarrollo de sistemas de información. Tal desarrollo comprende dos etapas íntimamente relacionadas que son el análisis y el diseño. El Análisis se refiere al proceso de entender en su totalidad al sistema y determinar si es posible la utilización de la computadora para mejorarlo ó para modificar el sistema actual, mientras que el diseño es el proceso de planificar, reemplazar o completar un sistema organizacional existente. La metodología que el diseño estructurado propone es la división del sistema en módulos para la construcción de un modelo que le permita crear programas formados por éstos

módulos independientes unos de otros y así poder dar un fácil mantenimiento a los programas cuando sea necesario debido a que se programan de manera estructurada.

Los elementos del diseño estructurado son: Diagrama de Flujo de Datos (DFD), Diagrama de Estructura de Datos (DED) y Diccionario de Datos.

Diagrama de Flujo de Datos (DFD).

El DFD es un modelo gráfico del sistema que identifica los elementos básicos de los procesos, el flujo de datos, el lugar donde se almacenarán los datos y las fuentes y destinos. Los elementos de un DFD se muestran en la **figura 5.1**.

Nombre	Símbolo	Función que realiza
Burbuja		Representa a los procesos o transformaciones de los datos que entran a él.
Flujo de datos		Indica el lugar de donde provienen los datos y hacia donde se dirigen (flujo de los datos).
Almacenador ó archivo		Son almacenes de datos en los cuales se puede leer, actualizar, guardar información.
Fuente ó Destino		Es la fuente que proporciona los datos o el destino de los mismos

Figura 5.1. Elementos que componen un Diagrama de Flujo de Datos.

A su vez cada proceso puede desglosarse como otro DFD hasta llegar a la unidad mínima de detalle para comprender en su totalidad al sistema.

Diagrama de Estructura de Datos (DED)

Un DED revela la estructura modular de un sistema (por ejemplo su partición en módulos) y la jerarquía de sus módulos. Los elementos que requiere el DED se ilustran en la **figura 5.2**.

Nombre	Símbolo	Función que realiza
Módulo		Representa la operación que se realiza a los datos ya sea de recepción (Eferentes), entrega (Aferentes) o de Transformación
Conexión		Une por medio de las interfaces a los módulos relacionados a su conexión
Parámetro o Dato		Representa a los datos, de donde provienen y a donde se dirigen

Figura 5.2. Elementos que componen un Diagrama de Estructura de Datos.

Diccionario de Datos

Los Diccionarios de Datos ayudan al analista a comprender el sistema, en ellos se realizan todas las definiciones y descripción detallada de los elementos del sistema tales como procesos, flujos y almacenes de datos. Contienen la definición del nombre de los datos, descripción particular de un flujo de datos o la transformación que sufre el dato en el proceso.

V.1.2 Diseño de prototipos

En éste método la participación del usuario es muy importante para el análisis y diseño, ya que ellos evalúan el diseño y la información que se genera en el sistema por que un prototipo está construido por software que acepta entradas, realiza procesos, produce información impresa o en pantalla, o bien lleva acabo actividades complementarias significativas para la empresa. El

prototipo es el modelo original (prueba o piloto), es la primera versión o iteración de la aplicación. En la **figura 5.3** se muestra un modelo que ejemplifica lo que es el desarrollo de un prototipo.

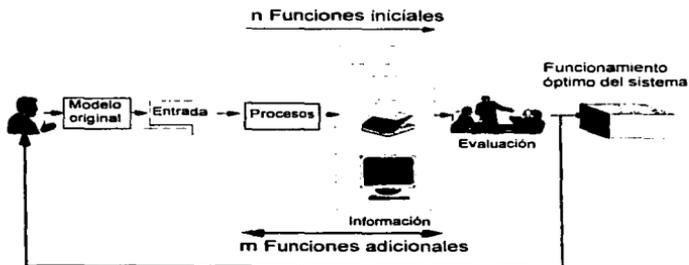


Figura 5.3. Modelo del desarrollo por prototipo.

Justificaciones para el desarrollo de prototipos

- 1.- Los requerimientos de información no siempre están bien definidos.
- 2.- Permiten evaluar situaciones donde los encargados de diseñar e implantar sistemas no tienen experiencia en alguno similar.
- 3.- Existen situaciones de riesgo y costo elevado.
- 4.- El diseño propuesto es novedoso y aún no ha sido probado.
- 5.- Como es un sistema iterativo cada vez se añaden o quitan funciones o requerimientos que el mismo sistema debe satisfacer.

La metodología para el desarrollo de prototipos y algunas herramientas (SQLWindows, Uniface, PowerBuilder) están orientados a construir sistemas prototipo con rapidez (en días o semanas). Utilizan un bajo costo a

comparación de un sistema convencional y funcionan casi con la misma eficiencia. Los métodos consisten en desarrollar prototipos con no todas las características y toques finales, se pueden tomar partes de otros programas, de otros sistemas o se utilizan librerías de código reusable.

V.1.3 Técnicas Orientadas a Objetos

Conforme los sistemas de información se fueron haciendo más grandes y complejos, las técnicas de análisis y diseño tradicionales se van volviendo más difíciles de aplicar por lo que es necesario el uso de otra técnica para el desarrollo de aplicaciones: la Orientada a Objetos.

Esta técnica involucra un cambio radical en la manera de diseñar ya que los analistas en vez de pensar en procesos y descomposición, piensan en objetos y su comportamiento e involucra una combinación con herramientas CASE, generadores de código, programación visual y desarrollo basado en repositorios. Un análisis orientado a objetos es más natural, fácil y poderoso, además de ser independiente de la herramienta de desarrollo.

La técnica se basa en un análisis y diseño orientado a objetos y se divide en dos modelos relacionados entre sí: el de la Estructura y el de Comportamiento del Objeto, **figura 5.4**.

Los modelos de objetos se representan por diagramas denominados esquemas, donde los **esquemas de objetos** muestran las estructuras del objeto (base de datos) y los **esquemas de eventos** lo que le ocurre a los objetos (procesos).

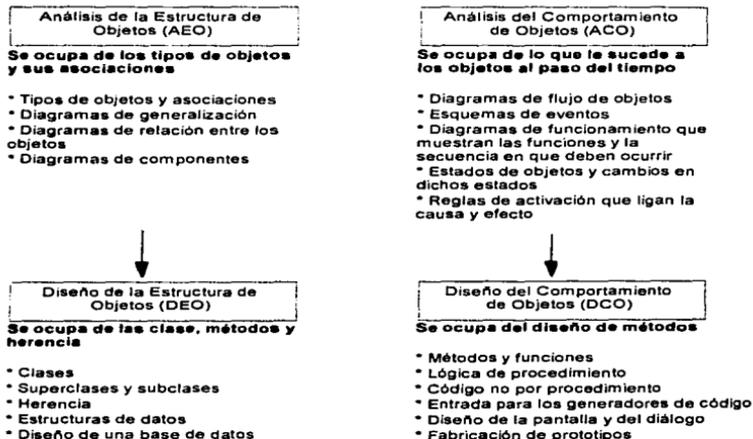


Figura 5.4 Análisis y Diseño de la estructura y comportamiento del objeto¹.

Las tecnologías Orientadas a Objetos proporcionan muchos beneficios, algunos de ellos son:

- Construcción de clases complejas a partir de otras ya existentes
- Confiabilidad, diseño rápido y calidad
- Integridad en la petición de datos
- Modelo más realista
- Bibliotecas de clases
- Migración
- Independencia del diseño
- Interacción de software de varios proveedores
- Facilidad de mantenimiento

¹ Martin J.-Odell J. Análisis y diseño orientado a objetos. Ed. Prentice Hall H., Edo. México 1994, pág. 80

- Programación más sencilla
- Objetos Binarios de Gran Tamaño (BLOB)
- Computación Cliente/Servidor, Distribuida masiva y paralela
- Reutilización de clases

V.1.4 Modelado de Datos

Una vez que se ha hecho el análisis de los requerimientos de la empresa se procede a diseñar un modelo de los datos. Dentro del diseño de una aplicación de bases de datos Cliente/Servidor el modelado ayuda a tener una representación abstracta o simbólica del sistema que se pretende desarrollar. El modelo permite visualizar la estructura de los datos y demuestra como se refleja con exactitud y en su totalidad el problema del sistema de información. Esto se debe a que contiene todas la unidades de información relevantes, íntegra, consistente y global. Antes de comenzar a construir la aplicación hay que tomar en cuenta que resulta menos caro hacer un cambio en el diseño que en la codificación, de ahí la importancia del modelado porque aquí es más fácil realizar los ajustes necesarios.

Otros de los beneficios del modelado de datos son la abstracción, transparencia y efectividad que dan al diseño. La **abstracción** consiste en tener un modelo económico y flexible para crear y dar mantenimiento utilizando pocos símbolos en la representación. La **transparencia** involucra que las ideas plasmadas en el modelo deben ser claras y entendibles. El modelo es **efectivo** cuando ayuda a tener un buen rendimiento y adaptación a los nuevos requerimientos y procesos de la empresa sin que se tenga que reconstruir.

V.1.5 Diagramas de Entidad-Relación

En el modelado de datos existen diagramas que sirven como lenguaje gráfico para representar las relaciones entre los elementos de la información, estos son los Diagramas de Entidad-Relación (DER). Uno de los objetivos del DER es simplificar el diseño y uso de bases de datos. Un diagrama de entidad Relación es una técnica gráfica que describe el modelo empresarial como una colección de ENTIDADES, RELACIÓN entre las entidades y los ATRIBUTOS asociados a las entidades y relaciones. En la notación estándar de Entidad-Relación las entidades se representan por rectángulos, las relaciones entre entidades por rombos y los atributos se representan por óvalos, figura 5.5.

Entidad	
Relación	
Atributo	
Cardinalidad de la relación	1:1=Uno a uno 1:N=Uno a muchos M:N=Muchos a muchos

Figura 5.5. Elementos que componen un Diagrama de Entidad Relación.

Entidades, Atributos y Relaciones.

“Las **entidades** son una representación abstracta u objetos físicos del mundo real”², es decir, una entidad puede ser desde una persona, lugar, evento, concepto o cualquier cosa que en una base de datos sirve para almacenar

² Kemper, Moerkotte. Object-Oriented data base management. Ed. Prentice Hall, New Jersey 1994, pág. 15

información (una tabla). Se debe identificar a una entidad mediante un sustantivo para que se pueda manipular con facilidad con el SQL de la base de datos. Ejemplo de entidades son: EMPLEADO, CLIENTE, FACTURA, etc.

Los **atributos** son las características o propiedades que distinguen a las entidades entre sí. Existen atributos llave (**clave primaria, PK**) y atributos no llave. La clave primaria es la que identifica de manera única a un elemento de la entidad (**instancia**), los atributos no llave son los que dependen de la llave primaria de la entidad. Es por medio de la llave primaria que se pueden establecer las relaciones entre las entidades, una llave primaria puede estar construida por uno o varios atributos.

Las **relaciones** sirven para enlazar, conectar o asociar dos o mas entidades que tienen algún parentesco en el mundo real. La relación puede tener una cardinalidad de Uno a Uno (1:1), de Uno a Muchos (1:N) o de Muchos a Muchos (M:N). Las relaciones se identifican mediante verbos, por ejemplo, se tiene dos entidades: INGENIERO y PRODUCTO:

- 1:1 Un INGENIERO <diseña> un PRODUCTO
- 1:N Un INGENIERO <diseña> uno o varios PRODUCTOS
- M:N Uno o varios INGENIEROS <diseñan> uno o varios PRODUCTOS

A continuación se muestra un ejemplo de DER en su notación estándar (**figura 5.6.a**) y el utilizado por algunas herramientas comerciales (**figura 5.6.b**).

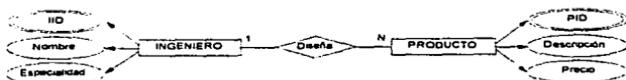


Figura 5.6.a. Ejemplos de DER



Figura 5.6.b. Ejemplos de DER

En la **figura 5.6.b** la llave primaria de la entidad padre INGENIERO se hereda a la entidad hijo PRODUCTO como una **llave foránea, FK**. El diagrama de DER incluye otros conceptos como **Agregación, Generalización o Clasificación y Asociación**.

V.1.6 Normalización de bases de datos

Dentro del desarrollo de las aplicaciones Cliente/Servidor de bases de datos es indispensable utilizar un método de Normalización de datos para controlar y eliminar la redundancia en el almacenamiento de los datos que es uno de los beneficios del modelo Cliente/Servidor. Por tal motivo la Normalización es muy importante para un buen diseño de bases de datos. En la **figura 5.7** se ilustran las características que debe tener un buen diseño de bases de datos producto de la Normalización.

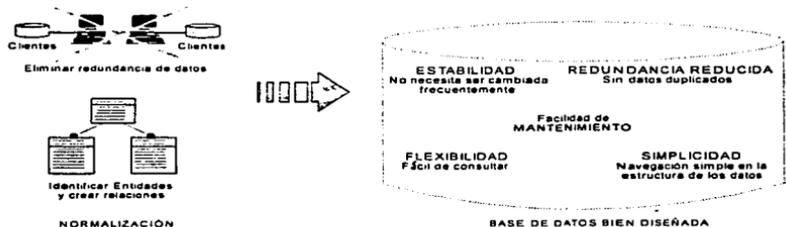


Figura 5.7. Características de una Base de datos Normalizada

Como podemos observar el proceso de Normalización elimina la redundancia (no controlada) de datos tratando de que varios usuarios no generen ni almacenen la misma información en lugares diferentes, además la identificación de llaves primarias en las entidades y la creación de relaciones donde cada atributo es funcionalmente dependiente de la llave que los relaciona, ayuda a realizar una base de datos que sea estable, con redundancia reducida, flexible, con simplicidad y facilidad de mantenimiento. Para realizar la Normalización de una base de datos es necesario tener un conjunto de tablas no normalizadas y realizar una serie de pasos de “descomposición” de la estructura de los datos para desarrollar el esquema final de la base de datos, los pasos son los siguientes:

Forma Normal Cero (0NF). Inicialmente los datos no están normalizados por lo tanto se dice que están en la Forma Normal Cero.

Primera Forma Normal (1NF). Cuando todas los registros tienen un identificador único (clave primaria), además cada atributo es atómico (son indivisibles) y no hay ninguna estructura en los campos.

Segunda Forma Normal (2NF). Cuando se cumple la 1NF y si la clave primaria es una clave compuesta (superclave: grupo de uno o más atributos que considerados conjuntamente permiten identificar de manera única a una entidad en el conjunto de entidades) entonces cualquier dependencia funcional esta en la clave primaria.

Tercera Forma Normal (3NF). Cuando se tiene la 2NF y además no existen dependencias transitivas y no se almacenan datos calculados o derivados.

Cuarta Forma Normal (4NF). Cuando se tiene la 3NF y no existen dependencias con múltiples valores.

V.2 Uso de herramientas CASE para el desarrollo de aplicaciones Cliente/Servidor

En el desarrollo tradicional de sistemas un cambio en el análisis de la aplicación ya en producción o en las etapas finales indica que no se hizo un buen análisis y por lo tanto involucra un gran costo efectuar tales cambios. Además el problema de la documentación de muchos de los desarrolladores esta latente, debido a que se preocupan más por terminar la aplicación y pocas veces realizan de manera paralela su documentación. El empleo de la Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering, CASE) para el desarrollo de aplicaciones Cliente/Servidor ha producido un cambio importante en el desarrollo tradicional y un ahorro en tiempo y costo. Las herramientas CASE ayudan a realizar desarrollos bajo estándares, son más poderosas y ayudan a la documentación, asegurando de esta manera una mayor calidad del software, además de auxiliar en el desarrollo de prototipos y generación de código con documentación. Existen herramientas CASE para modelado (ER-Win, Designer/2000 de Oracle, AME de Uniface) y otros para desarrollo (Developers/2000 de Oracle, ER-Win) que ayudan a los programadores a reducir el tiempo de desarrollo y producir aplicaciones más consistentes. Con las herramientas CASE a partir del modelo de una aplicación, ésta se puede convertir al código que se requiera, ya sea Oracle, Sybase, SQL Base, Visual Basic, Informix, etc., o realizar el proceso inverso, que a partir del código se genere el modelado de datos (Ingeniería de

reversa). Algunos de los beneficios de las herramientas CASE son los siguientes:

- 1.- Mejoran la velocidad y disminuyen el tiempo necesario para completar la tarea de desarrollo.
- 2.- Automatizan las tareas tediosas.
- 3.- Garantizan la consistencia de procedimientos (Generación de código por medio de reglas consistentes y exactas).
- 4.- Implementación de Ingeniería de reversa (Reingeniería).

V.3 Ejemplos de herramientas sql para el desarrollo de aplicaciones Cliente/Servidor

Algo que resulta muy importante en la generación de sistemas es la disponibilidad de herramientas de desarrollo de aplicaciones de manera rápida y eficiente, que permitan un acceso fácil y cómodo a la información, además de manejarla con flexibilidad y seguridad. Para ello existe una gran cantidad de este tipo de herramientas que pueden trabajar en todas las plataformas hardware y software. Muchas de ellas se pueden conectar utilizando un tipo de middleware que les permita interactuar entre sí, sin importar el hardware, sistema operativo o tipo de red. Algunas de estas herramientas están más orientadas al front end y otras al back end en los sistemas Cliente/Servidor.

V.3.1 Herramientas Front end

Son las encargadas de desarrollar la GUI para el sistema cliente y por medio de ellas se tiene acceso a la base de datos. De entre la gran variedad de éste tipo de herramientas solo se explicarán SQLWindows, Uniface y PowerBuilder.

SQLWindows

SQLWindows es una herramienta para desarrollar aplicaciones que sirven de interfaz para bases de datos SQL, sistemas de correo electrónico y Monitores de procesamiento de transacciones, corre bajo dos ambientes: MS Windows y OS/2 Presentation Manager. Proporciona un lenguaje de programación 4GL totalmente funcional llamado Lenguaje de Aplicación SqlWindows (SAL), incluye mas de 500 funciones construidas que dan flexibilidad y facilidad para el desarrollo.

Entre sus modalidades tiene tres formas de trabajar: utilizando *QuickObjects* que es una alternativa para desarrollar rápidamente aplicaciones mediante el uso de objetos visuales, inclusive los pueden utilizar para generar prototipos, *QuickForms* es una manera rápida de crear aplicaciones utilizando QuickObjects trabajando de manera gráfica sin necesidad de escribir código (se genera solo) y el *SAL*.

TeamWindows

Cuando se hacen desarrollos en equipos de trabajo SqlWindows contiene un componente llamado TeamWindows que tiene como núcleo un Repositorio Abierto y centralizado. Proporciona una estructura para ensamblar componentes reusables de la aplicación y ayuda a construir aplicaciones más robustas y fáciles de mantener. Facilita el control de versiones y la seguridad en el desarrollo y control del ciclo de vida de la aplicación. SqlWindows contiene un Depurador que ayuda a identificar puntos de ruptura entre variables, expresiones y mensajes.

Conectividad de SqlWindows

Gupta tiene una familia de productos de conectividad llamada SQLNetwork para que las aplicaciones que la utilizan puedan manejar un amplio rango de bases de datos sql. Utiliza los SQLRouters para conectarse a Oracle, SqlServer, Informix, Ingres y AS/400 y el ODBC para bases de datos pequeñas o de escritorio (Dbase, Paradox, entre otras).

Gupta incluye una variedad de interfaces abiertas para la integración de Fuentes de datos que no son sql como Lotus Notes, MS Mail, herramientas CASE, además de una copia de la base de datos SQLBase para desarrollo, pruebas y afinaciones locales.

Sistemas de Bases de datos soportados por SQLWindows

Conectividad Nativa con SQLNetwork

- Gupta SQLBase Server
- IBM DB2
- IBM AS/400
- IBM OS/2 Database Manager
- Oracle
- Sybase SQL Server
- MicroSoft SQL Server
- Informix
- Ingres
- HP ALLBASE/SQL

Conectividad con ODBC

- dBASE
- Paradox
- Oracle
- Rdb
- AS/400
- DB/2
- y otras

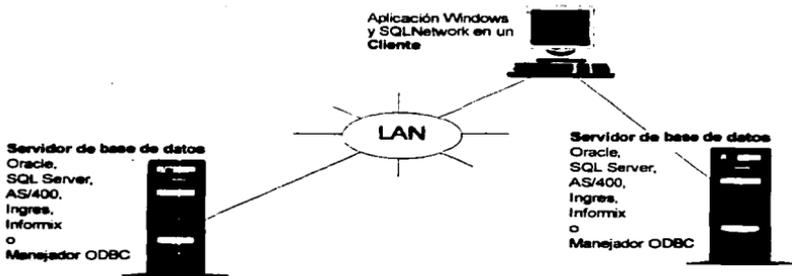


Figura V.8. *SQLNetwork* utiliza software de comunicaciones de diferentes vendedores de bases de datos para acceder a los servidores a través de una LAN o mediante una conexión directa³.

UNIFACE

Es una herramienta que integra cuatro elementos fundamentales para el desarrollo de aplicaciones Cliente/Servidor como son: Desarrollo a base de modelo, ambiente gráfico de objetos, distribución independiente de tecnologías y desarrollo en equipos. Contiene un lenguaje de cuarta generación llamado Proc y es utilizado para generar triggers que son activados cuando ocurre algún evento. A continuación se describen los elementos que integran el ambiente de desarrollo de aplicaciones Uniface.

Application Model Manager (Manejador del Modelo de la Aplicación)

Permite crear y dar mantenimiento a modelos de la aplicación mediante una interfaz para editar los componentes y tener acceso rápido a todos los

³ Gupta, *SQLNetwork: Guide to Gupta Connectivity*, USA 1994, pág. 1-3

objetos y sus propiedades. Permite crear objetos globales y modelos de procesos. El Editor de Modelo de la Aplicación (AME) ayuda a crear modelos de entidad relación.

Rapid Application Builder (Creador de Aplicaciones Rápidas)

Para diseñar formas y reportes, contiene un examinador de clase para identificar objetos y recorrer sus propiedades. Incluye undepurador visual y un servicio para desarrollar prototipos denominado fast form.

Deployment Manager (Manejador de Distribución)

Distribuye las aplicaciones Uniface haciéndolas totalmente independientes de la tecnología de múltiples plataformas hardware y software, sistema operativo, herramientas CASE, DBMS e Interfaz de Usuario.

Developer Service (Servicios para Desarrolladores)

Para ofrecer un control de versiones a nivel de objetos y aplicación, sobre todo en trabajo en equipo.

Personal Series (Serie Personal)

Son herramientas que permiten a los usuarios finales hacer consultas y reportes gráficos por toda la información de la empresa.

Application Objects Repository (Repositorio de Objetos de la Aplicación)

El repositorio de objetos es el núcleo de Uniface, es multiusuario, abierto, y esta documentado, estos objetos son reusables y están disponibles en cualquier fase y parte de la aplicación, además de dar soporte al modelado.

Configuración Cliente/Servidor con Uniface

Uniface utiliza POLYSERVER como estándar para proporcionar acceso simultáneo a múltiples DBMS corriendo en plataformas heterogéneas.

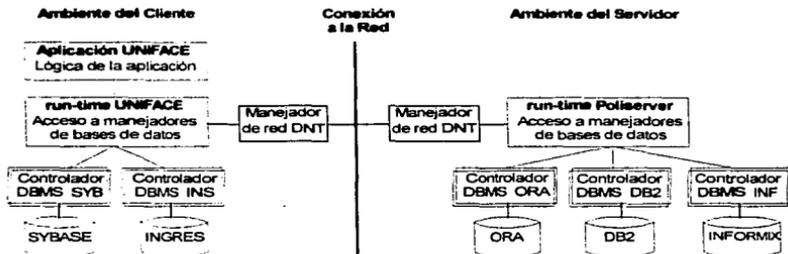


Figura V.9. Configuración Cliente/Servidor con Uniface, utilizando Polyserver⁴.

PowerBuilder

Es una herramienta propietaria de PowerSoft y esta diseñada para construir aplicaciones complejas y de alto rendimiento tanto para Cliente/Servidor como para Internet, proporcionando métodos orientados a objetos y código compilado. Las aplicaciones en PowerBuilder pueden desarrollarse y ejecutarse en plataformas como Windows (3.x, 95 y NT), Macintosh y Sun Solaris. Con PowerBuilder se puede implementar arquitecturas Cliente/Servidor para dos y tres niveles.

⁴ Uniface. UNIFACE SIX: Environment specific guide, Compuware Corporation 1995, pág. 1-17

PowerBuilder utiliza el concepto de Ambiente Abierto de Desarrollo Cliente/Servidor (CODE) para dar a los clientes la libertad de seleccionar los productos y tecnologías emergentes que más se ajusten al procesamiento distribuido o utilizando llamadas a librerías de enlace externas. Ofrece una manera rápida para desarrollar aplicaciones distribuidas sofisticadas utilizando objetos no visuales.

Se pueden distribuir objetos a través de la red para incrementar la escalabilidad y manejo centralizado.

Estos objetos distribuidos también se pueden compilar en objetos de usuario no visuales y grabar como objetos Proxy o alias los cuales almacenan toda la información necesaria para distribuir la aplicación en un servidor remoto.

Objetos Remotos PowerBuilder

Un objeto remoto contiene la lógica de la empresa escrita como métodos PowerScripts. Como caso especial de un objeto no visual, los objetos remotos pueden referenciar cualquier función no visual de PowerBuilder.

Aplicación del servidor PowerBuilder.

Aquí la aplicación contiene objetos no visuales para usuario (Non-Visual User Objects, NVUO) listos para ser invocados remotamente por una o más aplicaciones cliente. La aplicación del servidor PowerBuilder contiene un objeto PowerScript que llama a un TRANSPORTE que recibe las solicitudes de los clientes para sus objetos remotos.

Aplicación del cliente PowerBuilder.

Una aplicación del cliente invoca a los servicios y objetos remotos, contiene un objeto PowerScript que llama a una CONEXIÓN para invocar a la aplicación del servidor.

Objetos para Transporte

Un objeto del lado del servidor recibe solicitudes de clientes, crea un objeto de transporte que se prepara para ejecutar las peticiones de los clientes incluyendo los atributos necesarios para la comunicación.

Objetos para comunicación

Un objeto del lado del cliente manda solicitudes, crea un objeto de conexión al servidor incluyendo los atributos necesarios para la comunicación.

Objetos Proxy

Los Objetos Proxy son utilizados para guardar el nombre de un alias remoto para el objeto y generar un código de objetos Proxy interno para habilitar el objeto del usuario a ser invocado. El código del objeto Proxy compilado es almacenado en el cliente.

V.3.2 Herramientas Back end

Son las encargadas de administrar y mantener la seguridad de la base de datos y deben estar totalmente abiertas a todos los usuarios conectados a la base de datos (conectividad).

ORACLE

Dentro de las herramientas back end, Oracle es una de las importantes en cuanto a confiabilidad, escalabilidad, conectividad y flexibilidad en sistemas abiertos. Con Oracle es posible compartir datos residentes en diferentes servidores de manera transparente. Además de ser un potente DBMS tiene herramientas para Diseño y Desarrollo de aplicaciones Cliente/Servidor, estos son: Designer/2000 para modelado y reingeniería de procesos; el Developers/2000 para construir de manera rápida y fácil las aplicaciones Cliente/Servidor. Oracle tiene una gran variedad de servicios como DBMS, entre algunos de ellos están los siguientes:

Control de concurrencia

Con Oracle se elimina la congestión en la red utilizando niveles de bloqueo de datos a nivel de registro para datos e incluso a nivel índice. Soporta OLTP y brinda a los usuarios un acceso constante a los datos más actualizados.

Operación continua (en línea)

Puede realizar operaciones como el respaldo, recuperación y administración de bases de datos en línea, de esta manera se puede aislar a las aplicaciones de las fallas de hardware.

Integridad referencial

Aplica las normas ANSI/ISO de integridad referencial de cada una de las entidades sin necesidad de escribir código, los administradores de la base de

datos sólo definen las reglas de integridad al crear o alterar las tablas y Oracle se encarga de hacerlas cumplir automáticamente.

Procedimientos almacenados PL/SQL y Triggers

La confiabilidad de las aplicaciones con Oracle como DBMS se puede lograr mediante el uso de procedimientos almacenados PL/SQL y triggers.

Los procedimientos almacenados contienen código PL/SQL combinado con SQL y están almacenados de manera compilada en la base de datos Oracle y pueden ser invocados desde los clientes remotos o se activan al realizar un movimiento en los registros de las tablas.

Los triggers se pueden utilizar para mantener datos derivados, duplicaciones síncronas de tablas y se pueden ejecutar antes o después de la instrucción de activación y/o cada vez por registro o instrucción SQL.

Seguridad

La seguridad en Oracle se maneja en forma de funciones (roles, grupos de privilegios controlados por un administrador en particular) donde los privilegios de los objetos de la base de datos se agrupan y asignan a usuarios individuales o grupos de usuario, reduciendo la carga y costo del manejo de la seguridad.

Integración de tecnologías

Por medio de una base de datos distribuida y gateways transparentes proporciona una integración transparente de sistemas nuevos y de los ya existentes.

Base de datos y consultas distribuidas

Con Oracle las bases de datos distribuidas físicamente pueden ser vistas por los programadores y usuarios finales como una sola base de datos lógica, puede acceder y actualizar desde cualquier punto de la red distribuida.

Una simple instrucción SQL puede consultar información de múltiples bases de datos y ejecutar uniones de tablas complejas de manera transparente.

Transacciones distribuidas

Utiliza el protocolo two-phase-commit para asegurar la integridad de las transacciones en los ambientes distribuidos interactuando con otras bases de datos, sistemas de archivo (inclusive con sistemas antiguos que no soportan el protocolo), sistemas no-Oracle, monitores de procesamiento de transacciones que cumplan con la norma XA para el two-phase-commit.

Duplicación de tablas

Oracle utiliza los Snapshot para la actualización de copias de tablas de solo lectura en base a la tabla maestra a diferentes intervalos de tiempo definidos por el usuario: por hora, día, semanas o meses.

Integración de sistemas no-Oracle

Oracle tiene una arquitectura abierta que permite integrarlo con otros sistemas diferentes, herramientas, aplicaciones, productos de software de otros proveedores cumpliendo con los estándares de la industria.

SQLNET

SQLNET es el software de Oracle para conectividad y acceso a datos remotos utilizado entre las aplicaciones de usuarios y las bases de datos Oracle Cliente/Servidor o múltiples bases de datos. Con él las aplicaciones y bases de datos se pueden distribuir físicamente en diferentes máquinas y continuar comunicándose como si estuvieran en la misma localidad física. También los usuarios pueden acceder cualquier servidor sin importar el protocolo de comunicaciones (acceso a datos de redes heterogéneas).

SYBASE

Sybase tiene una amplia experiencia en el ambiente de computación distribuida Cliente/Servidor. Tiene una serie de productos de conectividad, entre ellos están el Cliente abierto y Servidor abierto ya sea juntos o separados ayudan a los desarrolladores crear aplicaciones para ambientes de computación distribuida heterogéneas.

Cliente abierto permite diseñar aplicaciones robustas y eficientes que solicitan servicios e información de aplicaciones de Servidor abierto, Servidor SQL, de Replicación, Navegación y el Servidor OmniSQL. Servidor abierto es un conjunto de herramientas que ayuda a los desarrolladores integrar cualquier fuente de datos o servicios en línea. Con el uso de RPC Sybase puede trabajar directamente de servidor a servidor o de cliente a servidor. Cliente abierto simplifica el desarrollo y distribución de las aplicaciones, soportando las mismas interfaces de programación en una amplia variedad de plataformas hardware.

Contribuye a la integración de herramientas y aplicaciones para computadoras de escritorio con datos almacenados en servidores distribuidos, con esto los usuarios pueden abrir múltiples conexiones de redes heterogéneas de manera simultánea. Está disponible en más de 35 plataformas y soporta los estándares ANSI/ISO, XA para el two-phase-commit y el ODBC. Los formatos en que los usuarios requieren de múltiples protocolos de transporte son transparentes, soportando los protocolos TCP/IP, DECnet, Apple Talk, Conductos nominados, IPX/SPX y el modelo OSI.

Servidor abierto permite acceder a los recursos de información sobre empresas de cobertura amplia, dando a los programadores la capacidad de integrar servicios y alimentar datos en tiempo real e integración de bases de datos en ambientes Cliente/Servidor, además de proporcionar una API Cliente/Servidor unificada para todos los datos y servicios. Está disponible en más de 28 plataformas y permite integrar cualquier fuente de información o servicios de cómputo en línea. Con el Servidor abierto los datos y aplicaciones No-Sybase pueden manejarse en una infraestructura común Sybase.

Arquitectura Cliente/Servidor empresarial de Sybase

System 10 de Sybase es una serie de productos que dan soporte a la arquitectura Cliente/Servidor empresarial de Sybase y proporcionan los siguientes beneficios:

Tecnología probada

Sybase fué el pionero en introducir el primer RDBMS Cliente/Servidor diseñado para trabajar en línea sobre aplicaciones de misión crítica desde

1987, ahora SQL Server de Sybase es utilizado para el diseño de aplicaciones Cliente/Servidor de misión crítica de cobertura amplia.

Capacidad Altamente Escalable

Con el Servidor de Navegación se da soporte a bases de datos extremadamente grandes (desde terabytes hasta petabytes) con miles de usuarios ejecutando miles de transacciones por segundo.

Interoperabilidad

OmniSQL Gateway es un producto que proporciona interoperabilidad a través de una amplia variedad de RDBMS, datos, incluyendo Sybase, Oracle, DB2. Utilizan los procedimientos almacenados en Sybase para acceder a los datos No-Sybase.

Confiabilidad

Sybase proporciona un Servidor de Replicación para tener una alta disponibilidad de datos y transacciones en ambientes distribuidos.

Control

La familia de productos de administración de Sybase ofrecen confiabilidad y control como si fuera una computadora mainframe pero en un ambiente distribuido.

VI.- ESTRATEGIAS CLIENTE/SERVIDOR DE ALGUNOS FABRICANTES

En este capitulo se describen las estrategias proporcionadas por IBM, Digital Equipment Corporation, Hewlett Packard y Tandem para manejo de sus propios sistemas, pero además ofrecen productos o servicios que permiten la conectividad e interoperabilidad entre sistemas de otros fabricantes en una implementación Cliente/Servidor, estos pueden ser desde plataformas para cliente o servidor, bases de datos, middleware, administradores y monitores de transacciones, servicios de red, entre otras.

CLIENTE / SERVIDOR EN IBM

IBM siempre se ha caracterizado por la producción de computadoras centrales o mainframes fomentando su Arquitectura de Sistemas en Red SNA, ahora también proporciona soluciones que involucran al modelo Cliente/Servidor donde existe interacción entre aplicaciones y software diferentes que trabajan en conjunto. Como originalmente SNA se utilizó en los mainframes su introducción en el modelo Cliente/Servidor no era adecuada por lo que IBM ha desarrollado estrategias para procesamiento distribuido con interoperabilidad de aplicaciones y soporte a varios proveedores.

Arquitectura de Bases de Datos Relacionales Distribuidas DRDA

Está diseñado para acceder sistemas IBM mediante el sql y principalmente a sistemas grandes que trabajan como Back end a los que pueden acceder los clientes mediante servidores mas pequeños basados en grupos de trabajo.

Conexión de Red BluePrint

Es la entrada de IBM a los sistemas Cliente/Servidor abiertos de múltiples vendedores para dar soporte a protocolos TCP/IP, el modelo OSI y algunas interfaces de aplicaciones como DCE de OSF. Su finalidad principal es adoptar aplicaciones de múltiples fabricantes que no utilizan los protocolos de comunicación de IBM. Entre los servicios que proporciona están las Llamadas a Procedimientos Remotos y una Interfaz de Cola de Mensajes MQI que supervisa el paso de mensajes entre aplicaciones que se ejecutan en diferentes plataformas. Tanto administradores como desarrolladores se benefician al seleccionar el hardware y software sin limitarse a ningún protocolo ni interfaz.

Sistemas Operativos

IBM es una de las compañías que tiene una gran variedad de plataformas de varios tamaños además de sus sistemas operativos incluyendo los de red, para servidor y sistemas de escritorio, estos son: Memoria Virtual VM, Memoria Virtual Múltiple MVS, OS/400, LANServer, Workplace OS, AIX que es la versión de Unix para los sistemas 370/390, RS/6000 y PS/2 además del OS/2.

Sistema de Control de Información al Cliente CICS

El Servidor de Aplicaciones CICS es una familia de productos de procesamiento de transacciones moderno que permite explotar las aplicaciones y datos de muchas plataformas con hardware y software diferente. Las aplicaciones Cliente/Servidor con CICS ofrecen: **Portabilidad** a través de

plataformas IBM (AIX, MVS, OS/2, OS/400 y VSE) y de diferentes vendedores (HP, Digital, Sun y Windows NT). **Interoperabilidad** a través de LAN's y WAN's manteniendo la integridad y dependencia. **Escalabilidad** de 10 hasta 1000 usuarios y **Manejabilidad** desde un simple punto de control hasta configuraciones complejas. Además CICS incluye un monitor y administrador de procesamiento de transacciones que esta disponible para trabajar en entornos distribuidos.

Plataformas RS/6000

IBM tiene una gran variedad de sistemas RISC (RS/6000) que se caracterizan por tener una velocidad de respuesta muy rápida y pueden ser estaciones de trabajo o servidores, contando con uno o varios procesadores que van desde los 60 a 166 MHz, ancho de bus de memoria desde 64 a 256 Kb, de 16 Mb a 2Gb de memoria. Es abierto debido a la versión del sistema operativo Unix proporcionada por IBM Advanced Interactive Executive AIX. Este tipo de computadoras trabaja con la filosofía de que el 20 % de las instrucciones de la computadora realizan el 80 % de los procesos, por eso es que se denomina Computación con un Conjunto Reducido de Instrucciones RISC.

Serie AS/400

Es una estrategia para bases de datos que permite a múltiples usuarios trabajar independientemente y al mismo tiempo integrar información de manera sencilla, proporciona un ambiente flexible para incrementar su productividad. La familia de AS/400 tiene un amplio rango de modelos que ofrecen conectividad. Se ha optimizado para soportar las aplicaciones Cliente/Servidor ya que ahora interactúa con TCP/IP y ODBC. Con el

procesador PowerPC AS de 64 bits y su sistema operativo OS/400 las aplicaciones en versiones anteriores son totalmente portables.

DB/2

Es una familia de productos de bases de datos relacionales que ofrece una administración abierta de base de datos para el soporte de decisiones, procesamiento de transacciones y aplicaciones de negocios en línea. Esta familia de productos esta disponible para sistemas AS/400, RS/6000, mainframes IBM, además de plataformas HP y Sun. Es totalmente escalable, es decir que puede crecer desde un ambiente pequeño monousuario hasta un ambiente donde las aplicaciones de base de datos se pueden compartir a cientos o miles de usuarios. Esta disponible para todas las plataformas IBM.

Information Warehouse IW

Es una estrategia que IBM presenta para utilizar sus mainframes como grandes almacenadores de datos y como servidores robustos para el modelo Cliente/Servidor de vendedores múltiples.

CLIENTE / SERVIDOR EN DIGITAL

Digital tiene una amplia experiencia en el manejo de redes LAN, WAN y de diferentes vendedores, también en el manejo de varios sistemas operativos incluyendo los de Microsoft, Novell, Banyan y Apple. Ha venido trabajando con bases de datos como Sybase, Oracle, INGRES, Informix y RdB. Las soluciones que proporciona se construyen bajo interfaces de usuario incluyendo MS Windows x y Motif. En lo que se refiere a Middleware puede

trabajar con DCE, CORBA y OLE, además de soportar los estándares de la industria en sistemas operativos incluyendo Unix, OSF/1, Open VMS y POSIX. Digital propone una serie de implementaciones Cliente/Servidor abierto, entre ellas: Administración, Integración de redes, bases de datos y una estructura orientada a objetos.

Integración de datos

- **Integrador DB.** Propone una completa integración, acceso a datos heterogéneos y distribución de datos en toda la empresa para que los usuarios que trabajan con sistemas de escritorio puedan accederlos. Ayuda a realizar consultas de respuesta rápida para reducir el tráfico en la red.
- **ACCESSWORK.** Con el Integrador DB ofrece conectividad para bases de datos como Rdb, Oracle, Sybase, DB2 además del uso de ODBC.
- **DBA Workcenter.** Administra múltiples bases de datos, incluye una GUI para monitorear el rendimiento de la base de datos.
- **Ruteador de Transacciones Confiable (RTR, Reliable Transaction Router).** Protege el acceso a los datos en la red, proporcionan un sistema tolerante a fallas. Su modelo de software escalable soporta un crecimiento flexible de los sistemas en respuesta a los cambios de requerimientos de la empresa.

Integración de red (Cualquier Cliente con cualquier Servidor)

- **PATHWORKS.** Es un conjunto de programas que integra diferentes plataformas con LAN's de PC a través de la red de la empresa reduciendo significativamente los costos de integración de sistemas.

- **Ambiente de Computación Distribuida (DCE, Distributed Computing Environment)**. Proporciona una serie de servicios de software basados en estándares para crear y desarrollar aplicaciones distribuidas con acceso a una amplia variedad de plataformas.
- **WaveLAN, RemoteConnect Solution y RoamAbout**. Son familias de productos para cómputo móvil que proporcionan servicios completos de red para grupos de trabajo y usuarios remotos con equipos inalámbricos.

Estructura orientada a objetos

- **ObjectBroker**. Permite la interoperabilidad transparente entre las aplicaciones ya existentes y las nuevas en todos los clientes comunes y servidores populares.
- **Modelo de Objetos Comunes (COM, Common Object Model)**. Es una arquitectura abierta para la computación con objetos distribuidos trabajando con ObjectBroker y Objetos OLE desarrollada por Digital y Microsoft.
- **Integración de Aplicaciones Cliente/Servidor FBE**. El FBE (Framework Based Environment) se utiliza para la integración de aplicaciones Cliente/Servidor, dando a las compañías la flexibilidad de seleccionar las aplicaciones comerciales que ellos deseen incluyendo sistemas operativos y el hardware. Utiliza los estándares IDL CORBA.

Administración

- **POLYCENTER**. Es un administrador para DEC OSF/1 en NetView que combina las ventajas del TCP/IP y del sistema operativo Unix para

administrar la red de la empresa desde una estación de trabajo. Ofrece un amplio rango para administración de aplicaciones y dispositivos heterogéneos utilizando el SNMP (Simple Network Management Protocol).

- **Software Connection.** Esta compuesto de una serie de productos y servicios que ayudan a la administración. Se encarga de la evaluación, manejo de licencias y soporte al usuario.

CLIENTE / SERVIDOR EN TANDEM

Tandem se caracteriza por sus sistemas tolerantes a fallas en los que se manejan varios procesadores para incrementar el rendimiento, seguridad y distribución de procesos, además está trabajando para brindar interoperabilidad, portabilidad, manejabilidad y escalabilidad en todos sus productos o servicios.

Sistemas operativos

Todos los productos Tandem están basados en dos sistemas operativos:

- **NonStop Kernel.** Es un sistema operativo multiprocesador orientado a red diseñado para dar una alta disponibilidad de información y aplicaciones. Soporta múltiples interfaces abiertas y permite ejecutar varias aplicaciones en un ambiente en paralelo.
- **NonStop-UX.** Es una implementación de Tandem sobre el sistema operativo Unix SVR4.

Tandem ofrece herramientas para desarrollo y administración bajo ambiente Cliente/Servidor incluyendo lenguajes 3GL, C, C++ y 4GL (PowerBuilder, Visual Basic, Gupta SQLWindows) y herramientas CASE. Las Soluciones escalables y modulares de Tandem pueden crecer en paralelo junto con las necesidades Cliente/Servidor sin modificar el software de aplicación.

Servidores NonStop Himalaya

Los Servidores NonStop Himalaya son escalables y cubren las demandas de la aplicación permitiendo a los clientes expandir a varios procesadores para soportar grandes bases de datos. Para estos servidores se incluye los modelos K-100 para sistemas de escritorio, el servidor paralelo escalable para cubrir los requerimientos de procesamiento de grandes bases de datos K-1000 y el K-10000 es un servidor masivamente paralelo con procesador RISC de 4,400 MIPS para altos niveles de rendimiento y escalabilidad.

Bases de datos

Utiliza una base de datos relacional que da el rendimiento requerido para aplicaciones de procesamiento de transacciones de gran volumen, este es el NonStop SQL/MP que permite realizar consultas de datos en paralelo.

Herramientas para procesamiento de aplicaciones y acceso a datos

Para soporte al procesamiento de transacciones Cliente/Servidor ofrece:

- Llamadas a Servidores Remotos (RSC, Remote Server Call) para dar conectividad entre clientes DOS, Windows, OS/2, Unix y Macintosh con servidores NonStop, además de soportar grandes volúmenes de OLTP.

- **EL POET (Pathway Open Environment Toolkit)** facilita la construcción de puentes entre clientes Windows y un servidor de transacciones NonStop apoyándose en llamadas a Librerías de Enlace Dinámico (DLL) para integrar servicios de transacciones corriendo en servidores NonStop y los protocolos de mensajes DDE en Windows para permitir el transporte de datos entre clientes Windows y servidores NonStop.

Tandem proporciona tres interfaces para facilitar el acceso a la información en un ambiente Cliente/Servidor (middleware), estas son:

- **EL SQL Server Gateway.** para que las aplicaciones de los clientes de servidores Microsoft/Sybase accesen transparentemente a la base de datos NonStop SQL/MP, soporta clientes DOS, Windows y Unix y múltiples plataformas de base de datos incluyendo VAX de DEC, Sparc de SUN, DB2 de IBM, Oracle, Ingres y Sybase.
- **Conectividad Abierta en Bases de Datos (ODBC)** para que los clientes Tandem tengan acceso abierto a servidores NonStop y desarrollen aplicaciones Cliente/Servidor utilizando herramientas avanzadas y estándares abiertos. Con el ODBC Tandem puede invocar procedimientos almacenados.

CLIENTE /SERVIDOR EN HEWLETT PACKARD

Hewlett Packard se ha caracterizado por proporcionar soluciones Unix para sistemas abiertos implementando el modelo Cliente/Servidor sobre todo con servidores Unix basados en sistemas RISC (HP-UX). HP tiene una

estrategia Cliente/Servidor que consta de varios productos y servicios que a continuación se describen.

Plataformas HP 9000 para Servidor

Soportan configuraciones Cliente/Servidor, son escalables ya que se pueden ajustar a las necesidades crecientes de los negocios. Son totalmente abiertas porque se ajustan a la industria de los estándares para garantizar seguridad a los usuarios en plataformas para un ambiente Cliente/Servidor.

Plataformas para Cliente

Tiene un amplio rango de sistemas de escritorio incluyendo PC, estaciones de trabajo, entre ellas DOS, Windows, OS/2, Unix y Macintosh. Para estos ambientes operativos de cliente HP soporta conexiones a todos los servidores y sistemas operativos de red.

Sistema Operativo HP-UX

Es la versión del sistema operativo Unix proporcionado por HP, ofrece flexibilidad, escalabilidad e integración del sistema operativo líder en sistemas abiertos. También da un control, seguridad e integridad de datos para aplicaciones críticas, además de la conectividad, interoperabilidad y capacidad de crecimiento basado en usuarios de PC's.

Soporte a bases de datos y otras herramientas

Los equipos HP soportan el software de diferentes proveedores, bases de datos, herramientas CASE y desarrolladores de aplicaciones, proveedores de bases de datos relacionales, entre ellas: Oracle, Sybase e Informix.

Middleware para procesamiento distribuido

Para desarrollar y ampliar las aplicaciones Cliente/Servidor, HP proporciona una infraestructura basada en estándares conocida como Servicios de Computación Distribuida para incrementar la interoperabilidad de las aplicaciones y proporciona seguridad en ambientes heterogéneos distribuidos. DCE/9000, es una mejora al DCE de la Fundación de Software Abierto OSF, incluye llamadas a procedimientos remotos, conversión de datos automática, un sistema de archivos distribuidos, servicios de directorios local y global, seguridad en transmisión de datos, soporta la ejecución de tareas en paralelo y una colección de herramientas gráficas para facilitar la administración. Combinando estos servicios se desarrolla y maneja el procesamiento de transacciones distribuidas, objetos y aplicaciones Cliente/Servidor.

HP soporta un amplio rango de productos middleware para OLTP distribuidas para reducir la carga de trabajo y el tiempo para realizarlo en ambientes distribuidos. Estas son Encina/9000, CICS para HP9000, Tuxedo para HP9000, TpEnd de NCR y UniKix de UniKix technology.

Tecnologías para red

Para facilitar la comunicación entre clientes y servidores HP soporta todas las tecnologías de red basadas en estándares. Los productos de red HP proporcionan conectividad y servicios de red a otros sistemas desde el rango de los mainframes hasta clientes de escritorio. HP soporta los protocolos TCP/IP, SNA, IPX y sobre todo el modelo OSI, en una variedad de enlaces tales como Ethernet, Token Ring, X.25, FDDI, ISDN y SDLC.

VII.- EJEMPLO DE UNA APLICACIÓN CLIENTE / SERVIDOR

El ejemplo que se describirá es una propuesta de desarrollo en conjunto en el Departamento de Soporte Informático de la Coordinación Informática de SIPE (Subdirección de Ingeniería de Proyectos de Explotación) del Instituto Mexicano del Petróleo para la Gerencia de Programación y Evaluación de PEMEX.

SISTEMA PROBGAS: Automatización de los procedimientos de balance de gas y condensados.

OBJETIVO:

Consolidar y regular la normatividad y procedimientos para la elaboración del Balance de Gas y Condensados en Sector Operativo, Distrito, Región y SEDE.

Automatizar la Normatividad establecida.

BENEFICIOS:

Generar una fuente única de información relativa a la distribución de gas natural y sus condensados.

Generar informes operativos, administrativos y ejecutivos en forma automática y funcional, partiendo de la captura de información en los sectores operativos, integrando en Distrito y Región y consolidando a nivel Nacional en SEDE:

- Producción y Distribución de gas natural
 - Diario
 - Mensual
 - Trimestral

- Producción y Distribución de condensados
 - Diario

Mensual Trimestral

Contar con una herramienta que permita dar seguimiento a los compromisos empresariales de Pemex Exploración y Producción PEP y maximizar el valor económico del gas natural y de los condensados detectando áreas de oportunidad.

Para lograr los objetivos del sistema es necesario enviar información de campo y sector a Distrito, de Distritos a Región y de Regiones a SEDE debe ser diario para mantener actualizada la base de datos y generar los reportes de producción y distribución de gas y condensados. En la **figura 7.1** se da una breve descripción de la Organización de Pemex por Sectores Operativos, Distritos, Regiones y SEDE para explicar el flujo información que tendrá el sistema. En la **figura 7.2** se muestra como queda integrada la información a nivel Nacional.

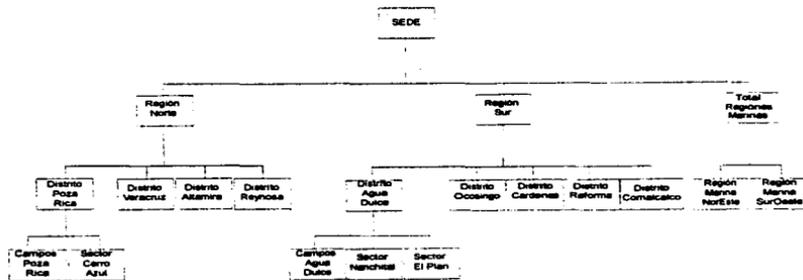
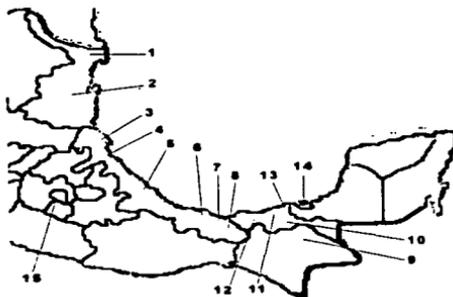


Fig. 7.1 Organización de los Distritos y Regiones de Pemex



Fig. 7.2 Integración de información



1. Reynosa
2. Altamira
3. Cerro Azul
4. Poza Rica
5. Veracruz
6. Agua Dulce
7. Nanchital
8. El Plan
9. Ocosingo
10. Cárdenas
11. Reforma
12. Comalcalco
13. Dos Bocas
14. Ciudad del Carmen
15. Ciudad de México

Fig. 7.3 Distribución geográfica de los Distritos y Regiones de Pemex

ESTADO ACTUAL DEL SISTEMA

El sistema se compone de las regiones Norte, Sur y Marina, además de SEDE (o sistema total) que es donde se integra la información procedente de cada una uno de los módulos.

En la Región Norte se encuentra instalado un prototipo del sistema con un modelo standalone (PC independiente) bajo ambiente Windows en Access 2.0. Como se debe integrar la información de producción y distribución de gas y condensados es necesario el envío diario de datos por lo que bajo en este modelo solo se puede realizar generando un archivo con los datos y enviarlo por ftp para su integración, esto se vuelve tedioso y difícil de administrar porque se trabaja con datos operativos y al cierre del mes o antes se realizan ajustes a los datos para que sean facturables y es necesario reenviar la información para integración.

Estos problemas se resuelven utilizando un ambiente Cliente/Servidor en donde se interconectarán todos los Sectores, Campos, Distritos, Regiones y Sede, utilizando SQL Windows para los Clientes y Oracle RDBMS como Servidor y el SQL Net para conectividad entre el front end y el back end. Esta configuración y la distribución de funciones para los Clientes y Servidores se muestran en la **figura 7.4**.

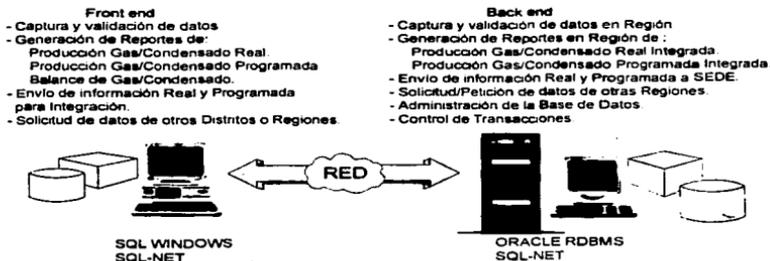


Fig. 7.4 Distribución de funciones Cliente/Servidor

MÓDULO DE LA REGIÓN NORTE

En éste Módulo se tiene el Sector Cerro Azul y Campos Poza Rica que pertenecen al Distrito Poza Rica. Cerro Azul tiene su propia base de datos y su parte de la aplicación correspondiente y envía su información al Distrito Poza Rica. Mientras que Campos Poza Rica sólo tiene su parte de la aplicación y sus datos se almacenarán de manera remota en la base de datos de Distrito Poza Rica y al mismo tiempo integrará la información junto con la de Cerro Azul y la enviará a la Región Norte (que tiene su sede en Poza Rica y en el mismo edificio). Veracruz, Altamira y Reynosa envían su información para integración a la Región Norte, figura 7.5.



Fig. 7.5 Diagrama Cliente/Servidor de la Región Norte

MÓDULO DE LA REGIÓN SUR

De manera análoga que en el Distrito Poza Rica, el Distrito Agua Dulce Integrará la información correspondiente al El Plan, Nanchital y Campos Agua Dulce para enviarlos a la Región Sur. El Distrito Ocosingo tiene su propia base de datos e integra en Región Sur. Los Distritos Cárdenas, Reforma y Comalcalco son interdependientes por lo que sus datos se almacenarán de manera remota en la Región Sur ya que estos también se utilizarán en las Regiones Marinas, **figura 7.6**. La Región Sur enviará la información integrada para Sede.



Fig. 7.6 Diagrama Cliente/Servidor de la Región Sur

MÓDULO DE REGIONES MARINAS

El flujo de información entre las Regiones Marinas esta ligado con información proveniente de la región Sur y es bastante dinámico por lo que se tendrá una base de datos centralizada en Total Regiones Marinas (TRM) ubicada en RM NorEste (Ciudad del Carmen). El acceso a la información por cada uno de las RM NorEste y SurOeste (Dos Bocas) será de manera remota y solo tendrán los programas de la aplicación correspondientes a cada una de ellas. TRM Es quien se encargará de enviar la información de RM NorEste y RM SurOeste por separado para su integración en Sede, **figura 7.7**.



Fig. 7.7 Diagrama Cliente/Servidor de las Regiones Marinas

INTEGRACIÓN DEL SISTEMA TOTAL

Una vez que las regiones Norte, Sur y Total Regiones Marinas envíen su información a Sede ya se tendrá el nivel más alto del sistema que es la integración de la producción y distribución de gas natural y condensados de todo el país, generando reportes de producción real y programada total o por Regiones, figura 7.8.

Los reportes de balance de gas y condensados se generarán en sus niveles respectivos: Sectores, Campos y Distritos.

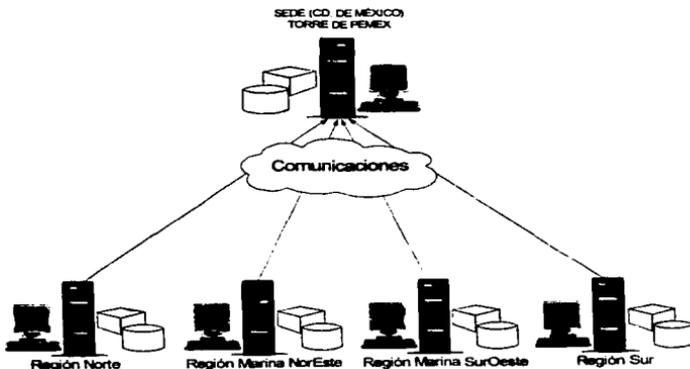


Fig. 7.8 Diagrama Cliente/Servidor para integración del Sistema Total

CONCLUSIONES

Cliente/Servidor es la integración eficiente de las nuevas tecnologías en computación tanto de software, hardware y comunicaciones. Esto incluye Bases de Datos, Sistemas Operativos, Herramientas para diseño y desarrollo de aplicaciones y productos de conectividad.

Cliente/Servidor es un tema bastante mencionado por casi todos los proveedores de tecnologías de computación, quien no lo mencione en los productos o servicios que ellos mismos proporcionan prácticamente quedan en desventaja en relación con quienes si lo hacen.

Existen varias alternativas del modelo Cliente/Servidor como son: funciones distribuidas, acceso a datos remotos, presentación remota, distribución de datos, monitores de procesamiento de transacciones, por lo que su interacción en el ámbito de las redes de computadoras es bastante amplio.

El hecho de que se tenga otra alternativa de procesamiento de información como lo es el proceso distribuido y su modelo Cliente/Servidor, no significa que las aplicaciones con grandes computadoras centrales tengan que ser desplazadas, aún existen aplicaciones que requieren de procesamiento centralizado, por tal motivo se tiene que realizar una evaluación cuidadosa al tratar de diseñar o migrar a un modelo Cliente/Servidor.

La infraestructura de comunicaciones es quizá el aspecto más crítico para que Cliente/Servidor pueda tener éxito, debido a que la transparencia en el

proceso y acceso a datos, reducción del tráfico en la red y velocidad de proceso dependen totalmente de ella y se tiene que invertir bastante en ello.

Cliente/Servidor es caro y difícil de implementar, pero el rendimiento de éste modelo y las aplicaciones inclinan la balanza para que las empresas adopten este modelo de procesamiento de información y los beneficios que proporciona.

Aunque hoy en día existen herramientas y controladores que permiten la interconexión entre sistemas heterogéneos, no es tan fácil implantarlos. Los proveedores tanto de software, hardware y comunicaciones mencionan que sus productos son abiertos y proporcionan transparencia en el proceso de la información, pero en realidad hay que tener mucho cuidado al instalar los controladores correspondientes tanto en el cliente como en el servidor.

Cliente/Servidor ha tenido un auge que en la famosa Internet se ha adoptado este modelo, los clientes cuentan con un navegador para realizar sus peticiones de información hacia servidores que nunca sabrán donde están ni de donde les traerá la información, es totalmente transparente.

Las nuevas versiones de herramientas sql para el desarrollo de aplicaciones de bases de datos Cliente/Servidor ya tienen incorporado módulos para migrar sus aplicaciones a páginas de Internet para que sean accesadas desde cualquier parte del mundo.

Las nuevas tendencias del modelo Cliente/Servidor son los objetos distribuidos (que es la combinación de las tecnologías orientadas a objetos con el modelo Cliente/Servidor) permitiendo que las aplicaciones se encapsulen

como objetos y se distribuyan a través de una red heterogénea donde todos los componentes pueden interoperar entre sí como si fuera una sola entidad, aunque también es bastante complicado y difícil de implantar.

El Auge de la computación móvil (redes inalámbricas) ahora permite que las aplicaciones Cliente/Servidor se puedan controlar y administrar desde computadoras portátiles o una PC conectada a un módem (por ejemplo cargar y descargar la base de datos, delegar privilegios, etc.), sin necesidad de que el responsable se encuentre físicamente en el lugar donde se localice el Back end.

BIBLIOGRAFÍA

- Digital Equipment Corporation
Guide to building Client/Server solutions
USA 1993, 507 págs.
- Digital Equipment Corporation
Integrated desktop computing
USA 1991, 453 págs.
- Sheldon, Tom, et al
Guía de interoperabilidad: Soluciones para la interconectividad en red
Osborne McGraw Hill (LAN TIMES)
México 1995, págs.
- Sheldon, Tom, et al
Enciclopedia de redes. NetWorking
Osborne McGraw Hill (LAN TIMES)
México 1995, págs.
- Sheldon, Tom
Novell Netware 386. Manual de referencia
Osborne McGraw Hill
México 1992, 743 págs.
- Stevens, Guengerling
Downsizing information systems
Editor-Series
USA 1992, 429 págs.
- Day, Michael
Downsizing to Netware
Ed. New Riders Publishing
USA 1992, 710 págs.

- H. M. Deitel
Sistemas operativos
Addison-Wesley Iberoamericana
USA 1993, 938 págs.
- Martin J.-Odell J.
Análisis y diseño orientado a objetos
Ed. Prentice Hall H.
Edo. México 1994, págs.
- Kemper, Moerkotte
Object-Oriented data base management
Ed. Prentice Hall
New Jersey 1994, 687 págs.
- Gupta
SQLNetwork: Guide to Gupta Connectivity
USA 1994, 151 págs.
- Uniface
UNIFACE SIX: Environment specific guide
Compuware Corporation 1995, 254 págs.
- Shwartz, Mischa
Redes de telecomunicaciones
Addison-Wesley Iberoamericana
USA 1993, 787 págs.

OTRAS PUBLICACIONES

- Sun Microsystems Computer Corporation
Managing a dsitributed environment: alternatives for data distribution
USA 1994, 29 págs.
- Sun Microsystems Computer Corporation
RIGHTSIZING: Re-engineering Information Systems Through Client-Server Technology
USA 1994, 24 págs.
- Texas Instruments
Information Engineering Facility For Client/Server
USA 1993
- Sybase
The road Map to Enterprise Client/Server Computing
USA 1992, 16 págs.
- Oracle
Oracle 7 Database Management System Release 7.1
USA 1995, 6 págs.

DIRECCIONES DE INTERNET

- **The Client-Server Computing Page®**
<http://www.wenet.net/~jtmalone/>
Febrero 1997
- **SYBASE Open Client and Open Server**
<http://www.sybase.com/software/conectivity.html>
1996
- **Powersoft - Technology Briefing Series**
http://www.powersoft.com/tools/pb50/dpb_wh3.html
1996
- **Open Client/Server Solutions Guide**
<http://www.dmo.com/gsy/1b1.html>
Junio 1996, Hewlett Packard
- **Managing Risk in Large-Scale Client/Server Systems**
<http://www.merc-int.com/products/whitepaper2/>
Abril 1996
- **Distributed Objects for business**
<http://www.sun.com/sunworldonline/swol-04-1996/sowl-04-ooobook.html>
Abril 1996