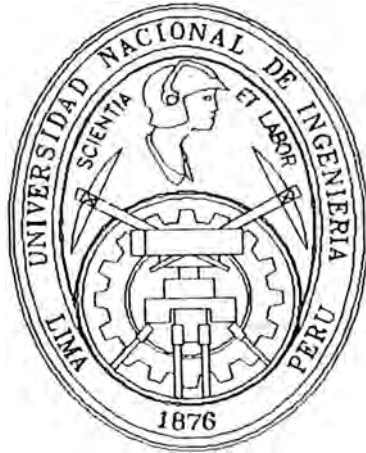


Universidad Nacional de Ingeniería
FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS



**SOLUCION CLIENTE-SERVIDOR APLICADO A UNA EMPRESA
COMERCIAL DEL RUBRO DE AUTOPARTES**

TESIS

Para Optar el Título Profesional de:

INGENIERO DE SISTEMAS

ISABEL SAAVEDRA BASALDUA

ROXANA RAMOS CARRION

**Lima - Perú
1998**

**A mi madre Rosa:
Por su invaluable apoyo y sabios consejos
que me ayudaron a lograr mi carrera.**

**A la memoria de mi padre Vicente:
Igualmente por su apoyo invaluable,
sus consejos y su ejemplo de profesionalismo.**

**A mis hermanos Graciela y Fernando:
Con su ejemplo fueron una luz que me guió
permanentemente para lograr mis
objetivos y en particular este.**

**A mi esposo Jaime y
mis hijos Jaime Fernando y Brian Angel:
Por haberles privado de mis cuidados
y dedicación muchas horas, y ser un constante
estímulo en mi superación.**

A la memoria de mi padre Erasmo Ramos:

por el infinito amor que me brindó.

Y

A mi madre Ana Carrión:

por su cariño y dedicación.

Agradecimientos

**Al Ing. Luis Bullón, por sus invaluable
consejos y por todo el apoyo brindado
para la culminación de esta tesis.**

**Al Sr. Ciriano Aguilar por la facilidad
que nos brindó para la tramitación de esta tesis.**

INDICE

Sumario		8
Introducción		9
Cap.I	Análisis de la situación actual	13
1.1	Areas de la compañía	14
1.2	Principales funciones de las áreas	14
1.3	Recursos de Hardware	15
1.4	Recursos de Software	16
1.5	Personal de Area de sistemas	17
1.6	Requerimientos de las Areas de la Compañía	17
1.7	Diagnóstico del estado actual de las necesidades de la compañía	20
Cap. II	Fundamentos Teóricos de la Arquitectura Cliente/Servidor	26
2.1	Principales componentes	26
2.2	Características de los Sistemas C/S	28
2.3	Atributos de una Aplicación C/S	30

2.4	Modelos de distribución C/S	31
2.5	Categoría de los servidores en una Arquitectura C/S	33
2.5.1	File Servers	33
2.5.2	Database Servers	33
2.5.3	Transaction Servers	34
2.5.4	Groupware Servers	36
2.5.5	Object Servers	36
2.5.6	Web Servers	38
2.6	Administración de las Bases de Datos	40
2.6.1	Arquitectura de los sistemas administradores de BD	41
2.7	SQL (Structure Query Language)	47
2.8	Principales administradores de BD	52
Cap. III	Solución Cliente/Servidor	62
3.1	Principales Objetivos Organizacionales	62
3.2	Objetivos de la Solución Planteada	65
3.3	Fases de la solución planteada	67
3.4	Características principales del Hardware	69
3.5	Características principales del administrador de BD de la solución planteada	72

3.6	Características de la Arquitectura de Comunicaciones	74
3.7	Características del Groupware Server seleccionado	76
3.8	Características principales del Sistema Operativo	78
3.9	Recursos Humanos	80
3.10	Metodología para el desarrollo de los aplicativos	81
3.11	Consideraciones administrativas del modelo C/S	84
3.12	Aspecto organizativo y consideraciones del aplicativo	86
3.13	Consideraciones básicas de seguridad del modelo	87
3.14	Herramientas de desarrollo	88
3.15	Estudio Costo/Beneficio	92
3.16	Características modulares del Nuevo Sistema de Información Integral	95
	Conclusiones	99
	Recomendaciones	102
	Bibliografía	104
	Glosario	105
	Anexos	108

SUMARIO

El objetivo de nuestra tesis se centraliza en el desarrollo de una Solución Cliente/Servidor orientada a una empresa peruana del rubro de autopartes. La solución que se plantea en esta tesis busca descentralizar la información y hacerla más asequible y manejable a todas las áreas de la empresa tal que ésta pueda alcanzar sus objetivos organizacionales.

La empresa en estudio cuenta en la actualidad con una arquitectura centralizada bajo la plataforma IBM AS/400, que como sabemos es una plataforma propietaria. La solución que planteamos en esta tesis sigue un proceso metodológico que comprende conceptos de vanguardia acerca de la tecnología Cliente/Servidor, con procesos distribuidos, bases de datos relacionales, conexiones Internet, correo electrónico y programación visual.

Descriptores Temáticos

Arquitectura Cliente/Servidor

Rubro de Autopartes

SQL Server

Stored Procedures

Triggers

INTRODUCCIÓN

A sólo un año del advenimiento del Nuevo Milenio, nos encontramos viviendo en un mundo inmerso en una economía de libre mercado y globalización. Es en este mundo cambiante donde cada país, individuo o empresa requieren de una actualización constante en lo que a Ciencia y Tecnología se refiere, porque, las ventajas competitivas no radican ya solamente en los recursos naturales que posee un país o en la mano de obra disponible o barata, en la actualidad las ventajas competitivas están dadas por el know-how, la tecnología y el manejo más eficiente de la información para la toma de decisiones.

Hoy en día todas las empresas asignan una alta prioridad al recurso INFORMACION, de tal forma que la tecnología de la información o de sistemas sea considerada una real inversión reflejándose sus beneficios en el crecimiento competitivo sobre la base de procesos modernos y efectivos, bajo un marco global de estandarización. Las empresas que puedan implementar rápidamente cambios en su línea de productos y servicios de acuerdo a los requerimientos del mercado sin tener que enfrentar modificaciones traumáticas a su infraestructura de información tendrán una potencial ventaja frente a sus competidores.

En este marco la Ingeniería de Sistemas cumple un papel primordial como soporte del negocio, buscando soluciones que puedan ajustarse a la estructura organizativa y a los procesos del mismo de una forma natural, debe proporcionar también la infraestructura necesaria y definir los estándares de tecnología informática de toda la empresa. Actualmente la solución Cliente/Servidor es una

buena alternativa que cumple con las exigencias del negocio y tiene importantes implicancias en el futuro de la empresa, aunque sus costos no son tan económicos a corto plazo, pero en cambio los beneficios serán múltiples.

Es por este motivo que el tema de Investigación de nuestra tesis plantea una solución óptima que pueda ser adecuada a la realidad de la empresa en estudio a la cual en el transcurso de la tesis denominaremos como "COMPAÑIA". La tesis propondrá una solución Cliente/Servidor tomando en consideración un estudio costo-beneficio.

En el Capítulo I describimos la situación actual de la compañía y la funcionalidad de cada una de sus áreas, así como la situación del hardware y software con que cuenta identificando sus problemas más críticos.

En el Capítulo II se describe detalladamente los principales conceptos, componentes y fundamentos de una Solución Cliente/Servidor, así como sus características y tipos.

En el Capítulo III planteamos como solución al diagnóstico, implementar una Solución Cliente/Servidor que cuente con un servidor principal al sistema AS/400, y utilizar los aplicativos que actualmente tiene. Además tendremos una réplica del modelo de datos en un servidor para la red, donde instalaremos como manejador de base de datos el SQL SERVER el cual servirá como soporte de base de datos a los aplicativos desarrollados en programas visuales que en un inicio será para todas las consultas y reportes, y en futuro para los aplicativos de un nuevo sistema integral. Además contaremos con otros 2 servidores, uno que contendrá todos los aplicativos de BACK-OFFICE y el otro contendrá un sistema de mensajería. El

modelo de procesos y datos será distribuido y la arquitectura de aplicaciones será de tres niveles o 3-tier. La plataforma operativa se desarrollará bajo Windows NT.

También describimos en este capítulo las consideraciones básicas para el planeamiento integral de nuestra solución Cliente/Servidor, tales como arquitectura, fases y tiempos de implantación, diseño de aplicaciones, plataformas operativas, administración de las Bases de Datos y el entorno de desarrollo para un nuevo sistema integral de acuerdo al análisis de requerimientos recogido de los usuarios y para hacer uso óptimo de todas las herramientas planteadas en esta solución.

CAPITULO I

ANALISIS DE LA SITUACION ACTUAL

La Compañía en estudio es una empresa peruana con más de 20 años de fundada que tiene como rubro principal la fabricación y comercialización de autopartes. En la actualidad esta compañía cuenta con dos sedes, una en Lima, donde se encuentran ubicadas las oficinas administrativas y otra en Chincha donde se encuentra la planta de fabricación.

Sus productos líderes en ventas son los muelles y bujías, siendo los muelles los productos que por su calidad, han logrado convertirse en productos de exportación con mucho éxito en USA, Canadá y algunos países del Grupo Andino.

La compañía cuenta actualmente con un staff de 200 empleados que conforman la plana administrativa. El Area de ventas constituye actualmente una de las áreas con mayor impulso y crecimiento y cuenta con un staff de 13 Vendedores y 2 supervisores de Ventas.

Los Vendedores para recoger los pedidos y realizar las cobranzas deben recorrer el país según las zonas de venta que se les haya destinado. Estas zonas de Ventas se distribuyen en: Norte, Sur, Centro, Oriente y Lima.

Organigrama General de la Compañía

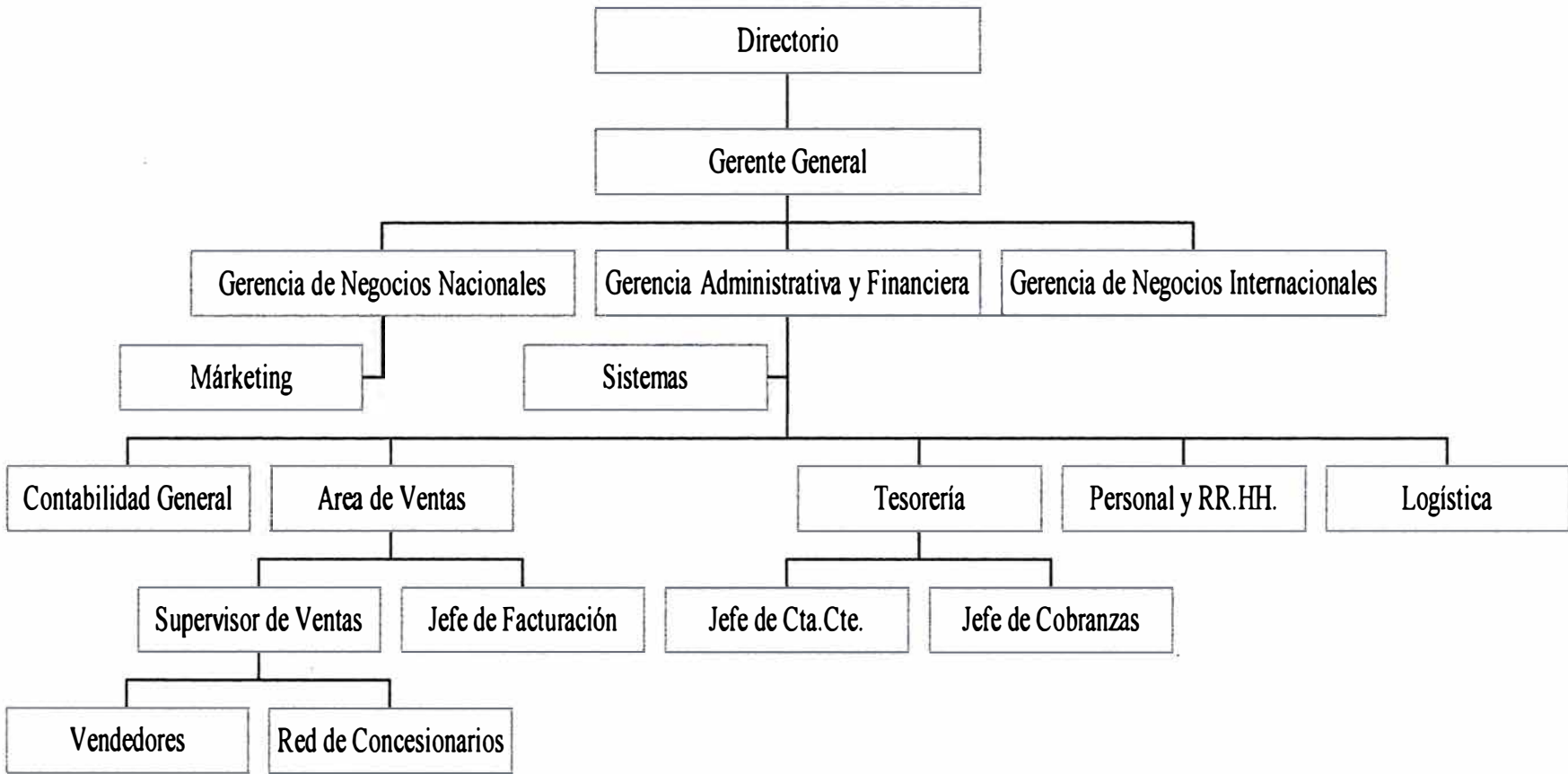


Fig. 1

La compañía cuenta en el ámbito nacional con aproximadamente 220 concesionarios los cuales gozan de ciertos beneficios y precios especiales que la compañía les brinda. Estos 220 concesionarios son los principales distribuidores mayoristas en el ámbito nacional.

Estos Concesionarios conforman lo que se denomina la "Red de Concesionarios" y son la principal fuerza de ventas de la compañía y están muy identificados con los objetivos de la misma.

1.1 Áreas de la Compañía.-

Se pueden identificar las siguientes áreas dentro de la compañía:

1. Area de Ventas
2. Area de Tesorería
3. Area de Contabilidad
4. Area de Logística
5. Area de Personal
6. Area de producción

1.2 Principales funciones de las áreas.-

1.2.1 Area de Ventas. - Funciones principales:

- Registrar pedidos de ventas provenientes de los clientes en mostrador y concesionarios.

- Registro y control de las ventas realizadas por la compañía.
- Emisión de facturas y boletas.
- Elaboración de Presupuestos ó Cotizaciones Valorizadas.
- Controlar y calcular las comisiones de ventas para los vendedores.
- Administración de los vendedores de la empresa y las ventas que estos realicen

1.2.2 Area de Tesorería.- Sus principales funciones son las siguientes:

- Control de las Cobranzas de los clientes cuyas compras sean al crédito y al contado.
- Control de los límites de crédito de los clientes
- Control de las Cobranzas por vendedor referente a los concesionarios asignados a su cartera de clientes.
- Control de movimientos con las entidades Bancarias.

1.2.3 Area de Contabilidad.- Sus principales funciones son:

- Registrar las compras y ventas.
- Cálculo de las provisiones para el pago a los proveedores
- Cálculo de Balances y Estados de Pérdidas y Ganancias.
- Cálculo y Pago de los Impuestos.
- Generar asientos de cierre y de apertura.

1.2.4 Área de Logística.- Sus principales funciones son:

- Clasificar los productos en forma flexible y codificarlos basándose en estándares internacionales.
- Control del ingreso y salida de productos mediante kárdex manuales.
- Cálculo de Stocks actualizados de los productos.
- Registrar ordenes de venta y guías de remisión.

1.2.5 Área de Personal y RR.HH.- Sus principales funciones son:

- Manejar información actualizada del personal de la empresa.
- Cálculo de las planillas y emisión de boletas de pago.
- Administrar los procesos de registro y cálculo del seguro Privado de Pensiones (AFP).
- Control de asistencia del personal

1.3 Recursos de Hardware.-

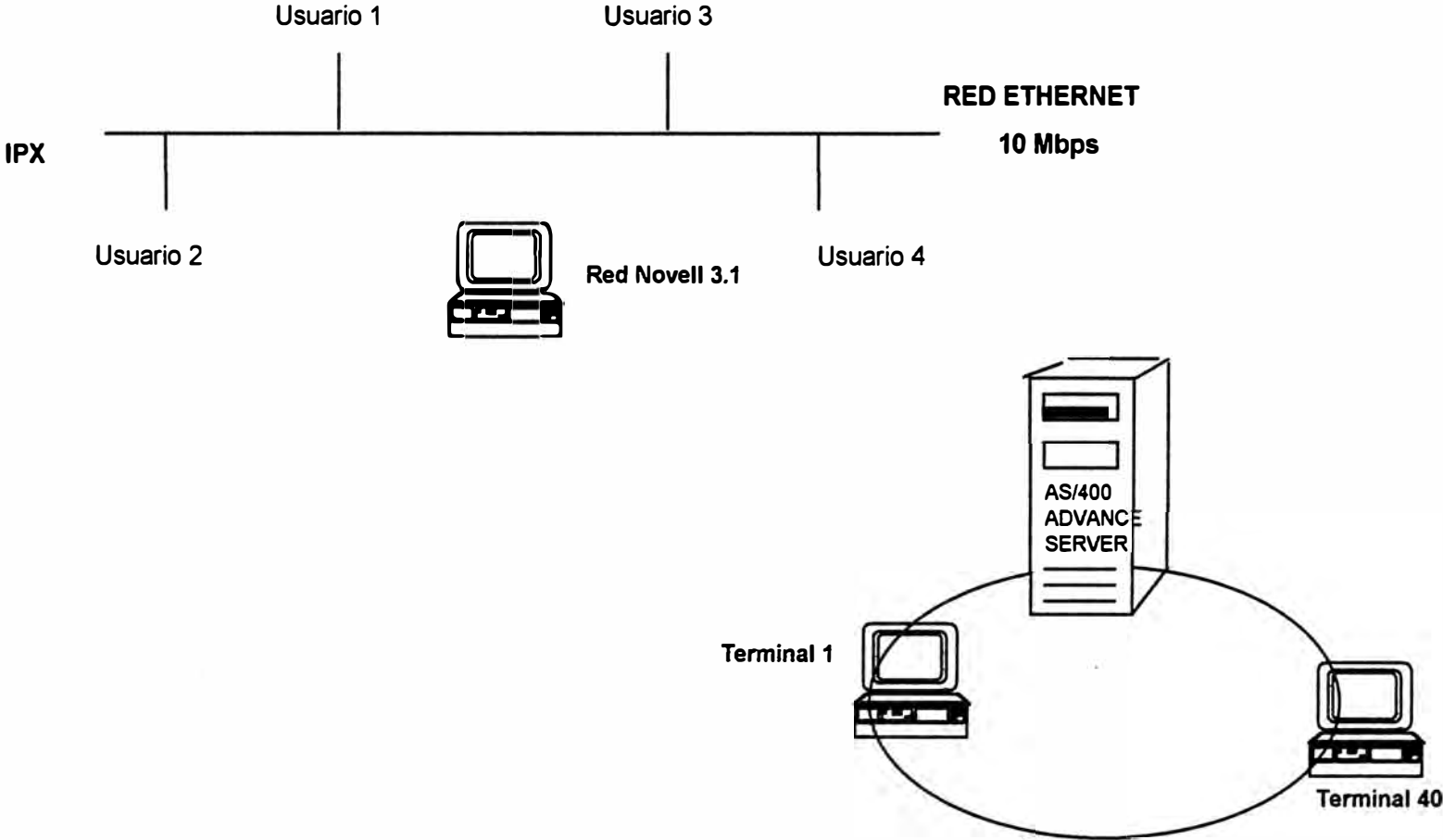
Actualmente la compañía cuenta con los siguientes recursos de Hardware:

- Un Sistema IBM AS/400 Advance Server.

Características:

- 16 GB en disco Duro
- 64 MB de Memoria RAM

Plataforma Actual de la Compañía



Este se encuentra ubicado en las oficinas de Lima.

- 40 terminales IBM
- 34 PC's
 - a) 10 con microprocesador 80486
 - b) 24 con microprocesador 80386

13 PC's tienen instalada una tarjeta de emulación.

- 1 Red Novell topología Anillo con 4 estaciones de trabajo y aplicaciones en Qpro 4.0.
- 17 impresoras de diferentes marcas.

En la figura adjunta se puede apreciar la Plataforma Actual de la Compañía, en la cual se observan el sistema IBM AS/400 dispuesto en Lima y la red Novell de 4 usuarios.

1.4 Recursos de Software.-

Los principales aplicativos de la compañía están desarrollados en lenguaje nativo RPG/400.

El sistema de Contabilidad cuenta con un Sistema Contable desarrollado en QPRO el cual le permite realizar sus informes finales.

El área de personal cuenta con un Sistema desarrollado en LOTUS para el cálculo del pago a las AFPs y el pago de impuestos.

El área de ventas específicamente para el rubro de repuestos tiene un Sistema desarrollado en FOXPRO.

En el ámbito de computadores personales se trabaja bajo el Sistema Operativo DOS 6.0.

La Gerencia utiliza para sus trabajos y presentaciones el Windows 95 así como Word, Excel y Power Point.

1.5 Personal del Area de Sistemas.-

El personal que constituye el Area de Sistemas tiene a su cargo las labores de análisis, programación y mantenimiento de los sistemas desarrollados en RPG/400 que se encuentran actualmente en producción. Este personal es el siguiente:

(1) Jefe del Centro de Cómputo

(2) Operadores del Sistema AS/400

(1) Jefe de Proyectos

(3) Analistas-Programadores

(2) Digitadores

1.6 Requerimientos de las áreas de la Compañía.-

- 1) Actualmente la compañía carece de una conexión entre su sistema AS/400 y la red Novell del Area de Contabilidad, así como otras aplicaciones que generan resultados finales de los balances, pago de impuestos y otros. Porque para obtener la información final se debe ingresar nuevamente los resultados parciales de los reportes emitidos por el Sistema AS/400 a los archivos o tablas con que trabajan los software de PC, generando esto un considerable retraso.

- 2) Las diversas áreas tienen sistemas que los ayudan en el manejo de sus funciones pero estos sistemas tienen que ser alimentados mediante diskettes, los cuales acarrearán serios problemas por el tamaño de la información, diferencia de versiones y proveedores.
- 3) El ingreso de los pedidos de venta al Sistema no modifica los stocks en forma real y no controla automáticamente la disponibilidad de crédito del cliente el cual debe pasar por diversos visados que entorpecen la venta creando un terrible cuello de botella.
- 4) Es necesario emitir cotizaciones automáticas valorizadas, las cuales deben ser realizadas por el mismo vendedor o concesionario con pantallas de ingreso amigables. Las cotizaciones deben poder emitirse desde cualquier punto del país, así como se debe poder realizar consultas de precios y stocks.
- 5) El área de sistemas cuenta actualmente con un gran número de solicitudes pendientes. Los cuales no se atienden debido a que los programadores se dedican al mantenimiento de los aplicativos existentes, descuidando el desarrollo de nuevos sistemas que los lleven a una integración del flujo de información y la automatización de la misma. Existen muchos controles y funciones manuales que muy bien pueden ser automatizadas.
- 6) Actualmente no se generan automáticamente los asientos contables, existiendo islas de sistemas porque no existen interfaces automáticas entre ellas.
- 7) El Sistema AS/400 carece de un atractivo front-end para soportar el desarrollo de una Interfaces Gráfica para el Usuario (GUI).

1.7 Diagnóstico del estado actual de las necesidades de la Compañía.-

Para lograr un diagnóstico adecuado, ha sido necesario pasar por un proceso de investigación y documentación de los procedimientos existentes de cada una de las áreas de la compañía tal como se describió líneas arriba. Analizando los requerimientos de la compañía planteamos el desarrollo de un Sistema de Información Integral con interfaces y comunicación entre los sistemas desarrollados en lenguaje RPG para AS400 y los diversos sistemas desarrollados para PCS que ayudan a obtener información final. Así como la automatización de funciones y controles manuales.

Diagrama Contextual del Sistema de Ventas

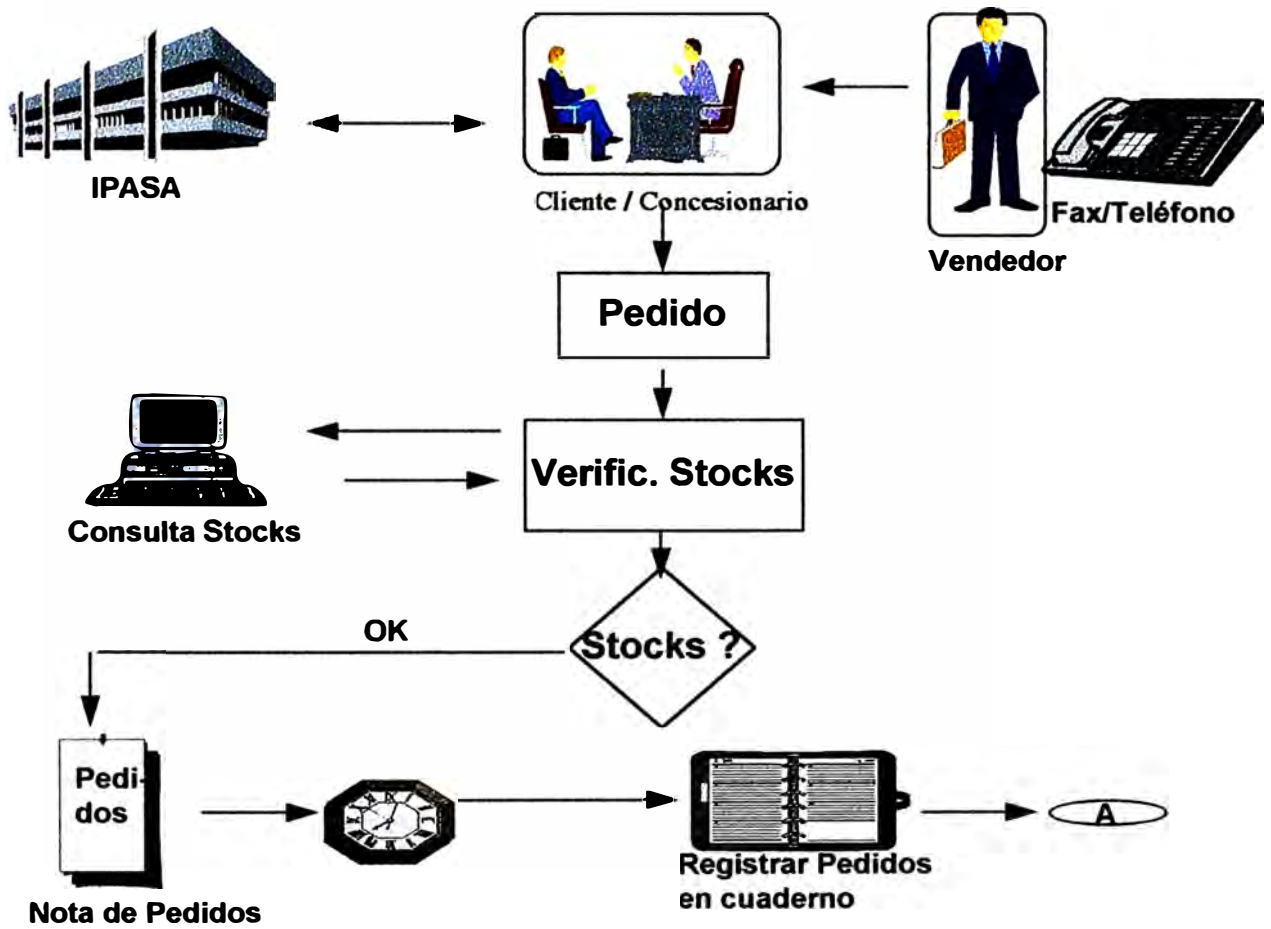


Fig.2

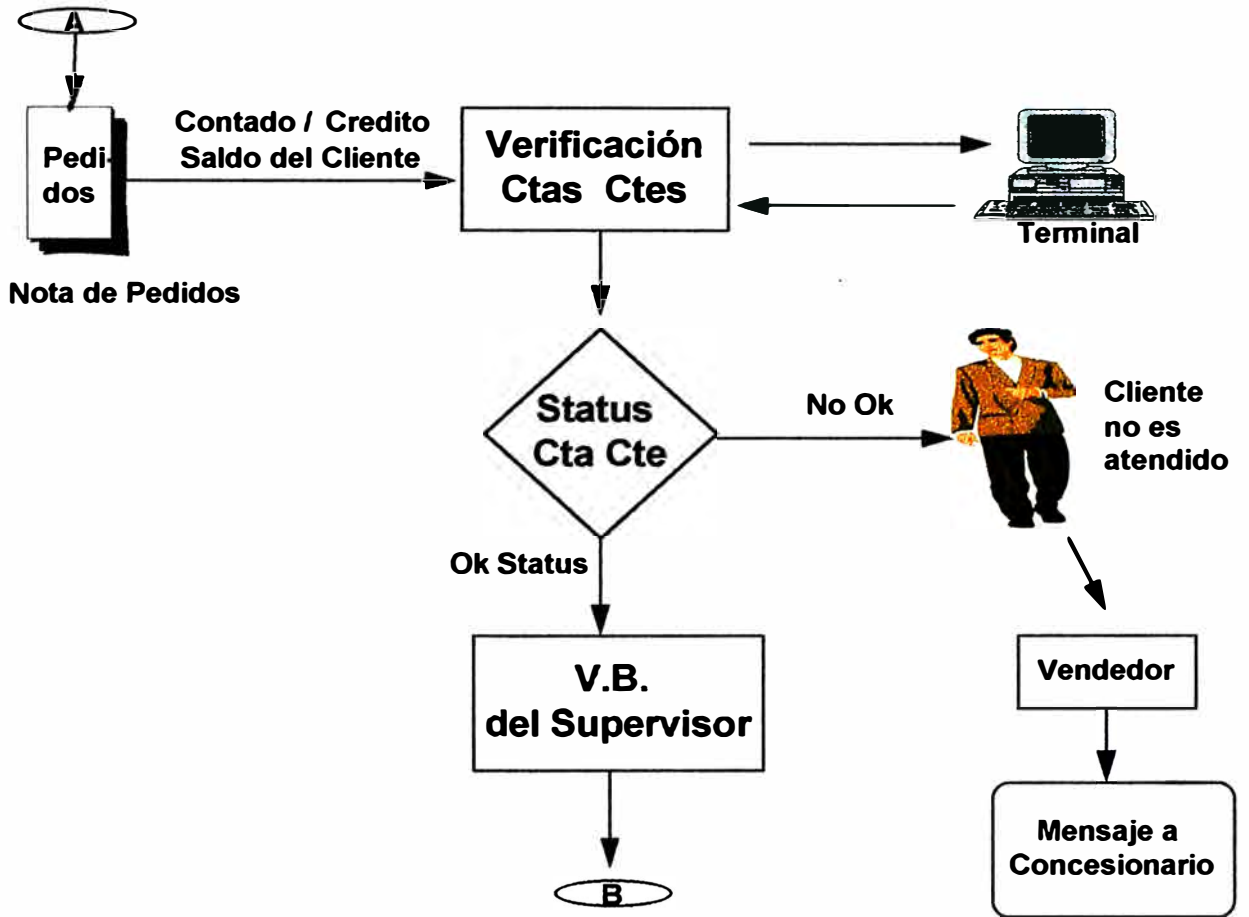


Fig.3

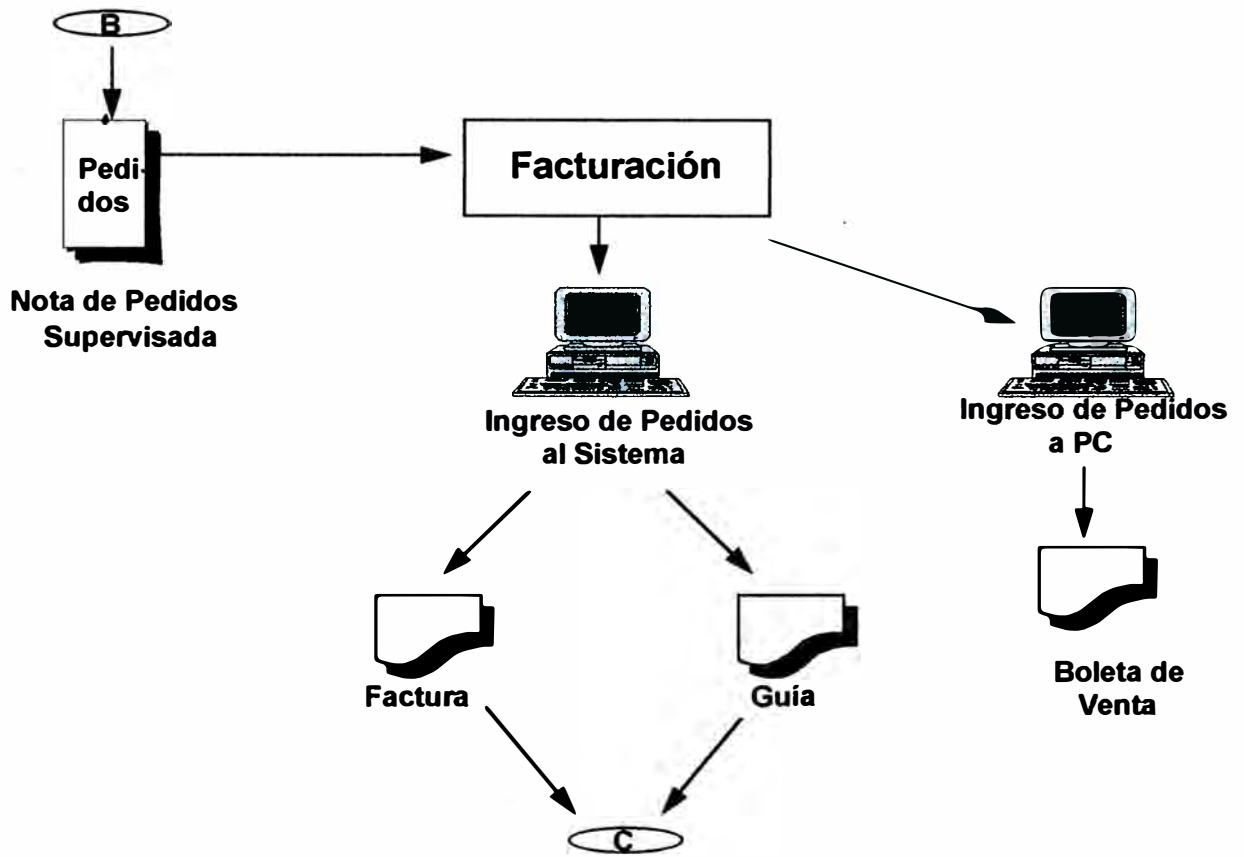


Fig.4

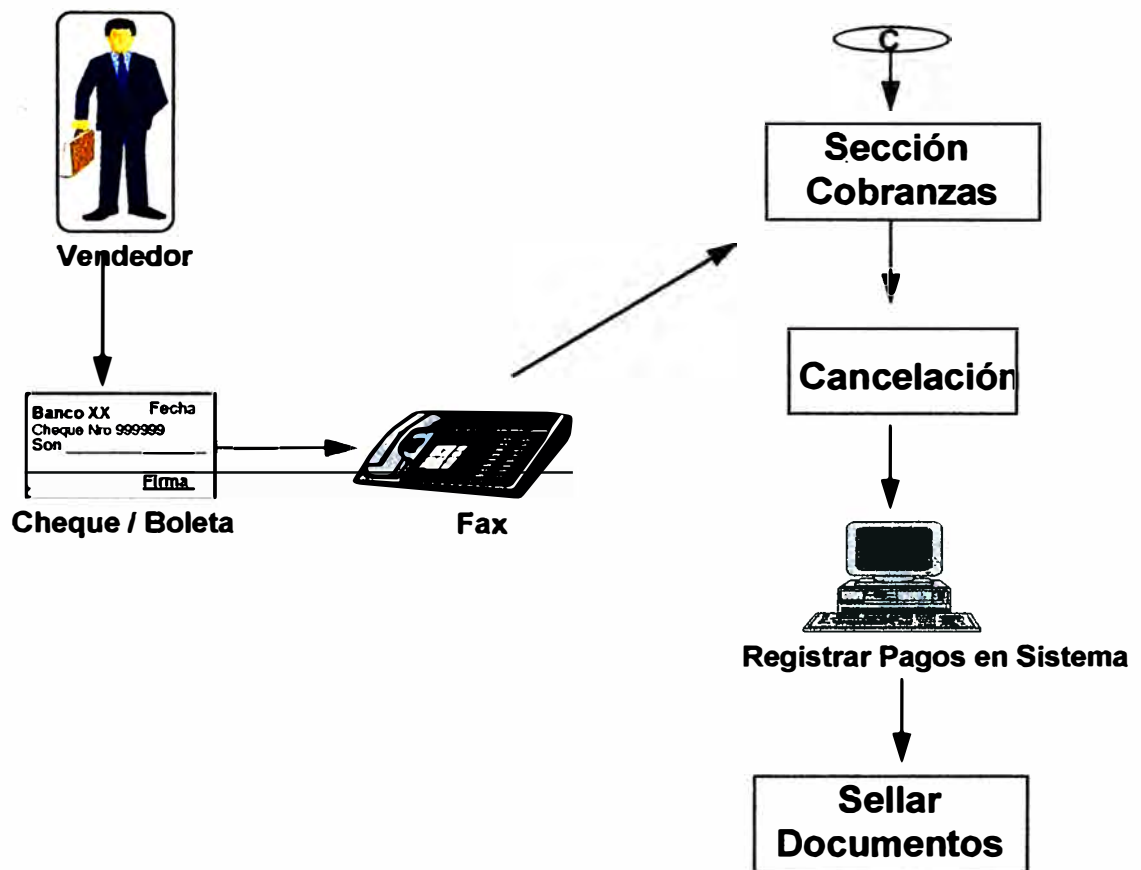


Fig.5

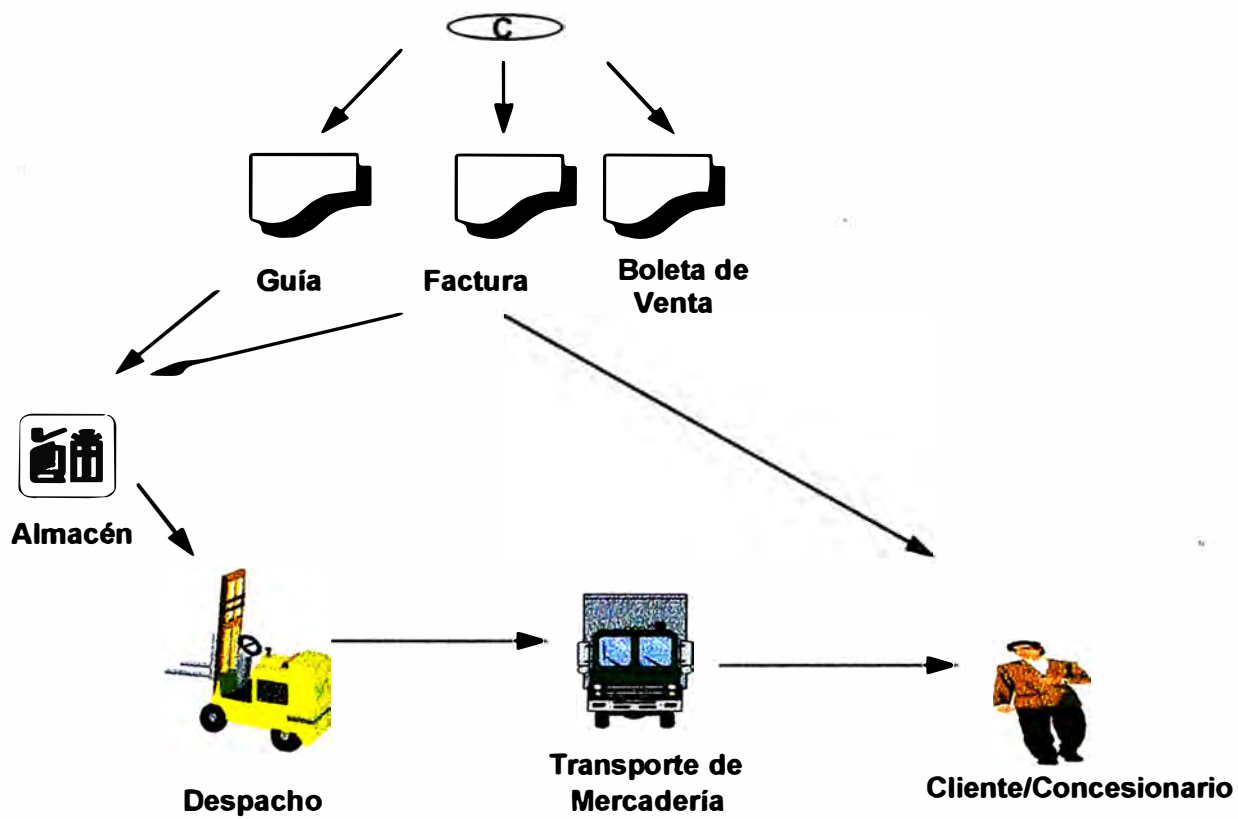


Fig.6

CAPITULO II

FUNDAMENTOS TEORICOS DE LA ARQUITECTURA

CLIENTE/SERVIDOR

La Arquitectura Cliente/Servidor se define como una tecnología que permite que los sistemas trabajados hasta la actualidad bajo sistemas propietarios o mainframes, puedan interconectarse a plataformas diferentes, tales como plataformas de redes con distintos servidores y estaciones de trabajo.

Por medio de esta tecnología se busca obtener una mejor interfaz con el usuario final mediante el desarrollo de aplicativos visuales, así como a trabajar bajo una gama de Sistemas Operativos diferentes y Bases de Datos heterogéneas, saltando así las barreras características que presentaban los “Sistemas Proprietarios”.

2.1 PRINCIPALES COMPONENTES.-

CLIENTE.- viene a ser el componente que inicia uno o múltiples requerimientos de trabajo. Puede ser por lo general una PC. El *Cliente* es el usuario de los datos y las aplicaciones y la ubicación de los mismos le es totalmente transparente.

SERVIDOR.- es cualquier recurso de cómputo dedicado a responder a los requerimientos del *Cliente*. Los “*Servers*” o *Servidores* pueden estar conectados a los *Cientes* a través de redes LANs o WANs para proveerles de múltiples servicios, tales como servicios de impresión, acceso a bases de datos, envíos de fax, procesamiento de imágenes y una gama cada vez más amplia de servicios.

MIDDLEWARE.- Es el software necesario para soportar las interacciones entre los Clientes y los Servidores. Podemos pensar en el middleware como el software que está en el medio del sistema cliente/servidor.

Actualmente en el mercado hay una gama de productos que caen en la categoría de Middleware y estos son:

- DCE,
- Named Pipes,
- LAN Manager,
- ONC+,
- LAN Server,
- Vines,
- Netware,
- TCP/IP,
- NetBIOS y
- APPC.

TCP/IP y Netware son middleware muy conocidos y utilizados en el entorno de sistemas de las compañías peruanas.

Entre los *Middleware* específicos para la conectividad vía Internet tenemos:

- HTTP,
- S-HTTP y
- SSL.

Y los *Middleware* para la conectividad con distintas Bases de Datos son:

- ODBC,
- DRDA,
- EDA/SQL,
- SAG/CLI y
- Oracle Glue.

En la actualidad podemos encontrar modelos Cliente/Servidor donde el Cliente, el Middleware y el Servidor pueden estar instalados y correr en un mismo computador (PC).

Una aplicación Cliente/Servidor se compone de varios procesos clientes y servidores que se pueden distribuir en una red. Como detallamos anteriormente es a través del Middleware que estos procesos clientes y servidores se conectan e

interactúan constituyendo aplicaciones. El Middleware agrupa una serie de funciones y servicios no incluidos en las aplicaciones ni en el sistema operativo.

Mientras que el Middleware define las plataformas y el nivel de transparencia de localización, las verdaderas ventajas de implantar C/S sólo se pueden conseguir mediante un adecuado diseño del sistema y de la aplicación, y con una adecuada elección de los elementos de desarrollo.

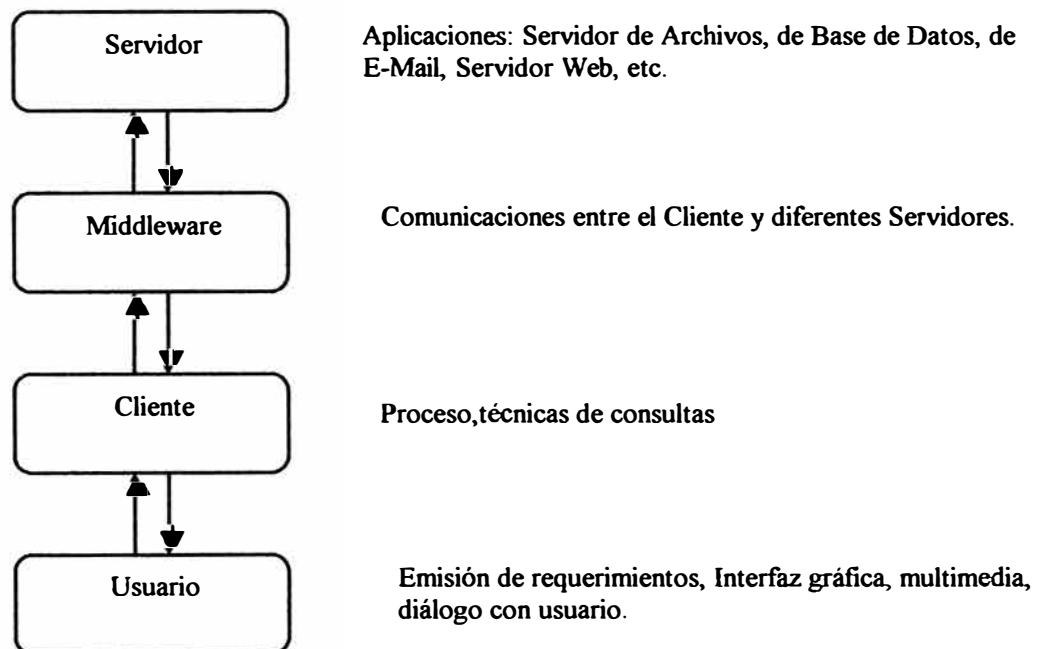


Fig. 7

2.2 Características de los Sistemas C/S.-

Entre las principales características de los Sistemas Cliente/Servidor que los distinguen de otras arquitecturas o modelos tenemos:

- ◆ Permite compartir recursos:

Un servidor puede servir a muchos clientes al mismo tiempo y regular sus

accesos a los recursos compartidos.

◆ Protocolos asimétricos:

Hay una relación de muchos-a-uno entre los clientes y el servidor.

◆ Transparencia de localización:

El servidor puede residir en la misma máquina con el cliente o en una diferente máquina a través de una red. Un software cliente/servidor generalmente enmascara la localización del servidor desde los clientes, redireccionando las llamadas de servicio, cuando ellos lo necesitan. Un programa puede ser un cliente, un servidor o ambos.

◆ Unión y combinación de diferentes plataformas:

El software cliente/servidor ideal es independiente del hardware o de las plataformas de software del sistema operativo.

◆ Encapsulación de servicios:

Un mensaje dice al servidor que servicio es requerido, luego el servidor determina como hacer el trabajo. Los servidores pueden ser mejorados sin afectar a los clientes puesto que la interface del mensaje no es cambiada.

◆ Escalabilidad:

Los sistemas cliente/servidor pueden ser escalados horizontalmente o verticalmente. La escalación horizontal significa adicionar o remover workstations clientes con sólo un pequeño impacto en la performance. La escalación vertical significa migrar a una más grande y rápida máquina servidora o multiservidores.

◆ Integridad:

En una Arquitectura Cliente/Servidor el código y la data del servidor está mantenida centralmente, lo cual resulta en un mantenimiento más barato y en la protección de la integridad de la data compartida, al mismo tiempo los clientes quedan personales e independientes.

2.3 Atributos de una aplicación C/S.-

Una aplicación C/S debe disponer de los siguientes atributos principales:

2.3.1 Separación de funciones.- Para poder distribuir partes de una aplicación entre los procesadores, su lógica debe dividirse en los siguientes módulos funcionales:

- Lógica de presentación
- Lógica de negocio
- Lógica de datos

Si se diseñan y se implantan de manera apropiada, los módulos de función pueden distribuirse y redistribuirse en distintos procesadores, y los flujos de trabajo pueden cambiarse en función de las necesidades del negocio.

2.3.2 Encapsulación de servicios.- Este atributo tiene el objeto de que el acceso a dichos módulos funcionales sea por clientes y servidores, las funciones deben encapsularse, es decir, rodearse de una interfase bien definida, accesible mediante mensajes o llamadas.

Un servidor puede ser considerado como un objeto. Este objeto, si se diseña correctamente, puede ser compartido por muchos procesos de clientes distintos y por diferentes aplicaciones.

2.4 Modelos de Distribución C/S .-

En un entorno C/S las aplicaciones se componen de partes de desarrollo propio y de aplicaciones disponibles comercialmente. La integración se consigue a través de los GUI, compartiendo datos comunes o mediante comunicación directa e intercambio entre aplicaciones. No existen reglas fijas sobre dónde dividir una aplicación. Como norma general, las funciones y los datos deben situarse lo más cerca posible del lugar donde sean necesarios para la operativa empresarial.

El objetivo de las aplicaciones C/S es conseguir una distribución de funciones entre los procesadores de una red departamental que permita el uso óptimo de los recursos. También hay modelos Cliente/Servidor que se basan en redes Ethernet. Las estadísticas nos informan que alrededor del 80% de las instalaciones Cliente/Servidor en todo el mundo, siguen este modelo, en el cual múltiples Clientes piden requerimientos a un Servidor local o a multiservidores.

Dependiendo de que funciones se distribuyan, distinguiremos los tres modelos de distribución siguientes:

- a) **Presentación Distribuida**, este modelo aísla la presentación del resto de la aplicación.
- b) **Datos Distribuidos**, este modelo separa el acceso a los datos del resto de la aplicación.
- c) **Función Distribuida**, este modelo permite asignar las funciones de la aplicación a distintos procesadores.

Como hemos visto, una aplicación C/S se compone de un número determinado de procesos cliente y servidor. Generalmente, los procesos cliente se ejecutan en una estación de trabajo y solicitan servicios de múltiples procesos servidor, locales o remotos, utilizando cualquiera de los tres modelos. En realidad, es

muy normal encontrar múltiples modelos de distribución dentro de una sola aplicación, y, desde luego, dentro de las distintas aplicaciones que se ejecutan en el entorno de un mismo sistema.

2.4.1 Presentación Distribuida.- En este modelo la parte de la aplicación que se ocupa de la presentación de los datos está situada remotamente respecto de la lógica del negocio y de los datos de la aplicación. La lógica de la presentación se desarrolla generalmente con algunas de las herramientas de desarrollo C/S. La Presentación Distribuida en su forma más simple puede consistir en una Interfaz gráfica de usuario para acceder a programas de proceso de transacciones ya existentes.

2.4.2 Datos Distribuidos.- Cuando se distribuyen los datos, su acceso y manipulación quedan separados del resto de la aplicación. Así podemos tener:

Archivos distribuidos: se utilizan principalmente para compartir los recursos de una red. Esta es la forma más sencilla de proceso distribuido que consiste en una redirección física de las solicitudes de lectura y escritura, con muchos mensajes fluyendo por la red para encontrar los datos pedidos.

Bases de Datos distribuidas: permiten utilizar de forma más eficiente el procesador que controla la base de datos, es decir el proceso cliente pasa las peticiones SQL en forma de mensajes al servidor de la base de datos.

2.4.3 Funciones Distribuidas.- Este modelo proporciona la máxima flexibilidad y permite a los departamentos de desarrollo un control total sobre donde situar las funciones en la red.

2.5 Categorías de los Servidores en una Arquitectura C/S.-

Por la naturaleza del servicio que brindan a los clientes, la Arquitectura Cliente/Servidor se puede dividir en las siguientes categorías:

2.5.1 File Servers.- Con este tipo de arquitectura, un *Cliente*, generalmente una PC, pasa requerimientos de registros de archivos sobre una red al *File Server*. Esta es una forma muy primitiva de servicio de datos que necesita muchos intercambios de mensajes sobre la red para encontrar la data requerida. Los *File Servers* son generalmente usados para compartir archivos a través de la red, ellos son indispensables para crear repositorios compartidos de documentos, imágenes, dibujos de ingeniería, etc.

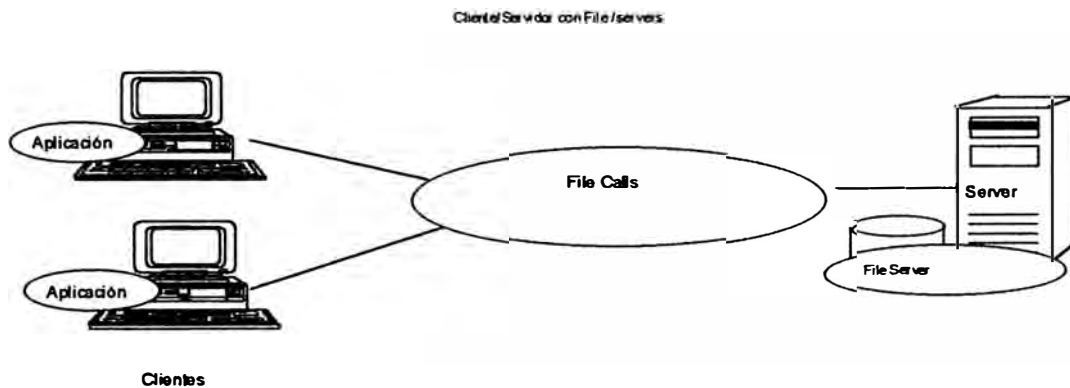


Fig. 8

2.5.2 Database Servers.- Con este tipo de arquitectura de Servidores de Bases de Datos, el cliente pasa requerimientos SQL -como mensajes- al servidor de bases de datos, luego los resultados de cada comando SQL son retornados sobre la red. El código que procesa el requerimiento SQL

y la data residen sobre la misma máquina o servidor. El servidor usa su propio sistema de procesamiento para encontrar la data requerida en vez de pasar otra vez todos los registros a un cliente y luego dejarles encontrar su propia data como fue el caso del File Server.

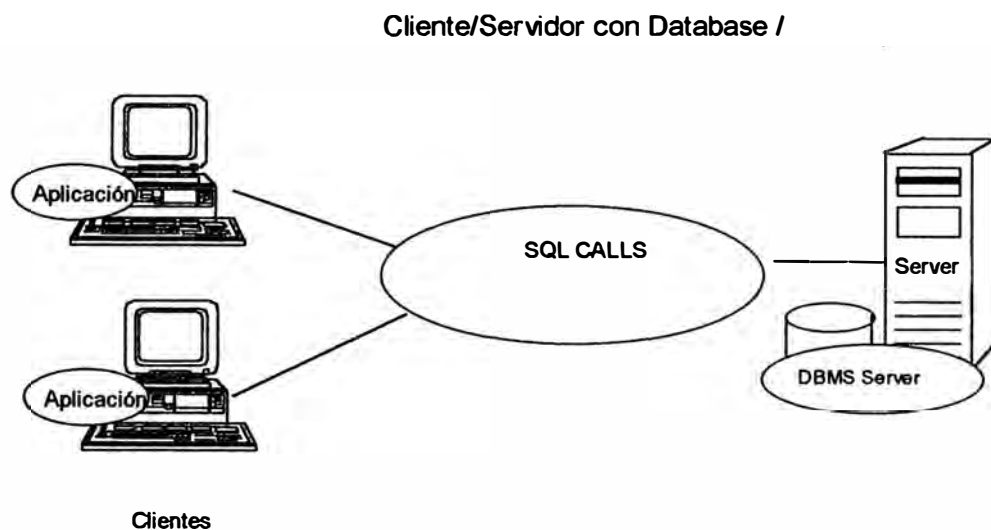


Fig. 9

2.5.3 Transaction Servers.- Mediante esta arquitectura se trabaja con un *Servidor de Transacciones* en el cual el cliente invoca procedimientos remotos que residen sobre el Servidor con un motor de base de datos SQL. Estos procedimientos remotos sobre el servidor ejecutan un grupo de sentencias SQL. La red de intercambio consiste de un simple mensaje requerimiento/respuesta. Las sentencias SQL sin embargo, tienen éxito o fallan como una unidad, estas sentencias agrupadas son llamadas “transacciones”.

Estas transacciones SQL son llamadas “Online Transaction Processing” (OLTP) - Procesamiento de Transacciones en Línea-, y tienden a ser aplicaciones de misión-crítica. Estas aplicaciones OLTP también requieren ajustar controles sobre la seguridad y la integridad de la base de datos.

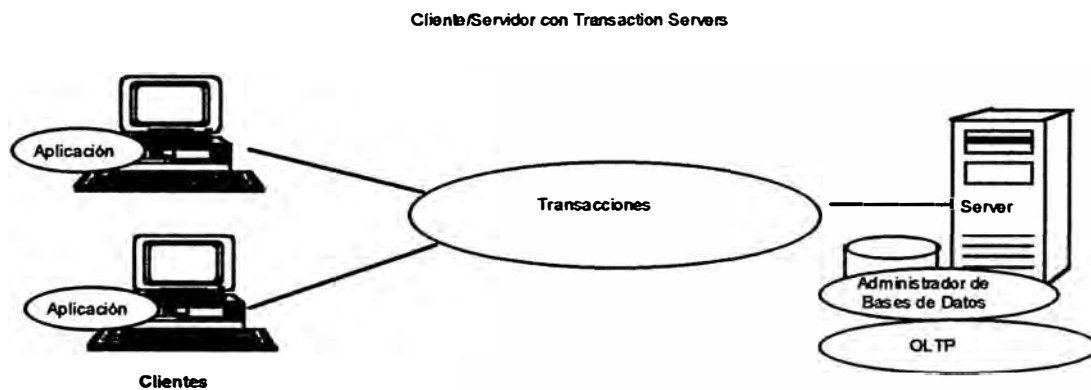


Fig. 10

Con un *Servidor de Transacciones* podemos crear la aplicación Cliente/Servidor escribiendo el código para ambos componentes tanto para el *cliente* como para el *servidor*. El *cliente* generalmente incluye una Interface Gráfica de Usuario (GUI), mientras que el *servidor* generalmente consiste de transacciones SQL contra una base de datos.

En la actualidad se conocen dos formas de OLTP:

- TP Lite, que está basado en stored procedures y,
- TP Heavy, que está basado en TP Monitors

2.5.4 GroupWare Servers.- Mediante esta arquitectura, el Servidor Groupware direcciona la administración de la información semi-estructurada tales como texto, imagen, email (correo electrónico), etc.

Estos sistemas cliente/servidor permiten la comunicación directa entre las personas, poniéndolas en contacto directo unas con otras. En esta arquitectura las aplicaciones son creadas utilizando un lenguaje script y las interfaces son provistas por el vendedor del Servidor Groupware, así como el Middleware de comunicación entre el *cliente* y el *servidor*. **Lotus Notes** es hoy en día el mejor ejemplo de este sistema.

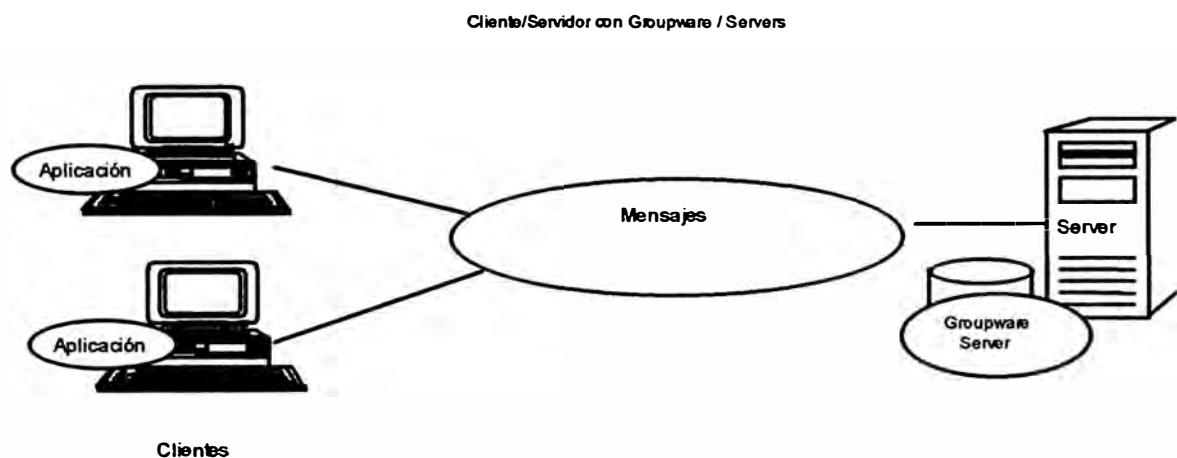


Fig. 11

2.5.5 Object Servers.- con un Object Server, la aplicación cliente/servidor es escrita como un conjunto de objetos que se comunican. Los objetos *cliente* se comunican con los objetos *servidores* usando un Middleware

específico para la Administración del sistema llamado **Object Request Broker (ORB)**.

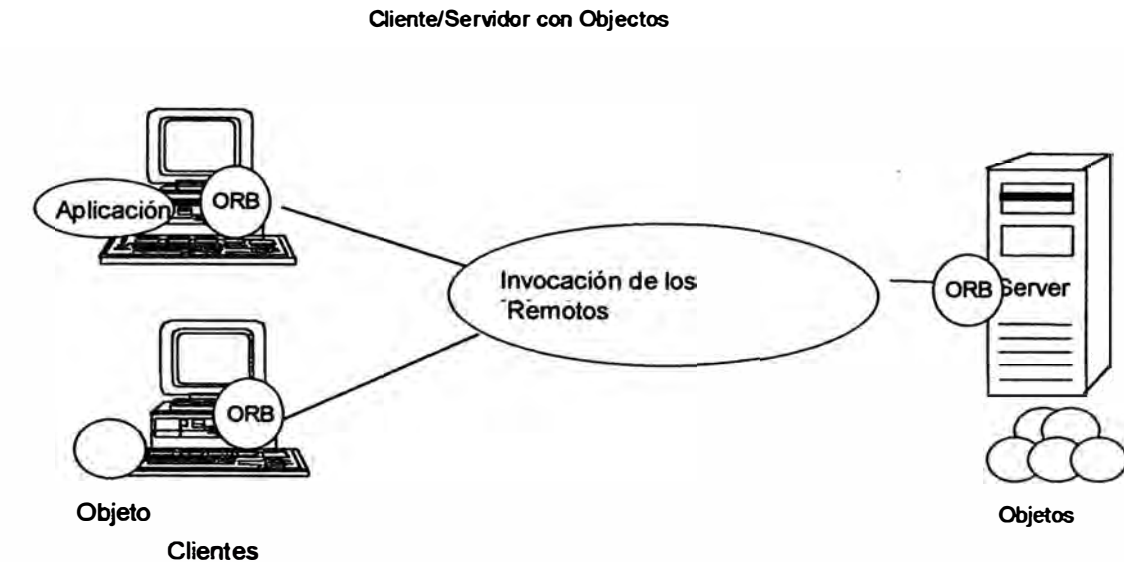


Fig. 12

Los objetos vienen a ser una combinación de datos y funciones, con propiedades características conocidas como polimorfismo, herencia y encapsulación.

Mediante este modelo Cliente/Servidor, el cliente invoca un método sobre un objeto remoto. El ORB localiza una instancia de esa clase del servidor objeto, invoca el método requerido, y retorna los resultados al objeto cliente. Los servidores objetos deben proveer soporte para la concurrencia y distribución, el ORB proporciona todo esto junto.

Entre los ORB comerciales que cumplen con el estándar de Grupos Administradores de Objetos (CORBA) tenemos los siguientes:

- ObjectBroker de Digital
- Orbix de Iona

- SOM 3.0 de IBM

- BlackWidow de PostModern

- NEO de Sun

- PowerBroker de Expersoft

- ORB Plus de HP

Distributed COM (DCOM) ó Network OLE de Microsoft.

2.5.6 Web Servers.- Mediante esta arquitectura, un Servidor Web retorna documentos cuando un cliente se los pide por nombres. Los clientes y servidores se comunican utilizando un protocolo llamado HTTP. Este protocolo define un simple conjunto de comandos y los parámetros son pasados como cadenas.

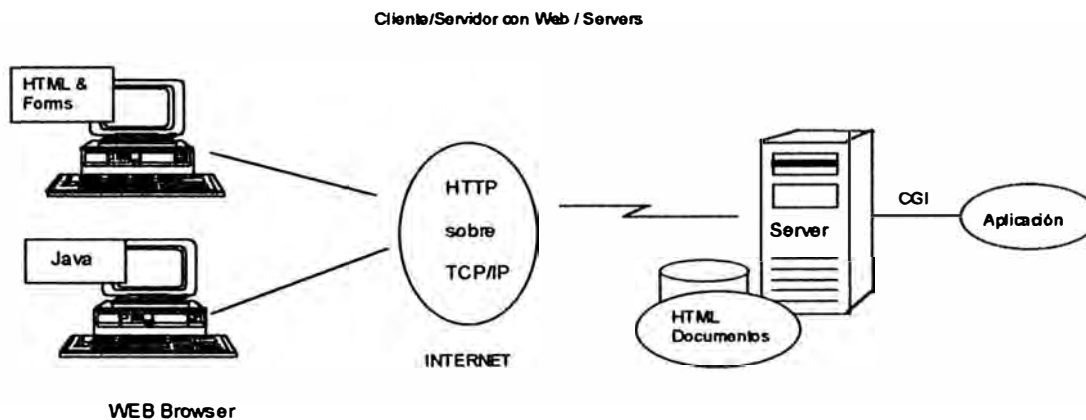


Fig. 13

Las estadísticas nos informan que en la actualidad el Web consiste de más de 100,000 servidores conectados soportando 500,000 home pages y 15 millones de usuarios (creciendo a un ritmo de 1 millón de usuarios por mes)

Hoy en día el Web está siendo extendido para proveer formas más interactivas de computación cliente/servidor. Como podemos apreciar actualmente, el Web y los Objetos distribuidos están empezando a ir juntos.

Los clientes en este modelo de Web Servers son los **Web browsers**.

Un Web browser es un cliente que interpreta la información que recibe un servidor y la displaya graficamente a un usuario. El cliente sólo interpreta los comandos del servidor y entrega los contenidos de una página HTML al usuario.

A continuación presentaremos una Tabla Descriptiva que muestra los cambios tecnológicos que han ido experimentando las Ciencias de la computación por décadas, hasta llegar a la aparición de la Arquitectura Cliente / Servidor.

Década	Tema	Tecnología
1950	Cerebro Electrónico	Lenguajes de Programación.
1960	Máquina de negocios	Sistemas Operativos.
1970	Corporación basada en datos	Terminales, BD, Redes.
1980	Ingeniería de Software	Herramientas CASE, computadoras personales, redes de área local.
1990	Cliente / Servidor	Computación Distribuida.

2.6 Administración de las Bases de Datos.-

El Sistema Administrador de Base de Datos Relacional del AS/400 es llamado DB2/400 y tiene su raíz en las implementaciones del S/36 y S/38 y de los conceptos de diccionarios de datos. El DB2/400 garantiza la integridad para todas las aplicaciones.

Según hemos podido constatar en las diferentes empresas donde hemos laborado y por estadísticas dadas, es que, en la mayoría de organizaciones el archivo plano es la norma, actualmente el 70% de la data no está constituida como base de datos, el 20% está en base de datos no relacional y un 10% de la data operativa si está guardada como base de datos relacional. Desafortunadamente el archivo plano aún es popular y la mayor parte de los Sistemas Administradores de Base de Datos Relacionales no pueden fácilmente usar esta data.

Por otro lado debe tomarse en cuenta que en la actualidad las bases de datos relacionales están empezando a utilizarse para depositar faxes, imágenes, archivos HTML, películas de sonido, mensajes e-mail y otros, por lo que su utilización se hará prioritaria en el futuro.

Actualmente se está viendo la forma de que las transacciones simples puedan actualizar múltiples BD distribuidas tales como DB/2, DB2/VM o DB2/400. Por ejemplo el DB2/400 ha eliminado los problemas de incompatibilidad que existían entre las bases de datos de AS/400 y DB/2. IBM también está considerando crear una librería de triggers para usuarios de AS/400.

Haciendo un poco de historia podemos decir que la primera generación de las aplicaciones C/S operaban en un ambiente de servidores de archivos o también

llamados File Servers, estos servidores tenían que ver con estructuras de Sistemas Administradores de Bases de Datos o simplemente archivos. Los servidores de archivos permitían el acceso a más de un usuario al mismo documento aunque no al mismo tiempo y permitían que un archivo o dato pudiera ser visualizado por cualquiera dentro de la red.

2.7 Arquitecturas de los Sistemas Administradores de Bases de Datos.-

Los tipos de sistemas de cómputo sobre los que corren las Bases de Datos están divididos en cuatro arquitecturas:

- a) Arquitectura Centralizada
- b) Arquitectura PC
- c) Arquitectura Cliente/Servidor
- d) Arquitectura Distribuida

a) Arquitectura Centralizada.-

En esta plataforma todo corre sobre un host (Mainframe o Minicomputador) incluyendo el propio Manejador de Base de Datos, también corren las aplicaciones que accesan la Base de Datos y las facilidades de comunicación que envían y reciben data desde los terminales de usuarios.

En un sistema centralizado los usuarios accesan la base de datos a través de terminales locales o remotos. Los sistemas basados en PC's tienen la habilidad de comunicarse con sistemas centralizados a través de

combinaciones de hardware y software que emulan los tipos de terminales que son comúnmente usados con un host particular.

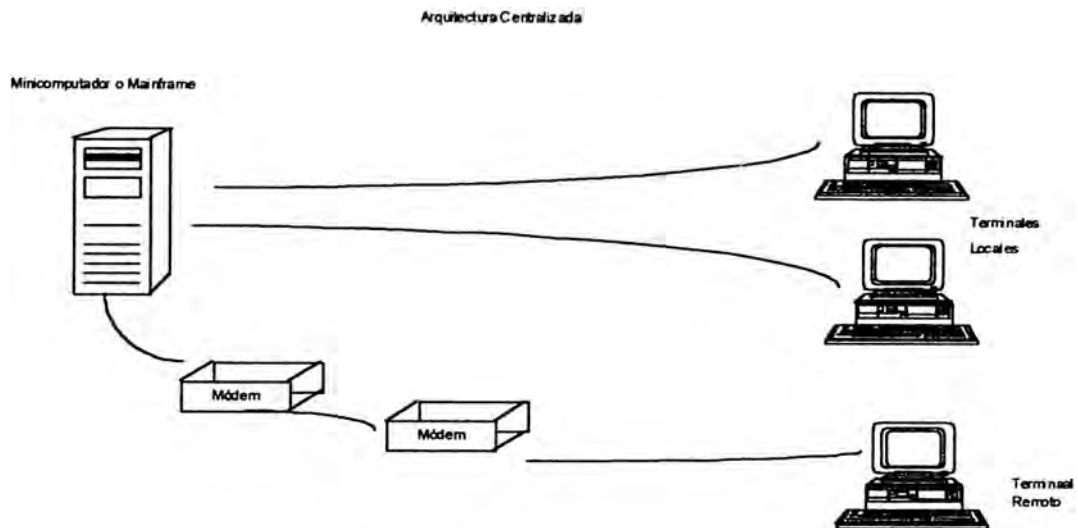


Fig. 15

Entre las Ventajas que presenta esta Arquitectura tenemos que la seguridad está centralizada y su habilidad de manejar enorme cantidad de datos. Además soporta numerosos usuarios simultáneamente, más de 1,000 al mismo tiempo.

Entre sus Desventajas están se encuentran los altos costos en la compra de hardware y mantenimiento del mismo.

b) Arquitectura PC LAN.- Esta arquitectura se da cuando un Administrador de Base de Datos corre sobre una PC y ésta actúa entre el host y el terminal. Las funciones separadas que son hechas por el Administrador de Base de Datos y por la Aplicación de Base de Datos son combinados dentro de una misma aplicación. Las aplicaciones de Base de Datos sobre una PC manejan las entradas y salidas.

Entre las desventajas de esta arquitectura es que el tráfico de la red se incrementa cuando muchos usuarios accesan a la Base de Datos.

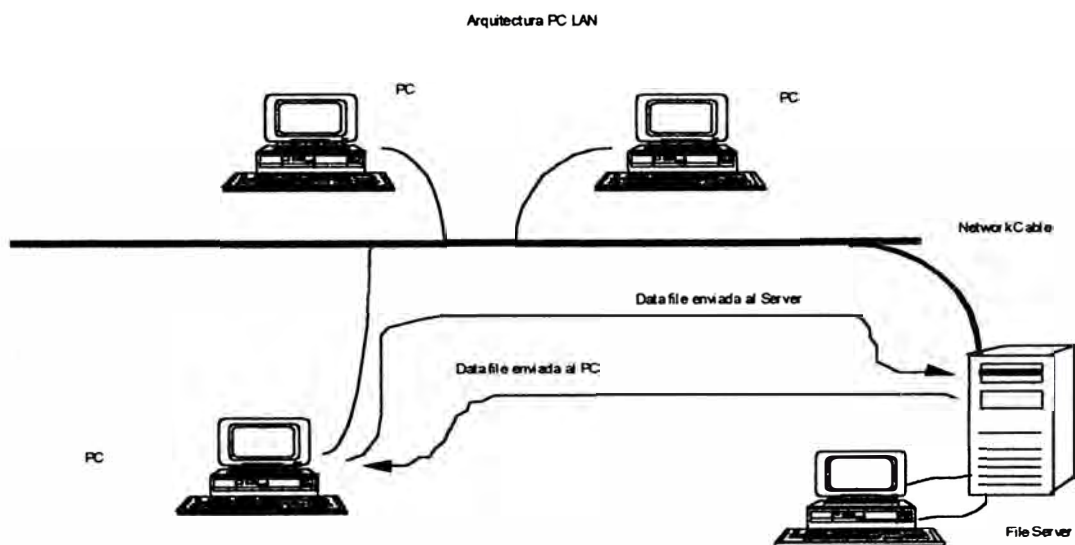


Fig. 16

c) **Arquitectura Cliente/Servidor.-** Divide el procesamiento de las Bases de Datos entre dos sistemas: El Cliente -generalmente una PC- corriendo la Aplicación de Bases de Datos y el Servidor de Bases de Datos que corre todo o parte del actual Administrador de Bases de Datos.

El sistema Back-End sobre el servidor maneja el procesamiento de datos. Por ejemplo un usuario sobre el Front-End crea un requerimiento, también conocido como query y la aplicación Front-End envía el requerimiento a través de la red al Servidor. El servidor de Base de Datos hace la búsqueda y envía de regreso sólo la data que cumple totalmente con el query del usuario.

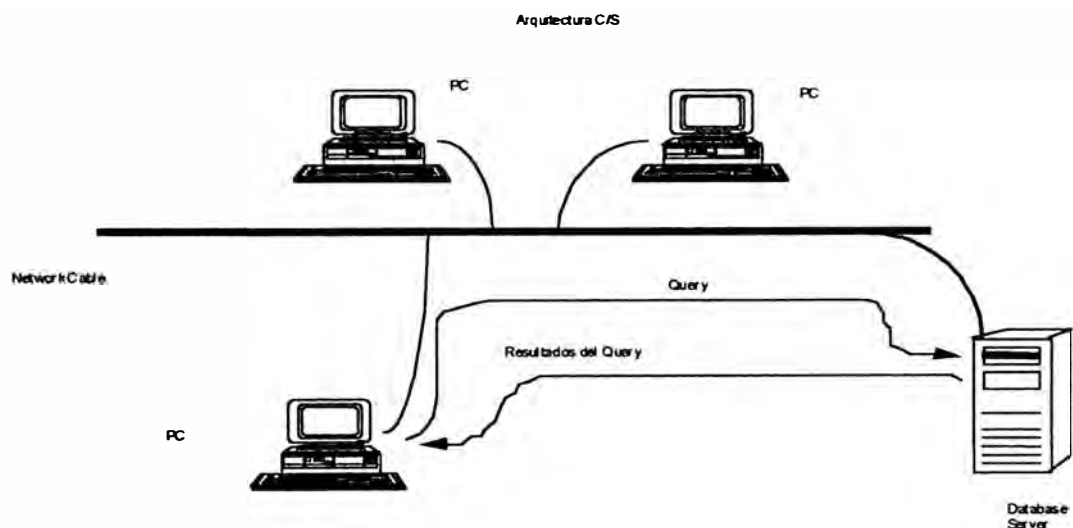


Fig. 17

La Arquitectura Cliente/Servidor incrementa el nivel de Procesamiento de las Bases de Datos separando al **Sistema Administrador de Base de Datos (DBMS)** de la **Aplicación de Bases de Datos**, la aplicación corre sobre una ó más estaciones de trabajo (generalmente PC's) y se comunica con una ó más DBMS que corren sobre otras computadoras o sobre una red.

Entre las Ventajas de esta arquitectura tenemos que, dividiendo el procesamiento entre 2 sistemas, la cantidad de tráfico de datos sobre el cable de red es reducida significativamente. También la performance se mejora corriendo el Administrador de la Base de Datos sobre un sistema más potente.

Entre la principal desventaja que encontramos en esta arquitectura es que requiere que la data sea depositada sobre un sistema simple y esto puede ser un problema para las grandes compañías.

d) Arquitectura Distribuida.- Mediante esta arquitectura se comparte data entre varios Hosts a través de conexiones directas sobre una red local o sobre conexiones remotas, vía teléfonos o líneas de datos dedicados.

Si un usuario requiere data desde el host local, el host local determina que éste no tiene la data y pasa los requerimientos por la red para conseguirla. El sistema envía luego la data de regreso al usuario, sin que el usuario sepa que la data fue recuperada desde un sistema diferente.

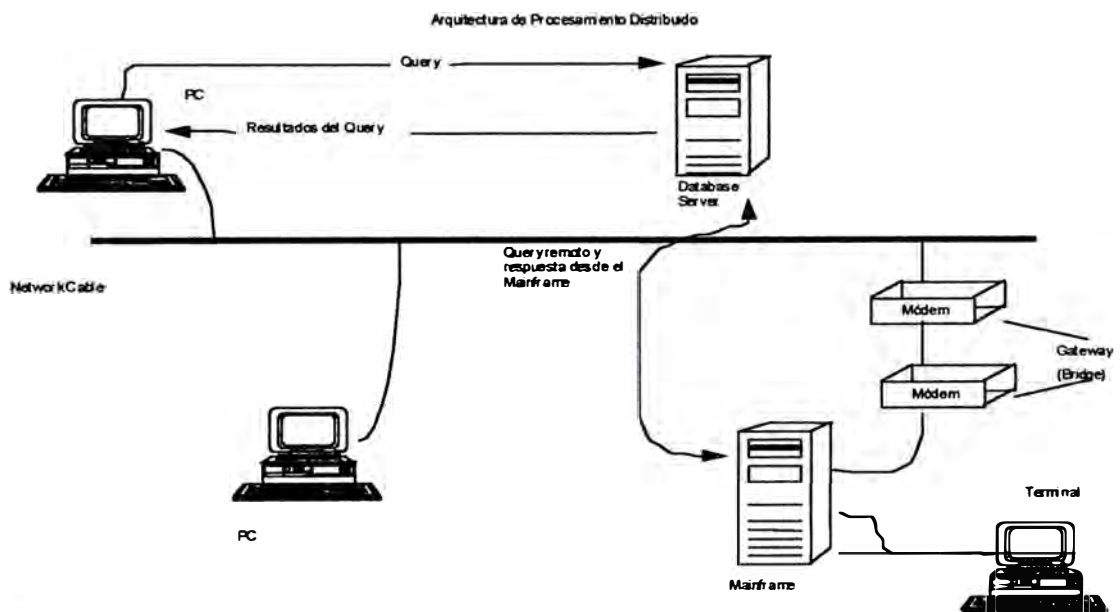


Fig. 18

2.8 SQL - STRUCTURED QUERY LANGUAGE.-

SQL fue diseñado explícitamente como un lenguaje de Bases de Datos Relacionales. La versión inicial del lenguaje apareció primero como **SEQUEL** a mediados de 1970 y fue desarrollado por IBM como el lenguaje estándar para acceder a una Base de Datos Relacional que corriera sobre Mainframes. A fines de los 70's el nombre se recortó a **SQL**.

Cuando el SQL es diseñado para Consultar Bases de datos interactivas, entonces es descrita como **DINAMIC SQL**, y cuando es usada como parte de una aplicación escrita en uno de los lenguajes procedimentales, entonces se denomina **EMBEDDED SQL**.

Usando el lenguaje de programación del SQL se pueden desarrollar un conjunto de instrucciones para una aplicación, tales como: **CREATE, UPDATE, SELECT, DELETE, INSERT INTO, ADD, COMMIT, MODIFY**, etc.

COMMIT

El comando **COMMIT** finaliza una transacción y hace permanentes todos los cambios ejecutados en una transacción. Para ejecutar este comando no se necesita ningún tipo de privilegios.

ROLLBACK

El propósito del Comando **Rollback** es rehacer el trabajo hecho en una transacción. Si la transacción no es ejecutada en su totalidad, el **ROLLBACK** permite realizar una reversión del efecto del procesamiento sin afectar las bases de datos.

Para ejecutar manualmente el Rollback no es necesario ningún tipo de privilegios. Si ya se hubiera hecho un Commit en una transacción, entonces se debe tener un Sistema de Privilegios FORCE TRANSACTION. Si el Commit fue hecho por otro usuario se debe tener el sistema de privilegios FORCE ANY TRANSACTION.

2.8.1 STORED PROCEDURES

Los Stored Procedures son sentencias SQL y de lógica procedural que es compilada, verificada, y depositada en el servidor de base de datos. El acceso al Stored Procedure es controlado a través de mecanismos de seguridad del servidor.

Los Stored Procedures aceptan parámetros de entrada tal que un simple procedimiento puede ser usado en la red para múltiples clientes usando diferentes datos de entrada (input data).

Por ello consideramos que los Stored Procedures simplifican el desarrollo de aplicaciones cuando se usan diferentes front-ends para acceder al servidor de bases de datos. Las aplicaciones front-end usan stored-procedures simplemente generando una llamada EXECUTE con el nombre del procedimiento y algunos parámetros que necesite para completar la operación. Así los desarrolladores de aplicaciones sólo tienen que cambiar los stored procedures del servidor y no de todas las aplicaciones cliente.

El más importante beneficio de usar Stored Procedures es que pueden mejorar la performance de la Base de Datos y disminuir el tráfico de la Red.

La familia DB2 de IBM V 2.1 implementa Stored Procedures como funciones DLL escritas en lenguaje de programación estándar. Los Stored Procedures residen en el mismo servidor como la base de datos, pero no son depositados dentro de la base de datos. Las aplicaciones Cliente llaman stored procedures DB2 que no necesitan saber que lenguaje fue usado para codificar el procedimiento.

IBM intenta lanzar una herramienta visual para construir stored procedures, triggers, y tipos de datos definidos por usuarios. Esta herramienta permitirá crear la lógica de procedimiento usando un lenguaje 4GL que es muy similar a Visual Basic de Microsoft.

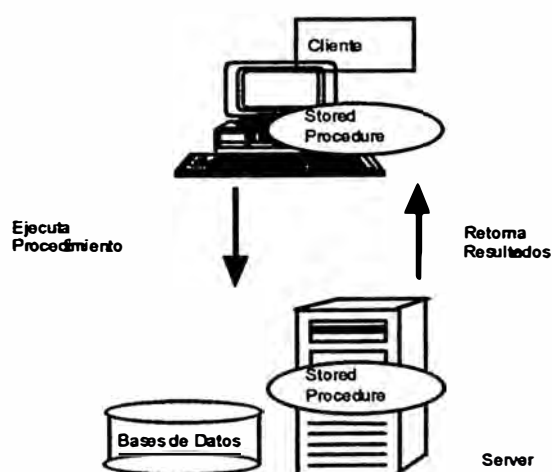


Fig. 19

Estructura de un Procedure

```
Create o Replace PROCEDURE P1 (  
Parametro IN Number)  
IS  
Var1 Number(x);  
Begin  
-----  
-----  
  
Exception  
-----  
  
End;
```

En el anexo A se presentan algunos ejemplos prácticos de STORED PROCEDURES.

2.8.2 TRIGGERS

Los triggers son una clase especial de stored procedures que automáticamente ejecutan cualquier data que sea insertada, actualizada o deleteada en la columna de la tabla con la cual el trigger esté asociado.

Los triggers son creados sólo sobre la base de datos actual, cada uno está asociado con una tabla particular y cada tabla puede tener triggers para insertar, actualizar y deletear comandos. Así como los stored procedures, los triggers son compilados cuando se ejecutan la primera vez o cuando el administrador de base de datos emite un comando explícito.

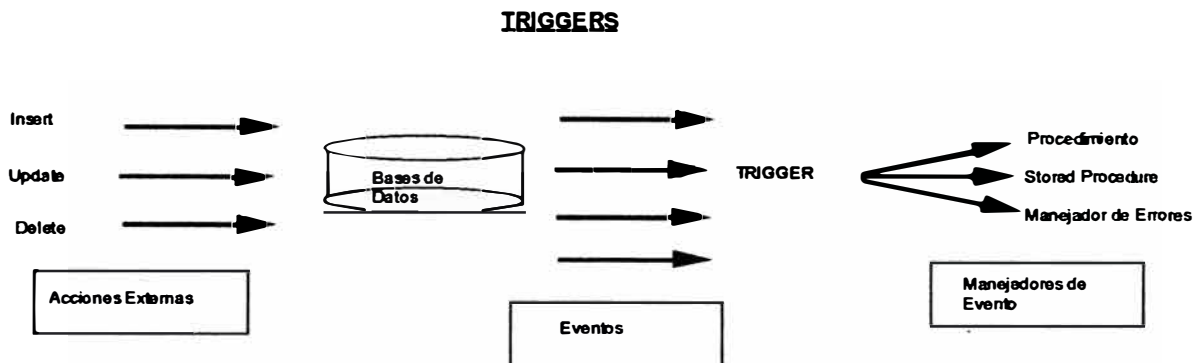


Fig. 20

En general los triggers pueden llamar a otros triggers o stored procedures. Lo que diferencia a los triggers de los stored procedures, es que los triggers son llamados implícitamente por eventos generados por la base de datos, mientras que los stored procedures son llamados explícitamente por las aplicaciones clientes.

Estructura de un Trigger

Triggering Event **BEFORE INSERT OR UPDATE OF rental_status
ON RENTALS**

**FOR EACH ROW
FOR EACH STATEMENT**

Trigger Restriction **WHEN (amount_paid IS NOT NULL)**

Trigger Body **DECLARE**

BEGIN

END;

En el anexo B se presentan algunos ejemplos prácticos de TRIGGERS.

2.9 PRINCIPALES ADMINISTRADORES DE BASES DE DATOS.- Según informes estadísticos del año 1995, Microsoft ha sido la empresa que más bases de datos ha vendido en todo el mundo, se estiman sus ventas en \$435 millones de dólares que se distribuyen de la siguiente manera entre sus productos:

SQL Server	\$ 85 millones
Access	\$300 millones
Visual FoxPro	\$85 millones

La empresa Gartner Group de Estados Unidos predice que para el año 2000 los cinco grandes vendedores de bases de datos serán IBM, Informix, Oracle y Sybase.

Las ventas de estos productos para el año 1995 fueron:

IBM	\$ 1,482 millones
Oracle	\$ 1,158 millones
Sybase	\$ 506 millones
Informix	\$ 320 millones

2.9.1 SQL SERVER 6.5.- SQL Server es un producto Microsoft que provee todas las características esenciales de una arquitectura Cliente/Servidor inteligente lo que permite realizar operaciones de procesamiento, distribución y protección de la información.

La nueva versión 6.5 incluye un Asistente Web que permite generar documentos HTML y Páginas Web rápidamente.

En un Sistema Cliente/Servidor basado en SQL Server, las estaciones de trabajo pueden ser PC's compatibles, Macintosh, estaciones Unix o combinaciones de éstas. En suma podemos decir que el SQL Server puede

correr sobre cualquier sistema operativo, como MS-DOS, Microsoft Windows, OS/2 de IBM ó quizás en un Sistema 7 de Apple.

El dispositivo Físico del SQL Server se denomina DEVICE y el dispositivo Lógico se denomina SEGMENTO.

Los volúmenes de información que puede manejar el SQL Server sólo están en relación directa al tamaño del dispositivo de almacenamiento que se utilice más no hay mayor restricción por el manejador en sí.

Las características principales del SQL Server se pueden resumir en las siguientes:

- 1) Su alto grado de interoperabilidad con otras plataformas, esto nos permitirá habilitar el acceso a la información a una amplia variedad de usuarios.
- 2) Utiliza Stored Procedures y Triggers que mejoran la performance y tiempo de respuesta de la red.
- 3) Permite crear DataWindows desde Stored Procedures.
- 4) Brinda un buen nivel de seguridad, las vistas lógicas del SQL Server son definidas en el servidor, permitiéndonos gobernar la seguridad del sistema, de los usuarios y de los módulos del sistema.

- 5) Es accesible al usuario, las vistas lógicas permiten que la información pueda ser distribuida globalmente sin comprometer la seguridad del sistema. Los usuarios pueden incrementar su productividad accediendo a toda la información disponible.
- 6) Incluye un Asistente Web que ayuda a especificar queries; esto permite generar documentos HTML. Estos queries corren en intervalos regulares para generar páginas Web.
- 7) La replicación usada como una componente integral de Microsoft SQL Server para distribuir automáticamente copias de sólo lectura de datos desde un servidor simple a uno ó más servidores destinos en uno ó más lugares remotos. Cada servidor destino puede recibir una parte o toda la data replicada.

Para que una data esté disponible para replicación, es necesario que se cree una “publicación”. El servidor que contiene las base de datos fuente y que hace que la data esté disponible para replicación es llamado **Servidor de Publicación o Publisher**. El Servidor de destino requiere y/o que recibe la data es llamado el **Servidor de Suscripción o Subscriber**.

a) Dispositivos y Bases de Datos del SQL Server.-

A continuación describiremos los dispositivos de base de datos y las bases de datos que el programa **Setup** crea cuando el SQL Server estará instalado:

- Creará los dispositivos de bases de datos MASTER, MSDBDATA y MSDBLOG.

- Creará las bases de datos *master*, *model*, *tempdb* y *pubs* y las guarda en el dispositivo de bases de datos MASTER.
- Creará la base de datos *msdb* y la guarda en el dispositivo de bases de datos MSDBDATA. El *msdb* es guardado en MSDBLOG.

A continuación describiremos cada base de datos.

Base de Datos	Descripción
Master	Contiene las tablas del sistema. Comprende el Diccionario de Datos del SQL Server.
Model	Provee un prototipo para las nuevas bases de datos de usuarios.
Msdb	Provee un área de depósito para la información sobre “schedules”.
Tempdb	Provee un área de depósito para tablas temporales y otras necesidades de guardar trabajos temporales.
Pubs	Es una base de datos simple provista como un herramienta de aprendizaje.

Base de Datos master

A continuación describiremos las tablas que se encuentran en la base de datos *master*.

Tabla	Que información deposita
Syscharsets	Cada conjunto de caracteres
Sysconfigures	Cada opción de configuración
Syscurconfigs	Opciones de configuración que están siendo usadas por SQL Server.
Sysdatabases	Cada base de datos sobre SQL Server.
Sysdevices	Cada dispositivo de base de datos.
Syslanguages	Cada lenguaje instalado.
Syslocks	Activa locks
Syslogins	SQL Server login ID
Sysmessages	Cada mensaje del sistema en SQL Server.
Sysprocesses	Los procesos del sistema.
Sysremotelogins	Usuarios remotos con nombres de usuarios locales.
Sysservers	Cada SQL Server remoto y local.
Sysusages	Cada dispositivo localizado en una base de datos.

Base de Datos model

A continuación describiremos las tablas que se encuentran en la base de datos *model*.

Tabla	Que información deposita
Sysalternates	Cada alias de un usuario SQL Server a un usuario de base de datos.
Sysarticles	Contiene una fila por cada uno de los artículos puestos por la publicación del servidor.
Syscolumns	Cada columna en una tabla o vista.
Syscomments	Cada vista, regla, default, trigger y stored procedure con una sentencia SQL.
Sysconstraints	Contiene una fila por cada PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT y constraint UNIQUE.
Sysdepends	Cada procedimiento, vista o tabla que está referenciada por un procedimiento, vista o trigger.
Sysindexes	Cada índice.
Syskeys	Cada foreign o primary key definida.
Syslogs	El transaction log.
Sysobjects	Cada tabla, vista, stored procedure, regla, trigger y objetos temporales.
Sysprocedures	Cada vista, regla, trigger y stored procedure.
Sysprotects	Permisos de los usuarios

Syspublications	Contiene una fila por cada una de las publicaciones puestas para el Servidor de Publicaciones.
Sysreferences	Contiene una fila por cada constraint FOREIGN KEY.
Syssegments	Por cada segmento.

Base de Datos msdb

A continuación describiremos las tablas que se encuentran en la base de datos *msdb*.

Tabla	Que información deposita
Sysbackupdetail	Especifica un sumario de los dispositivos usados para ejecutar un backup.
Sysbackuphistory	Especifica un sumario de cada operación backup.
Sysoperators	Deposita todos los administradores, tales como horarios de trabajo, nombres de e-mail, números de página, etc.
Sysalerts	Deposita todas las alertas definidas por los usuarios.
Systasks	Deposita todas las tareas definidas por los usuarios.
Syshistory	Toda la información histórica.
Sysnotifications	Deposita todas las alertas activadas.
Sysrestoredetail	Especifica un sumario de los dispositivos usados a
Sysrestorehistory	Especifica un sumario de cada operación a restaurar.

b) Características técnicas del SQL Server.-

Data distribuida y replicación	<ul style="list-style-type: none">• Llamadas de procedimiento remoto a la base de datos Server-to-server• Replicación de la base de datos para actualizaciones a bases de datos remotas• Soporta replicación de datos horizontal y vertical• Configuración gráfica
Soporte de Base de datos	<ul style="list-style-type: none">• Ejecución de transacciones paralelas, indexación y carga de datos, chequeo de la integridad de los datos.• Backup y restore paralelos de alta velocidad; soporta mas de 32 discos o Tape backups.
Seguridad	<ul style="list-style-type: none">• Un simple login ID para la red y la base de datos provee una mayor seguridad y reduce la complejidad administrativa.• Password encriptado• Se encriptan los stored procedures para preservar la integridad y seguridad del servidor.

2.9.2 ORACLE 8.- Es en la actualidad uno de los Manejadores de Bases de Datos más robustos en el mercado, no sólo por su excelente performance y sus altos niveles de seguridad en el manejo de la data, sino también porque cuenta con un conjunto de herramientas CASE y de desarrollo que se está abriendo paso para el desarrollo de aplicativos visuales.

Es así que en el Perú instituciones públicas que manejan grandes volúmenes de información y que requieren de altos niveles de seguridad como SUNAD y CONTRALORIA GRAL. DE LA REPUBLICA ya han optado por Oracle.

Oracle al igual que el SQL Server soporta Stored Procedures y Triggers.

El dispositivo Físico del Oracle se denomina TABLE SPACE y el dispositivo Lógico se denomina SEGMENTO.

DESIGNER es el CASE de Oracle que ofrece una gama de herramientas para las fases de Análisis, Diseño y Desarrollo.

El Upper Case del Designer cuenta con el Modelador de Procesos, el Modelador Jerárquico Funcional y el Modelador de Flujo de Datos.

El Middle Case del Designer cuenta con el Diagramador Entidad/Relacional, el DataBase Wizard y el Application Wizard que convierte las entidades en Tablas guardándolas en un Repositorio de Base de Datos

El Lower Case del Designer cuenta con el Data Schema, Module Logic, Module Data y Module Structure.

DEVELOPER 2000 es la herramienta visual de Desarrollo de Oracle que ha reemplazado al Oracle de Caracteres con que se programaba hasta hace unos años. Developer 2000 trabaja con Forms o formatos de pantalla que están

ligados a un WINDOWS. En un Form pueden haber varios CANVAS y en cada Canvas pueden haber varios BLOQUES de TABLAS.

Es esta distribución de las herramientas propias del Developer lo que dificulta un tanto la programación.

2.9.3 INFORMIX.- Es la compañía de Bases de Datos de crecimiento de más rápido en el mercado UNIX, cuenta con un crecimiento en ventas de licencias de 50% anual. Informix al igual que otros manejadores de bases de datos también soporta stored procedures y triggers y se ha convertido en un fuerte competidor de Oracle.

CAPITULO III

SOLUCION CLIENTE/SERVIDOR

3.1 Principales Objetivos Organizacionales.-

Según Kast y Rosenzweig en su libro “Administración en las Organizaciones” -Enfoque de sistemas y de Contingencias, los objetivos de las organizaciones se refieren a “propósitos y condiciones deseados que la organización busca como una entidad individual”.

Según el Profesor Lauterborn se debe hablar hoy en día de las cuatro C's y ya no de las cuatro P's como se consideraba años atrás. Estos nuevos conceptos de márketing dan prioridad al cliente o consumidor, considerándolos como la razón de ser de un negocio. Esto es se deben reemplazar los siguientes términos en un negocio:

- **Producto** por → **Consumidor**
- **Precio** por → **Costo**
- **Plaza** por → **Conveniencia**
- **Promoción** por --> **Conocimiento**

El tema del Márketing se encuentra muy en voga en nuestros días y está siendo tomado por las empresas peruanas con mucha seriedad. Debemos tener claro que el márketing basado en el conocimiento requiere que la empresa domine la tecnología en la que compite, conozca a sus clientes y a las nuevas fuentes de tecnología que pueden alterar su entorno competitivo y el de su propia organización.

El nuevo concepto de marketing que asume hoy en día la compañía tiene por lo tanto dos objetivos básicos:

- 1) Integrar al cliente en la empresa**
- 2) Crear y sostener una relación sólida entre la empresa y el cliente.**

La compañía en estudio como sistema abierto que es, se encuentra inmersa en un medio donde el suprasistema ambiental determinará las respuestas y los medios alternativos para el logro de sus objetivos, así como los objetivos mismos que se trace.

A continuación describiremos los principales objetivos organizacionales de la compañía forjados en base al nuevo concepto de marketing:

1. Reforzar su posición en el área.

Esto es, lograr que la compañía siga manteniéndose como líder en ventas en el mercado Peruano en el rubro de Autopartes.

1. Innovación en el campo de Marketing.

Dado el clima de competitividad la compañía debe renovar constantemente sus productos y servicios, siendo el elemento tiempo, uno de los factores críticos de éxito. En este sentido, el tiempo de puesta en mercado (“ *time to market* “) es primordial por lo que debe reestructurarse el negocio para reaccionar mejor a las necesidades del mercado.

Estos nuevos conceptos de marketing trazados por la compañía nos conllevan a replantearnos estrategias en el ámbito de los Sistemas de Información y de la Plataforma Computacional con que se viene trabajando tal que el Area de Sistemas pueda cubrir estos nuevos requerimientos empresariales.

Las estructuras organizativas con mayor autonomía en sus líneas de negocios y departamentos, serán responsables de las soluciones informáticas en las que se apoyan. Si el Area de sistemas no es capaz de reaccionar ante esta nueva demanda es probable que los departamentos y las líneas de negocio incorporen soluciones independientes fuera del control de la organización informática. Es obvio que la proliferación de soluciones departamentales independientes desembocaría en un caos si esto ocurriera.

Por lo tanto se necesita de una amplia infraestructura informática en el ámbito de la compañía que sirva de base a los departamentos para construir sus propias soluciones. Esta infraestructura deberá utilizarse para desarrollar o rediseñar aplicaciones que soporten los objetivos del negocio o los potencien.

3.2 Objetivos de la Solución C/S.-

La solución Cliente/Servidor que planteamos en esta Tesis debe cubrir los principales objetivos organizacionales de la compañía basándose en tecnologías factibles de ser implementadas y adecuadas a la realidad peruana.

Entre los requerimientos que hemos logrado identificar y que de ser solucionados le permitirán alcanzar sus objetivos organizacionales tenemos:

- a) Lograr un manejo más ágil y eficiente de la información que permita la toma de decisiones empresariales de manera óptima.
- b) La compañía necesita aplicativos con interfases, comunicación y conectividad que estén específicamente diseñados para cubrir sus funciones.
- c) Aplicar el nuevo concepto de Márketing desarrollado por el Departamento de Ventas de la compañía, este concepto consiste en lograr una comunicación global con los clientes, concesionarios y vendedores, llevándose a cabo esta interacción de una manera más ágil y continua.
- d) El nuevo concepto de Márketing en la compañía define también que cada Concesionario en cualquier punto del país debe estar interconectado vía Internet a la información de la compañía, así como también tener acceso a Correo Electrónico para mantenerse en permanente contacto con el Area de Ventas de la compañía.
- e) Proveer a los clientes y a toda la Red de Concesionarios, de una información totalmente actualizada y oportuna en el momento en que la requieran. Esta información será concerniente por ejemplo a: precios de ventas, stocks actualizados, líneas de crédito por cliente, comisiones y

otros, mediante software desarrollados e implantados en su centro de ventas.

- f) Las Areas Gerenciales de la compañía necesitan contar diariamente con información global de ventas, estados financieros, estados contables, cuadros estadísticos, etc.

Estas exigencias de las áreas conllevan a plantearnos una Solución Cliente/Servidor dadas las siguientes condiciones:

- Cambios estructurales y organizativos en la compañía.
- Respuesta a la dinámica del mercado.
- Cambios de los procesos en la compañía.
- Por la demanda de sistemas más fáciles de usar que contribuyan a una mayor productividad y calidad.
- Por el Precio/Rendimiento de las estaciones de trabajo y de los servidores.
- Por la creciente necesidad de la información para tomar decisiones y de soportar los procesos mediante unas aplicaciones más ajustadas a la estructura organizativa de la empresa, que permita realizar las operaciones de forma más natural.
- Por la utilización de nuevas tecnologías y herramientas de alta productividad más aptas para la dinámica del mercado.
- Por dar una respuesta oportuna a la dinámica del mercado
- Por los cambios en los procesos del negocio

3.3 Fases de la Solución planteada.-

La Solución Cliente/Servidor que planteamos se desarrollará en cuatro fases durante un intervalo de tiempo de ocho meses. Estas fases comprenden lo siguiente:

FASE I

- Instalación de una Red LAN bajo Windows NT.
- Esta Red LAN contará con cuatro Servidores dedicados y 50 estaciones Clientes.
- La duración de esta fase será de 25 días, para lo cual se recomienda contratar los servicios de una empresa especializada en instalación de redes.

FASE II

- Instalación en el Servidor de Base de Datos del SQL Server 6.5.
- El tiempo de instalación y pruebas del SQL Server no debe ser superior a 5 días.
- Instalación del Groupware Lotus Notes 4.0, esta instalación se llevará a cabo en el servidor destinado al Sistema de Mensajería.
- El tiempo de instalación y pruebas de Lotus Notes 4.0 no debe ser superior a 15 días.
- La duración de esta fase será de 20 días.

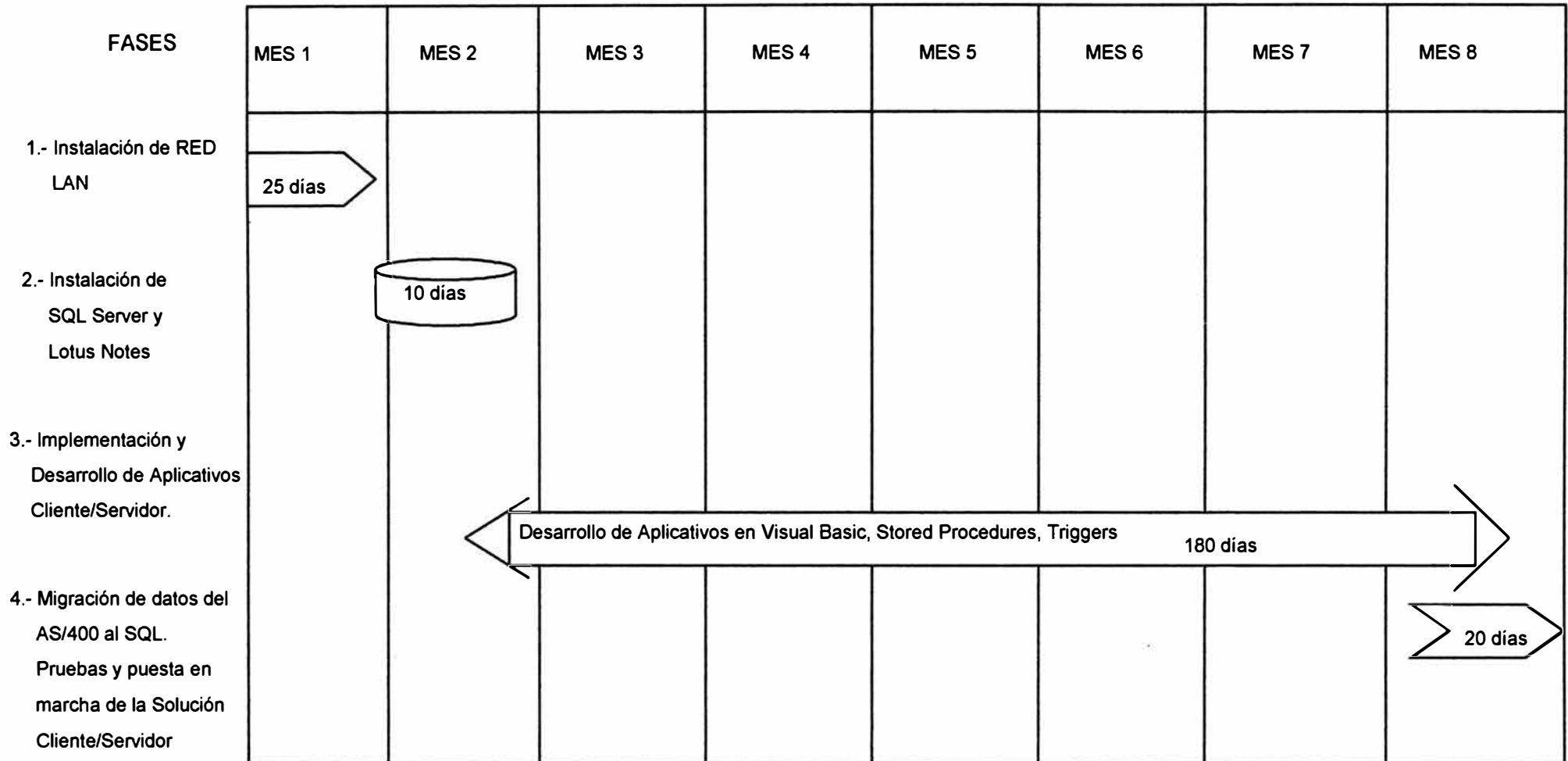
FASE III

- Identificación de los Aplicativos neurálgicos o críticos del sistema actual a desarrollarse en la plataforma cliente/servidor.
- El tiempo estimado para el desarrollo de los aplicativos será de 6 meses.

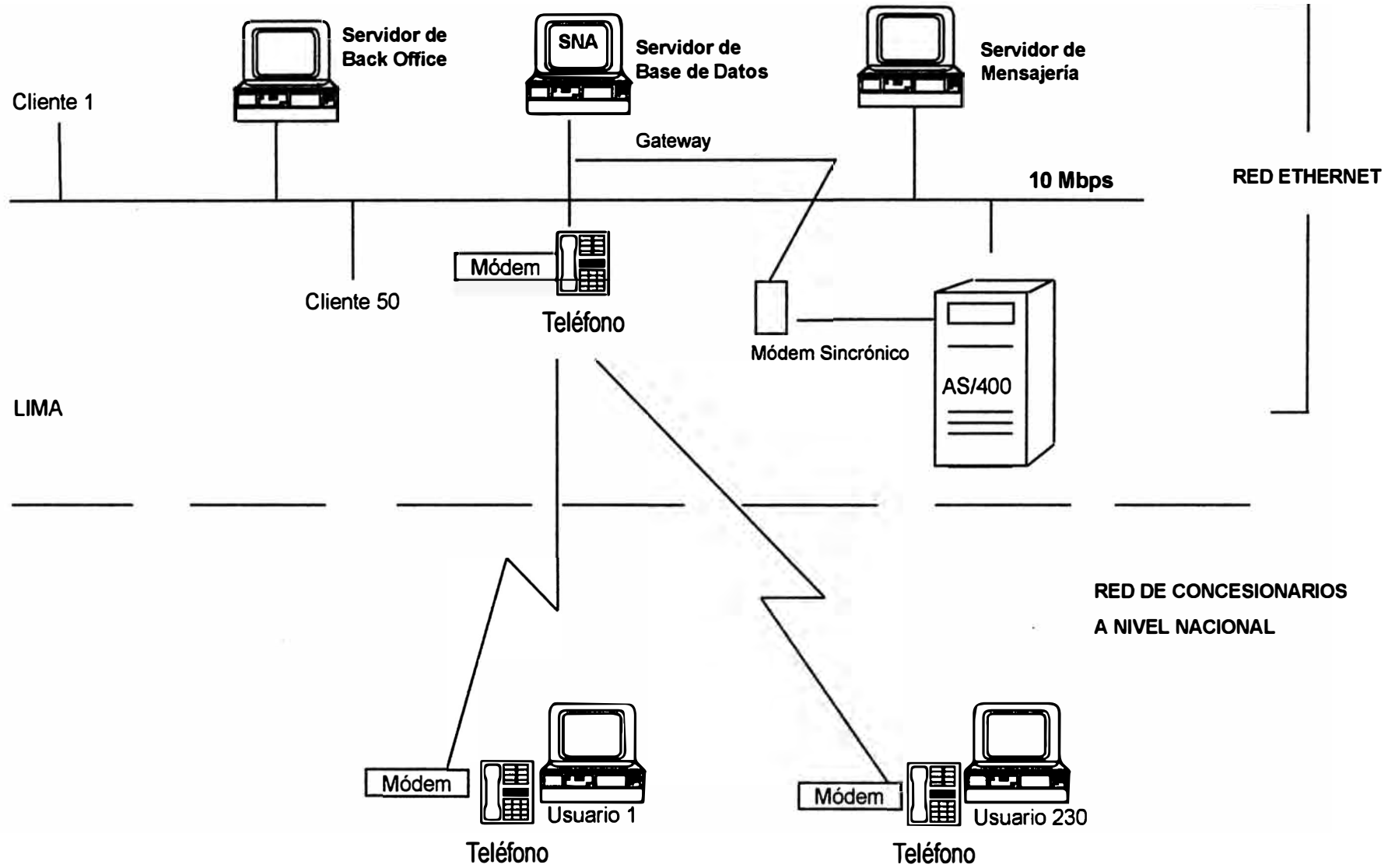
FASE IV

- Modelamiento de las estructuras de los archivos y tablas en el SQL Server con la ayuda del case ERWIN, se tomará como base las estructuras de los archivos del Sistema AS/400. El tiempo destinado será de 10 días.
- Migración de Datos del AS/400 al SQL Server
- El tiempo de migración y pruebas de datos se realizará en unos 10 días
- Pruebas y puesta en marcha de la solución Cliente/Servidor.
- El tiempo aproximado para esta fase será de 20 días.

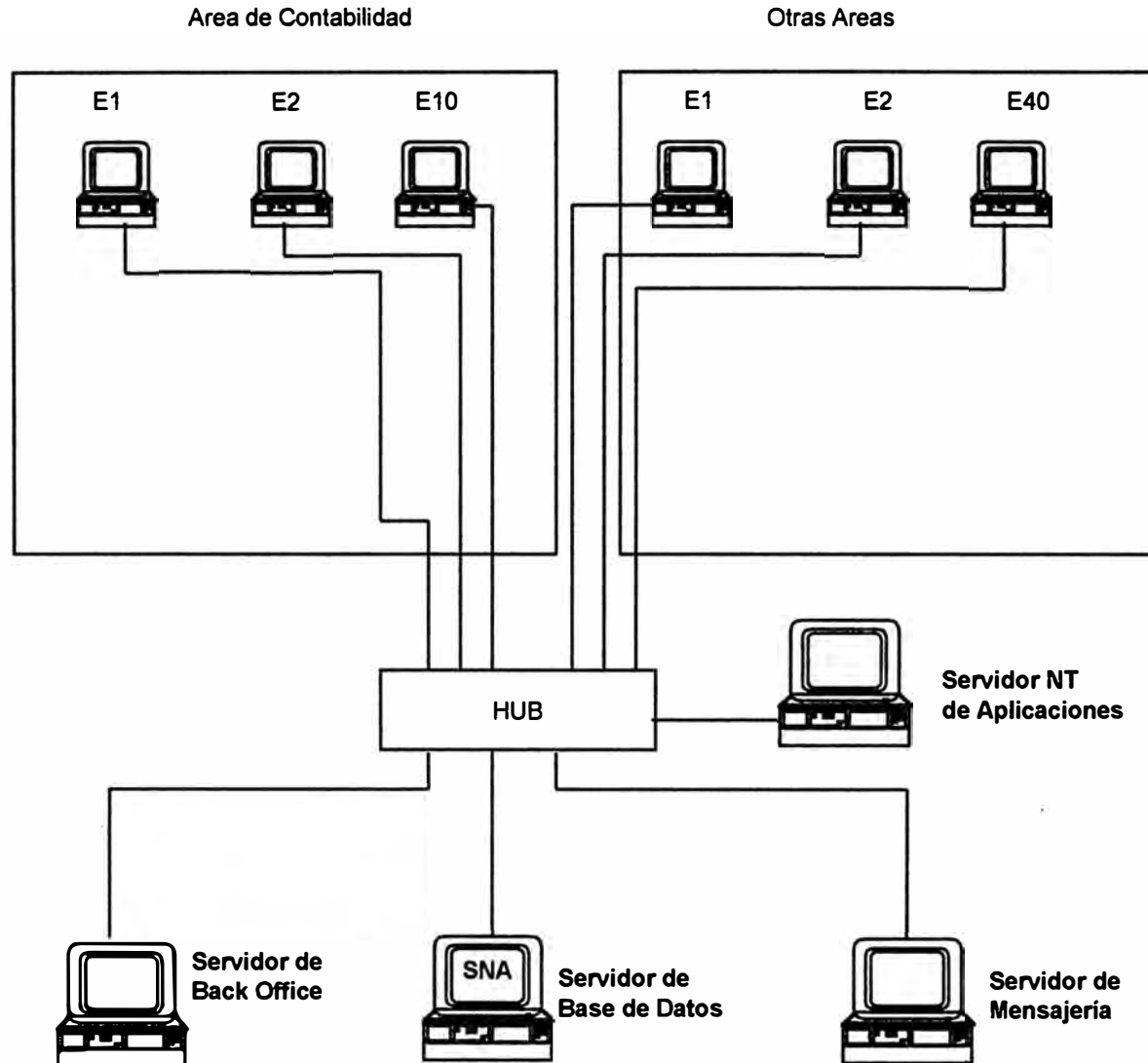
**DIAGRAMA DE GANTT INTEGRAL PARA LA
SOLUCION CLIENTE/SERVIDOR**



Plataforma Propuesta



Plataforma Propuesta de Servidores y Estaciones - LIMA



Leyenda
E1= Estación 1

3.4 Características principales del Hardware .-

Enseguida detallaremos los principales componentes hardware para la arquitectura Cliente Servidor que planteamos.

3.4.1 Características de la Red LAN.-

Red LAN
Arquitectura Ethernet
Conexión mediante 3 Hubs Apilados
50 estaciones cliente
Un Servidor NT para las estaciones cliente de todas las áreas.
Un Servidor de Back Office para los sistemas y archivos de oficina, bajo Microsoft Office.
Un Servidor de Mensajería o Groupware Server donde se instalará el Lotus Notes.
Un Servidor de Datos o Database Server donde se instalará el Manejador SQL Server 6.5 bajo SNA Server.
Conexión PCs con el sistema IBM AS/400 bajo SNA Server

3.4.2 Características físicas del Servidor SQL Server.-

Las características físicas con que debe contar el Servidor SQL Server serán detalladas a continuación:

Componente	Requerimientos
CPU	80486 Intel, ó Pentium.
RAM	Un mínimo de 16 MB ó un mínimo de 32 MB requerido para replicación.
Hard Disk	Un mínimo de 60 MB. Adicional 15 MB para SQL Server Books Online.
CD-ROM	Es requerido un CD ROM para instalar el software SQL Server.
Sistema Operativo	Windows NT Server 3.51 para SQL Server.

3.4.3 Características físicas de las estaciones Cliente.-

Las características físicas con que deben contar las máquinas que desempeñarán el papel de Clientes serán:

Componente	Requerimientos
CPU	80486 Intel, ó Pentium.
Hard Disk	Un mínimo de 8 MB. Adicional 15 MB para SQL Server Books Online.
Sistema Operativo	Windows 95

Todos los **Cientes** en la red LAN estarán bajo el sistema operativo Windows 95 y para su instalación se deberá correr el programa **setup**. El programa setup crea los siguientes directorios y archivos:

Directorios y Archivos	
C:\MSSQLBINN	Windows NT –archivos ejecutables.
C:\MSSQLINSTALL	Archivos SQL Server Books Online.

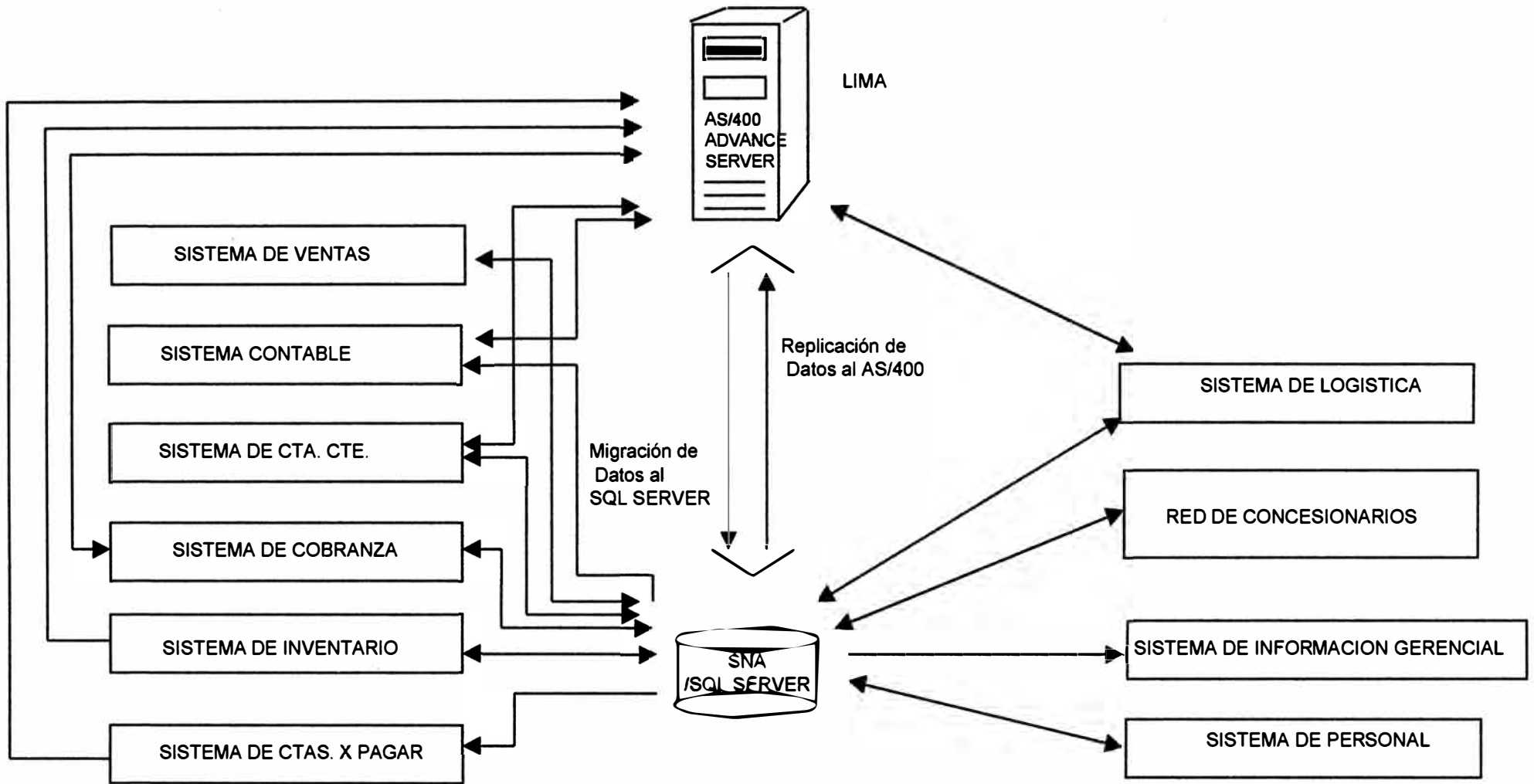
3.4.4 Servidor Host IBM AS/400.-

El Sistema IBM AS/400 se continuará utilizando como el servidor principal, hasta la finalización de la Cuarta Fase de nuestro proyecto dado que los aplicativos desarrollados en el lenguaje RPG 400 actualmente dan soporte a toda la compañía. Consideramos que en una primera etapa se va a replicar la data al Servidor SQL hasta que paulatinamente todos los aplicativos se hayan terminado de desarrollar con herramientas Cliente/Servidor.

Para la migración de datos se utilizará el CASE Erwin que es una herramienta sumamente eficiente para el Modelamiento de las Estructuras de Bases de Datos del AS/400 al SQL Server.

Todos los Aplicativos Cliente trabajarán con la data migrada del AS/400 al SQL Server, este proceso se hará en intervalos diarios, es decir la réplica se realizará al cierre del día.

DIAGRAMA DEL MODELO PROPUESTO



Posteriormente se irá incrementando la cantidad de Aplicativos en Visual Basic de acuerdo a las necesidades del negocio. Mientras se lleva a cabo el desarrollo de los aplicativos, el Area de Sistemas debe velar por la continuidad de los Sistemas actuales en RPG/400, brindando soporte y mantenimiento a todas las demás áreas de la compañía, así como también asesoría a los desarrolladores de los aplicativos visuales.

3.5 Características principales del Administrador de Base de Datos.-

3.5.1 SQL Server 6.5

Hemos optado para nuestra solución C/S por el manejador SQL Server tanto porque sus características principales se adecuan a los requerimientos de la compañía:

1. La magnitud de la data manejada por los aplicativos de la compañía están dentro de la capacidad que puede soportar el SQL Server adecuadamente.
2. La diferencia de precios respecto a otros manejadores de base de datos es sustancial.
3. SQL Server es muy versatil para comunicarse con los lenguajes de programación visual como el Visual Basic y Power Builder.
4. La reversión del efecto del procesamiento denominada **ROLL BACK** que permite recuperar las bases de datos sin afectarlas, en el caso de actualizaciones erroneas.
5. Permite hacer backups en tiempo real mientras la Base de Datos está siendo accesada.

6. La Replicación que se usa para distribuir automáticamente copias de sólo lectura de datos desde un servidor simple a uno o más servidores destino en uno o más lugares remotos.
7. Guardar Archivos Encriptados.
8. Disk Mirroring, es decir la data puede ser escrita automáticamente a una base de datos duplicada u otra partición sobre el mismo disco duro
9. Disk Dúplex, esta característica permite que la data sea escrita a una base de datos duplicada sobre otro disco duro.

Debemos acotar que en un futuro próximo pueda darse la necesidad de migrar a otros manejadores de bases de datos más robustos tales como Oracle, Sybase o Informix esto según el crecimiento de la información manejada.

Microsoft SQL Server incorpora un extensible Marco de trabajo de Administración distribuida orientada a objetos (SQL-DMF) que direcciona las necesidades del medio cliente/servidor en:

1. SQL Enterprise Manager, provee una consola administrativa de 32 bits que permite un control visual sobre las bases de datos distribuidas desde un punto central o site central.
2. SQL-Distributed Management Objects, maneja objetos OLE que le permiten adicionar nuevas capacidades administrativas con herramientas de programación OLE.
2. SQL Executive, permite automatizar rutinas de tareas y responder dinámicamente a condiciones no predecibles.

3.5.2 Conexión con SQL Server.-

La conexión con el SQL Server lo haremos mediante los objetos de Datos Remotos (RDO) que son un conjunto de objetos diseñados con las especificaciones del acceso remoto a datos. RDO permite establecer las conexiones, crea conjuntos de resultados y cursores y ejecuta procedimientos complejos, usando una mínima parte de los recursos de estación de trabajo.

Por ejemplo podemos usar el siguiente código para establecer la conexión con la base de datos Microsoft SQL Server con el nombre del origen de datos existente MyRemote:

```
Dim Cn As rdoConnection, -en As rdoEnvironment, Conn As String
Set En = rdoEnvironment(0)
Conn$ = "DSN= MyRemote; UID = Oriana; PWD = Chaski:Database_
= "MyDb;"
Set Cn = En.OpenConnection("", rdDriverPrompt, False, Conn$)
```

3.6 Características principales de la arquitectura de comunicaciones.-

3.6.1 SNA SERVER (System Network Architecture)

Para nuestra solución hemos optado por el producto Microsoft SNA Server que permite conectar con suma facilidad a las Pc's con las aplicaciones del AS/400. Este producto también permite la conectividad con la red LAN.

Las características técnicas de Microsoft SNA Server son:

Protocolos Cliente/Servidor	<ul style="list-style-type: none">• Named Pipes: protocolo de comunicación de redes de Microsoft• TCP/IP: protocolo LAN/WAN estándar• IPX/SPX: protocolo LAN propietario de Novell• Banyan VINES IP: protocolo LAN propietario de Banyan• RAS: protocolo LAN asíncrono de Microsoft
Capacidad del servidor	<ul style="list-style-type: none">• Mas de 2000 clientes por servidor• Mas de 10000 sesiones LU por servidor• Mas de 250 conexiones simultáneas por servidor, en cualquier combinación de hosts.

3.6.2 Requerimientos del SNA Server.-

Para la Plataforma AS/400 se requiere:

- Sistema operativo OS/400 versión 2 o posterior a SNA Server
- Windows NT Advanced Server 3.1 o Windows NT Server 3.5
- 16 MB RAM
- 20 MB disponibles de disco duro

3.7 Características principales del Groupware Server.-

Para el sistema de Mensajería se trabajará con Lotus Notes 4.0, este servidor nos permitirá realizar comunicaciones con los Concesionarios de provincias vía Internet y Correo Electrónico.

3.7.1 Lotus Notes.-

Entre las principales características de Lotus Notes tenemos:

- Lotus Notes es un servidor de Bases de datos de Documentos que deposita y administra clientes multiusuarios para acceder a texto, imágenes, audio y video. Esta versión 4.0 de Lotus Notes soporta a más de 1000 clientes.
- Es un servidor email que administra el acceso de los clientes multiusuarios al correo electrónico. Este servicio será utilizado por cada uno de los Concesionarios de provincias
- Es un medio cliente GUI que presenta vistas de las bases de datos documentos y provee un correo electrónico front-end. Los usuarios pueden navegar a través de las bases de datos y los documentos que contengan. Las vistas son “queries” o sentencias de consulta SQL depositados, que actúan como filtros para la información en las bases de datos.

Hemos considerado a Lotus Notes en nuestra solución Cliente/Servidor porque permitirá brindar los servicios apropiados de Internet, Correo Electrónico y aplicativos Notes integrados a Internet. Esta funcionalidad de Notes brindará la comunicación interactiva entre la compañía y los Concesionarios de provincias.

Los aplicativos Notes a desarrollarse estarán destinados básicamente a lo siguiente:

- Consulta de líneas de Crédito y
- Emisión de Cotizaciones Valorizadas.

a) Consulta de líneas de Crédito.- mediante este aplicativo los Concesionarios podrán verificar sus líneas de crédito disponibles antes de realizar cualquier compra.

b) Cotizaciones.- mediante este aplicativo Notes los Concesionarios desde cualquier punto del país, podrán hacer llegar sus cotizaciones valorizadas a la compañía vía Internet. Para realizar estas cotizaciones se tendrá acceso a niveles de stocks de los productos y también a los precios de día.

Los aplicativos Notes trabajarán directamente con la Base de Datos del servidor SQL, es decir la data que registre las cotizaciones y pedidos de compras, serán administradas por el SQL Server, este intercambio de datos se podrá llevar a cabo gracias al API de Lotus Notes que permite a los aplicativos Notes tener acceso al Servidor SQL.

También los API y los drivers ODBC permitirán realizar el caso contrario, es decir cuando se requiera que los aplicativos SQL tengan acceso a la data del Servidor Notes tal como se puede apreciar en la fig. 21.

Intercambio de datos entre Lotus Notes y SQL Server

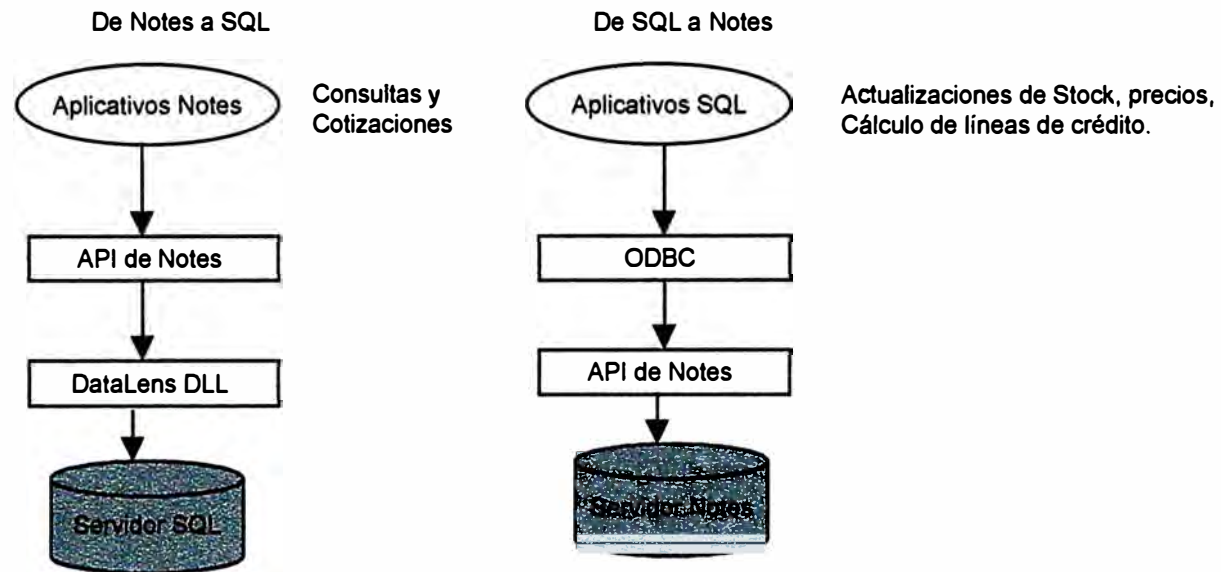


Fig. 21

3.8 Características principales del Sistema Operativo.-

El Sistema Operativo para nuestra solución será el Windows NT Server de Microsoft porque éste constituye una sólida base de red que brinda fácil acceso a la información que se necesita. Windows NT es un sistema operativo de fácil uso y administración, constituyéndose en una plataforma escalable, es decir no se tendrán que volver a escribir las aplicaciones cuando se actualice la plataforma de hardware a una más potente.

3.8.1 WINDOWS NT Server

Cuenta entre sus principales características que se ejecuta en diversas plataformas de hardware.

La seguridad de este sistema operativo está dada por:

- Bandas de disco
- Doble escritura en discos y duplicación de unidades
- Soporte de suministro de corriente eléctrica ininterrumpido (UPS)
- Diseñado para soportar copias de seguridad en cinta

3.8.2 Especificaciones del Windows NT Server.-

Componente	Requerimientos
CPU	Pentium.
RAM	16 MB como mínimo
Hard Disk	Un mínimo de 90 MB.
CD-ROM	Requerido sólo para la plana Gerencial o Márketing
Disquetera	

Opciones de Red

Soporta las siguientes redes:

- AppleTalk
- IBM LAN Server
- Redes IBM SNA
- Microsoft LAN Manager
- Windows for Workgroups
- Novell Netware
- Redes TCP/IP

Soporta los siguientes clientes

- Apple Macintosh
- MS-DOS
- OS/2
- UNIX
- Windows 95
- Windows for Workgroups
- Windows NT Workstation

3.9 Recursos Humanos

Para llevar a cabo el Desarrollo e implementación del Nuevo Sistema Integral se requerirá del siguiente personal:

- Un Jefe de Proyecto con conocimiento de AS/400, conectividad, comunicaciones, desarrollo de aplicativos visuales y redes.
- Un Integrador de Sistemas con dominio en SQL Server y Lotus Notes.
- Dos Analistas de Sistemas con dominio en Visual Basic 5.0.
- Tres Programadores en Visual Basic 5.0

El resto de Analistas y programadores será tomado del personal que actualmente labora en el Area de Sistemas, quienes se irán integrando al Proyecto a la par que se vayan capacitando en las herramientas de desarrollo. También es necesario considerar la participación de Usuarios a tiempo parcial para el levantamiento de información. Esto es se contará con un Usuario Integrador y Usuarios Supervisores, agrupándose cada uno por áreas.

En la fig. 22 puede verse la conformación del equipo de trabajo.

EQUIPO DE TRABAJO

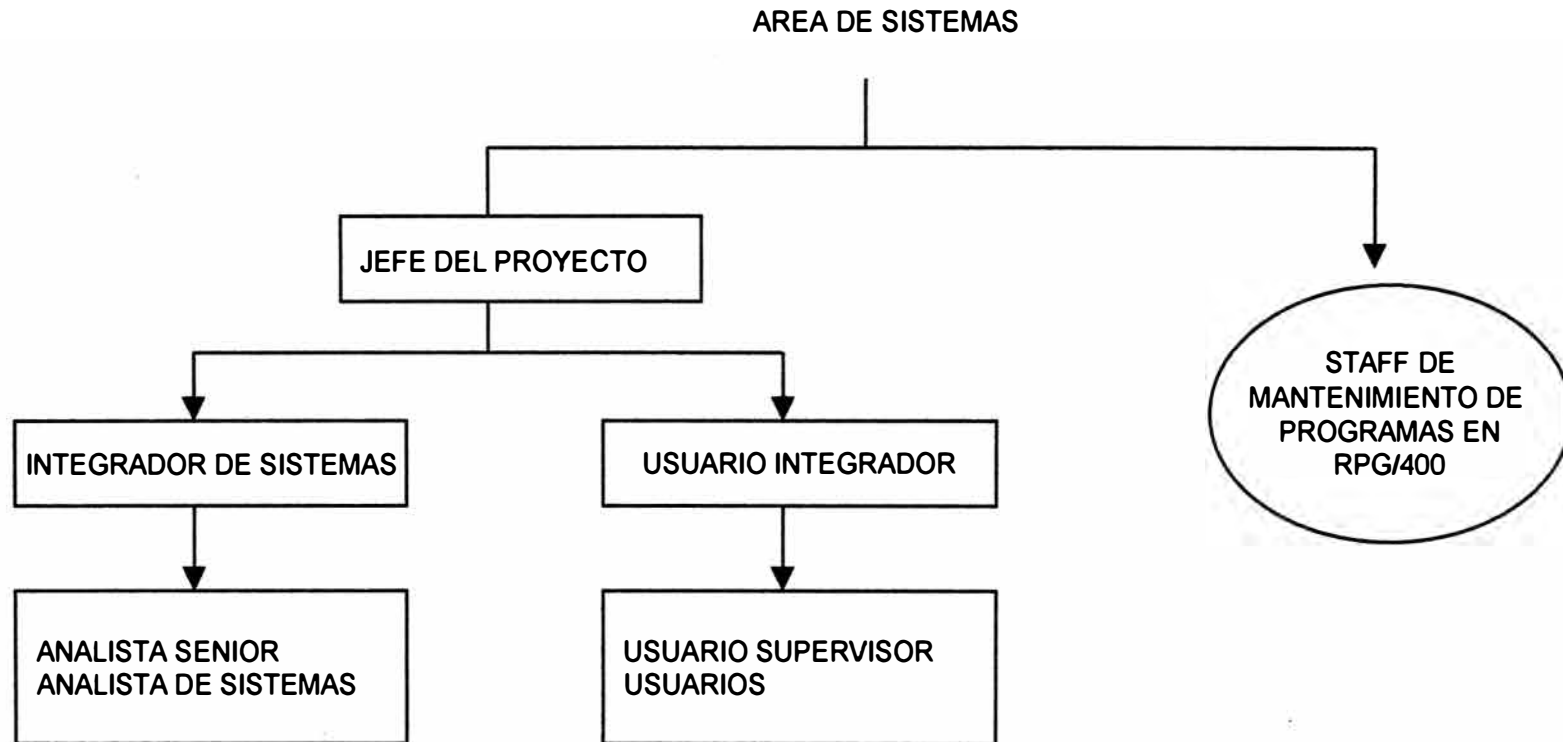


Fig. 22

3.10 Metodología para el desarrollo de los Aplicativos.-

Consideramos pertinente utilizar esta metodología que está compuesta de ocho fases y se basa inicialmente en un reconocimiento global de la compañía, en la selección de las herramientas de desarrollo, en la generación de prototipos hasta llegar a la fase de instalación en el Cliente de los aplicativos. A continuación describiremos detalladamente cada una de estas fases:

3.10.1. Fase Inicial.- En la fase inicial el Jefe del proyecto deberá hacer un reconocimiento global de la compañía y recabar toda la información necesaria proveniente de los usuarios.

El Jefe del Proyecto deberá tomar en cuenta los siguientes conceptos básicos de la Ingeniería de Información:

- a) Deberá considerar los objetivos y perspectivas de la Alta Gerencia de la compañía.
- b) Deberá conocer las necesidades, requerimientos y objetivos de la compañía.
- c) Deberá determinar las áreas de análisis y priorización.
- d) Deberá determinar los procesos de las áreas.
- e) Deberá definir la interrelación de los procesos.
- f) Deberá determinar que información se necesita.

3.10.2. Fase del Modelamiento.- En esta fase el Integrador de Sistemas debe diseñar junto con los analistas Senior el modelo de datos adecuado con el cual se basen el resto de las aplicaciones a desarrollar.

Para esta fase se deben seguir los siguientes procesos:

- a) Describir las bases de datos y tablas con que se trabajará, estas estarán conformadas por las Tablas propias de cada Aplicación y las Tablas Generales (también llamadas Paramétricas) que utilizan en común todos los aplicativos. Por ejm. Tabla de Monedas, Países y otras.
- b) Crear relaciones entre tablas.
- c) Determinar las relaciones entre entidades.
- d) Determinar que procedimientos residirán en el servidor, estos pueden ser: triggers, stored procedures.
- e) Determinar las reglas necesarias para conservar la integridad de los datos.

3.10.3. Fase de pruebas iniciales con las bases de datos.- Para realizar las pruebas iniciales con las bases de datos en el servidor, se deben seguir los siguientes pasos:

- Crear la Base de Datos en el Servidor SQL Server.
- Cargar datos de prueba en las tablas.
- Hacer las pruebas del lado del cliente pudiéndose reconstruir la base de datos más de una vez.
- Determinar el proceso de migración de los datos y hacer las pruebas pertinentes de Sincronización con el Erwin para la data del AS/400.

3.10.4. Fase de evaluación y diseño inicial en el cliente.- En el Cliente el sistema operativo standar será el Windows 95, de tal manera que las aplicaciones puedan desarrollarse con Visual Basic sin ningún inconveniente puesto que nos encontraríamos en la misma plataforma operativa de Microsoft.

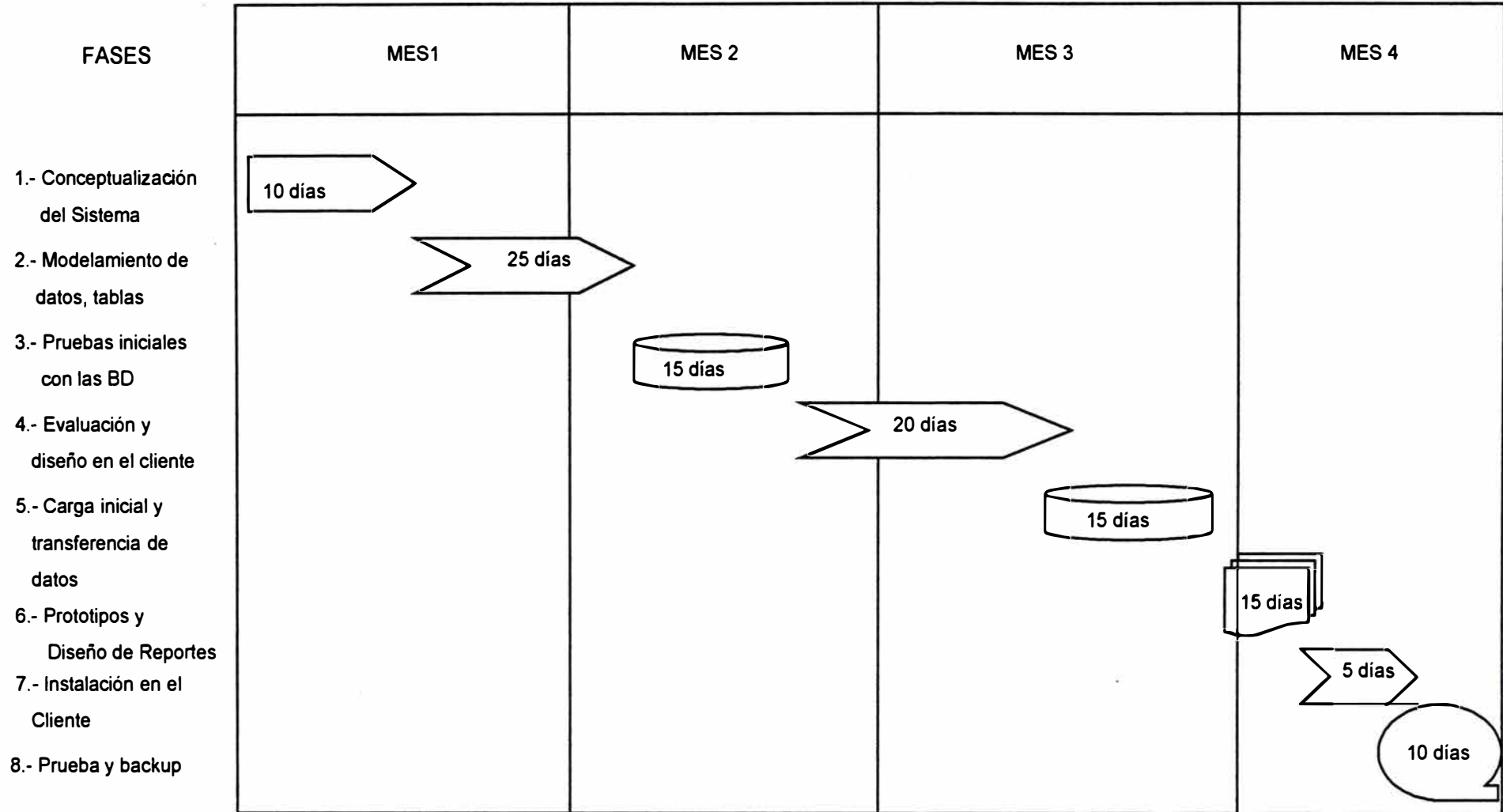
3.10.5. Fase de carga inicial y transferencia de los datos.- En esta fase migraremos la totalidad de la data pertinente a los aplicativos del Sistema Integral de Ventas, esta migración se realizará utilizando el Erwin desde el AS/400 al SQL Server.

La réplica de los datos será en forma vertical es decir del AS/400 a los servidores y se realizará al cierre del día. La réplica de los datos se realizará con la finalidad de mantener la data actualizada de una plataforma a otra.

3.10.6. Diseño de reportes.- En esta fase el usuario detallará los formatos de los reportes, estos reportes se generarán en el servidor y no en el cliente ya que esto minimizará los tiempos de respuesta. Es aconsejable revisar el diseño de los reportes con el usuario final tal que ellos sean partícipes en el desarrollo de los mismos.

3.10.7 Instalación en el Cliente y niveles de seguridad.- Una vez terminado de desarrollarse el Sistema Integral de Ventas, este debe ser instalado en cada Cliente, tal que el usuario final pueda empezar a probarla y determinar que cambios son necesarios.

**DIAGRAMA DE GANTT DE LA METODOLOGIA
CLIENTE / SERVIDOR**



Será necesario también ponerse las rutinas de control de seguridad en el servidor, limitando así los permisos a las diferentes estructuras de las bases de datos y a grupos de usuarios. Para esto deberá diseñarse un **Sistema de Perfiles de Usuarios**.

3.10.8. Etapa de prueba y Backup.- Deberán probarse tanto la aplicación, las reglas de seguridad, los stored procedures, triggers y analizar los resultados finales que entrega el sistema cliente/servidor haciendo un paralelo con los resultados obtenidos con los sistemas en RPG/400.

También deben definirse las rutas de respaldo para evitar pérdidas de información, como por ejemplo determinar los intervalos de tiempo en que se sacarán Backup ó respaldo de las bases de datos.

3.11 Consideraciones administrativas del modelo C/S planteado.-

Es aconsejable tener en cuenta las siguientes consideraciones:

- Dividir las operaciones de computación entre las que deben ejecutarse centralizadamente en el servidor y las que puedan ejecutarse en forma individual en el cliente.
- En esta división de recursos de computación, las aplicaciones que se ejecuten en el servidor son aquellas que graban, procesan, protegen y distribuyen los datos. Por otro lado, las aplicaciones que se ejecuten en el cliente son aquellas que solicitan los datos al servidor y procesan la interfaz que presentará dicha información en pantalla.

- Las consultas deben utilizar las técnicas más apropiadas para hacer que el servidor sólo envíe la información solicitada por el cliente, eliminando así la causa principal de la ineficiencia en los sistemas basados en redes: el gran tráfico de datos que existe en el medio de comunicación.
- Además de reducir el tráfico en la red, una verdadera arquitectura C/S debe mejorar el rendimiento del sistema, asignando las tareas de procesamiento de gran volumen de datos a la máquina más potente de la arquitectura, que normalmente es el servidor de la base de datos.
- Se deberá conocer exactamente el contenido de las Bases de Datos y de los usuarios a quienes se les debe otorgar acceso limitado a actualizaciones de datos.
- Por consiguiente, todas las reglas que regulan el acceso a las bases de datos y al procesamiento de las transacciones, deberán ser establecidas en el servidor junto a las mismas bases de datos, impidiendo de esa forma que los usuarios traspasen la seguridad del sistema.
- Sin la inteligencia del servidor se vería afectado uno de los principales beneficios de la arquitectura Cliente/Servidor que consiste en permitir que un alto número de usuarios tengan acceso a la información de misión-crítica.

3.12 Aspecto organizativo y consideraciones de nuestro aplicativo.-

***Usuarios** : Gerentes, Usuario Integrador y Usuarios finales.

***Areas** : Ventas, Tesorería y Logística.

***Sistemas**

- Gerenciales
- Front-end
- Back-end
- Staff: comunicaciones, Bases de Datos.

Las consideraciones para el desarrollo de nuestro Aplicativo Cliente/Servidor en Visual Basic serán las siguientes:

- Diseño del GUI (Interfaz Gráfica del Usuario)
- Diseño de procesos
- Diseño de Bases de Datos distribuidas
- Diseño de transacciones

3.12.1. Diseño del GUI.-

- Definir estándares de interfase gráfica con el usuario
- Identificar y analizar usuarios y tareas
- Definir interfaz inicial
- Desarrollar prototipo e iterar hasta terminar

3.12.2 Diseño de Procesos.-

- Identificación de procesos reusables
- Identificación de stored procedures
- Consideraciones de concurrencia y seguridad
- Distribución de procesos

3.12.3 Diseño de Bases de Datos Distribuidas .-

- Definición de BD relacionales normalizadas
- Identificar candidatos de denormalización
 - Denormalizar
 - Definir vistas e índices

3.12.4 Diseño de Transacciones.-

- Identificación de transacciones de larga vida
- Estimar tráfico diario en la red y espacio en disco al año
- Definir buenas transacciones
- Resolver problemas de concurrencia y seguridad.

3.13 Consideraciones básicas de Seguridad del Modelo.-

- a) Dar respaldo y restauración a los datos.
- b) Velar por la recuperación de desastres y establecer estrategias de contingencias.
- c) Administrar dispositivos de almacenamiento de acceso directo, archivo y administración de espacios.
- d) Rendimiento y afinación.
- e) Determinar problemas que se susciten en la base de datos.
- f) Diseñar las bases de datos necesarias.

3.14 Herramienta de Desarrollo.-

Para el desarrollo de los aplicativos en nuestra solución Cliente/Servidor hemos optado por el lenguaje Visual Basic 5.0 por las siguientes razones:

- Como hemos podido verificar por experiencia propia, Visual Basic está específicamente diseñado para proporcionar al programador un conjunto completo de herramientas y estrategias de diseño con las que se pueden crear soluciones Cliente/Servidor robustos y eficientes para los problemas empresariales.
- Cabe resaltar que Power Builder es también un excelente desarrollador de aplicaciones Cliente/Servidor y en el Perú grandes corporaciones lo utilizan para su desarrollo tales como el Banco de Crédito, SUNAD, etc. Sin embargo hemos optado por Visual Basic ya que conocemos a profundidad su manejo y lo consideramos una herramienta excelente para el desarrollo de aplicativos. También consideramos que por pertenecer a la empresa Microsoft, esto nos asegura una continuidad, estandarización y un soporte técnico adecuado para el manejo de los aplicativos dentro del entorno Windows.
- Visual Basic tiene incorporado el manejo en forma directa de varios tipos de archivos, entre los que se encuentran Access, Foxpro, Dbase, Paradox y Btrieve. También incorpora la posibilidad de utilizar drivers ODBC para la conexión con otros tipos de archivos . Por defecto Visual Basic incorpora también dos drivers para manejo de archivos tipo Oracle o SQL Server.

- Visual Basic es un lenguaje que hace fácil el crear aplicaciones complejas porque posee herramientas de diseño gráfico y de debug, a la vez que su última versión, la versión 5.0 permite una gran conectividad con Internet

En nuestro Sistema de Información Integral se manejará :

a) A nivel del Cliente

Pantallas

Gráficos Estadísticos

– Menús

(Algunos pantallas y menús del aplicativo en Visual Basic pueden verse en el Anexo C)

b) A nivel del Servidor

Reportes

Stored Procedures (Procedimientos y Funciones)

– Packages

Triggers

Mientras se lleva a cabo el desarrollo de los aplicativos en Visual Basic por parte del equipo destinado, el Area de Sistemas de la empresa deberá velar por la continuidad de los Sistemas actuales en RPG/400, brindando soporte y mantenimiento a todas las demás áreas de la empresa, así como también asesoría a los desarrolladores de los aplicativos.

Las metodologías para el desarrollo de aplicativos son muy variadas y se establecen según los procesos que se involucran en los sistemas, tales como Procesos de Planilla, Cálculos Estadísticos, Consultas o solamente Registro de información. Algunas metodologías establecen que toda la lógica del negocio vaya en el Servidor, otros plantean un desarrollo a nivel Cliente y muy poco en Servidor y otras metodologías son una mixtura de las dos anteriores.

El desarrollo del Sistema Integral de Ventas en Visual Basic, utilizaremos la metodología de los tres niveles o 3-tier.

3.14.1 Metodología de Tres Niveles ó 3-tier.-

Nuestra Solución Cliente/Servidor plantea la utilización de una metodología que facilite el desarrollo de los aplicativos del negocio dividiendo todo el desarrollo en tres niveles:

Niveles de usuario

Niveles de la organización

Niveles de datos

Nivel de usuario.-

Este nivel involucra la lógica de presentación es decir los aplicativos cliente o front-end. Estos aplicativos serán las presentaciones de pantallas con interfaz gráfica para el usuario y residirán en un programa ejecutable ubicado en la estación cliente del usuario final.

Nuestros aplicativos Cliente utilizarán herramientas OLE para las interfaces que se requieran con paquetes como Word, Excel u otros.

Para desarrollar este nivel de usuario se utilizará el lenguaje Visual Basic.

Nivel de la organización.-

Este nivel involucra la lógica de aplicación o de negocio.

Estos servicios responden a las peticiones del usuario y son un puente entre el Nivel de Usuario y el Nivel de datos. Una tarea de la organización es una operación definida por los requerimientos de la aplicación, como por ejemplo Ingresar un Pedido de compra o Imprimir una Lista de Clientes. Las reglas de la organización son normas que controlan el flujo de las tareas de la organización, por ejemplo un procedimiento para la Subida de Precios cada cierto Periodo y bajo ciertas condiciones, para hallar el límite de crédito, etc. Como las reglas de la organización tienden a cambiar con mas frecuencia a las reglas de la organización a las que se aplican podrían encapsularse en componentes independientes de la propia lógica de la aplicación.

En Visual Basic estos constituyen los programas conocidos como DLLs.

Nivel de datos.-

En este nivel se definen, mantienen, actualizan y acceden a los datos además de administrar y responder a las peticiones de datos del Nivel de la organización. Este nivel involucra a la lógica de datos es decir a objetos tales como Stored Procedures, Triggers o Views que se desarrollarán según el motor de datos SQL Server.

Entre la ventajas de esta metodología tenemos:

- El máximo nivel de flexibilidad en términos de distribución y redistribución de funciones y datos.
- Las aplicaciones resultantes con esta metodología son escalables, flexibles y abiertas.

- Pueden seleccionarse las herramientas y lenguajes más adecuados para cada una de las partes de la aplicación.

3.15 Estudio Costo/Beneficio.-

3.15.1 Costos

A continuación detallaremos los costos que involucrará nuestra solución Cliente/Servidor, estos costos han sido divididos en: Costo de Personal, Costo de Operación, Costo de Desarrollo y Costo de Servicios.

Costo de Personal :

2 Analista Senior por 6 meses	\$ 38,400
3 Analistas-Programadores por 6 meses	\$ 36,000
	<hr/>
Total	\$ 74,400

Costo de Operación:

4 Servidores NT	\$ 12,500
3 Hubs	\$ 4,500
Upgrade de 24 PC's a Pentiums	\$ 6,000
16 Pentiums	16,000
	<hr/>
Total	\$ 39,000

Costo de Desarrollo :

SQL Server 6.5 para 50 usuarios	13,000
Lotus Notes con 230 licencias Desktop	<u>45,000</u>
Total	\$ 58,000

Costo de Servicios :

Cursos de Capacitación	5,150
Instalación de Red LAN	<u>5,000</u>
Total	\$ 10,150

TOTAL DE COSTOS \$ 186,550

3.15.2. Beneficios.-

Los beneficios que traerá para la compañía nuestra solución Cliente/Servidor pueden clasificarse en Beneficios tangibles e intangibles.

Beneficios Intangibles:

- La Plana Gerencial tendrá un acceso más ágil dentro de un entorno Windows y con herramientas visuales, a la consulta de datos e interfaces gráficas, lo cual redundará en una más eficiente toma de decisiones empresariales.
- Los vendedores y concesionarios se sentirán más identificados con la compañía y a la vez mejor comunicados.
- Un mejor ajuste del sistema de información a la organización y a los procesos de negocio.

- Localización de funciones y datos donde sean necesarios para la operativa diaria sin cambiar las aplicaciones.
- Acceso a la información cuando y donde la necesiten los usuarios.

Beneficios Tangibles

- Ahorro en los costos de viajes y viáticos que acarrea el viaje semanal de cada uno de los 12 vendedores a las diferentes provincias del Perú, estos viajes se realizan para registrar las cotizaciones y pedidos de los concesionarios.
- Ahorro en los costos de mantenimiento de los programas, estos costos serán inferiores al de los sistemas desarrollados actualmente en el sistema AS/400.
- Se espera un incremento en el volumen de ventas de 20% en los primeros 5 meses.

Calculamos que el Valor Neto de las Ventas esperado en los primeros 5 meses es de \$ **120,000 U.S.** Por lo tanto comparando el beneficio esperado menos los costos generados por la implantación de la Solución Cliente/Servidor, la inversión se recuperaría aproximadamente en 10 meses.

3.16 Características modulares del Nuevo Sistema de Información Integral.-

De acuerdo al levantamiento de Información que realizamos para conocer las diferentes funciones de la empresa hemos logrado determinar algunas características principales que tendrá el nuevo sistema a desarrollarse:

VENTAS

CARACTERÍSTICAS FUNCIONALES

- 1.- Ingreso de la orden de pedidos.
- 2.- Historia del cliente en línea.
- 3.- Consulta por búsqueda alfabética por nombre y R.U.C del cliente.
- 4.- Comprobación en línea del crédito del cliente.
- 5.- Consulta de Stock para Muelles, Bujías y Repuestos.
- 6.- Cotizaciones y conversiones de pedidos.
- 7.- Ingreso, corrección y mantenimiento en línea de los pedidos.
- 8.- Precios y tipos de descuento o bonificaciones.
- 9.- Consulta completa del pedido por: RUC y nombre del cliente, código, artículo, vendedor, fecha, cantidad.
- 10.- Ingreso de pedidos en distintas monedas y traducidos en moneda nacional usando un tipo de cambio con posibilidad de cambiarse durante el armado de la orden.
- 11.- Consulta de disponibilidad para entrega en el momento de la entrada del pedido.
- 12.- Condiciones de variables de pago.
- 13.- Consulta histórico de envíos de mercadería a clientes.
- 14.- Histórico de Ventas por Vendedor.

- 15.- Récord de Ventas por Vendedor, por líneas y tipo de moneda.
- 16.- Histórico de Compras por Cliente, por año, mes y líneas de productos.
- 17.- Consulta de Comisiones por Vendedor.
- 18.- Hojas de Ruta de ayuda y control al vendedor con programa de visitas.

FACTURACIÓN

CARACTERÍSTICAS FUNCIONALES

- 1.- Facturación en línea y totalmente integrada para todos los tipos de items (muelles, bujías, repuestos).
- 2.- Emisión de Facturas, Boletas de Venta, Letras, Notas de Cargo y Abono.
- 3.- Opciones para eliminar, visualizar e imprimir documentos.

CUENTA CORRIENTE

CARACTERÍSTICAS FUNCIONALES

1. Registro y Control de Clientes.
2. Cálculo de Crédito disponible del Cliente por línea y tipo de moneda.
3. Gestión de pedidos en línea y en tiempo real que excedan límites de crédito autorización o bloqueo por la Gerencia Administrativa financiera y por Tesorería
4. Comprobación del crédito del cliente con revisión y liberación en línea de los pedidos retenidos sobre el crédito disponible.

5. Consulta del histórico del comportamiento crediticio por cliente y emisión de reportes opcionalmente.
6. Emisión de Calendario de cancelaciones y/o fecha de vencimiento de letras.
7. Consulta y reporte de cuentas pendientes de pago por cliente
8. Registro y control de amortizaciones y renovación de letras.
9. Generación de letras, cálculo de intereses y control de fechas de vencimiento.
10. Control de las facturas pendientes de cobranza.
11. Registro de los diferentes tipos de operación como: cancelación total, cancelación parcial, amortización , renovación de letras .
12. Consulta y emisión de reportes de comisiones pagadas a vendedores.
13. Consulta de liquidaciones por fechas, tipos de venta, servicio, montos.
14. Emisión de avisos a morosos.
15. Interfaces y actualizaciones completas y automáticas de cuentas por cobrar, control de inventarios, ventas e interfase con contabilidad.

DESPACHO

CARACTERÍSTICAS FUNCIONALES

- 1.- Órdenes de Despacho en línea luego de confirmado el pedido.
- 2.- Registro y control de Agencias
- 3.- Confirmación de Despacho o entregas
- 4.- Emisión de Guías

CONSULTAS GERENCIALES

CARACTERÍSTICAS FUNCIONALES

- 1.- Mantenimiento en línea sobre proyecciones de ventas considerando índice inflacionario, competencia, niveles de producción.
- 2.- Estadísticas de Ventas por Vendedor
- 3.- Estadísticas de Ventas por Cliente
- 4.- Estadísticas de Ventas por Zona Geográfica
- 5.- Ranking de Ventas por Clientes
- 6.- Ranking de Ventas por Vendedor
- 7.- Estadísticas de Ventas diarias, mensuales y anuales desagregadas por Volúmenes de Venta y Montos Totales,

19. Principales Entidades identificadas.-

1. CLIENTE
2. VENDEDOR
3. CUENTA CORRIENTE
4. FACTURA
5. BOLETA DE VENTA
6. GUÍA
7. LETRAS
8. NOTA DE CARGO
9. NOTA DE ABONO
10. ARTÍCULO
11. PEDIDO
12. VENTAS

CONCLUSIONES

1. El modelo Cliente/Servidor no sólo beneficiará a la compañía en cuanto al manejo eficaz y oportuno de su información, sino que también le permitirá lanzar al mercado nacional e internacional, una imagen competitiva y de avanzada “ad portas” el Nuevo Milenio, cumpliéndose así los objetivos iniciales del nuevo concepto de Márketing adoptado por la compañía.

2. Al estar los Concesionarios de provincias conectados con la central de la compañía vía Lotus Notes, podrán tener acceso a las consultas de datos y realizar sus pedidos de una manera más rápida y efectiva a como se venía haciendo hasta ahora, esto redundará en un futuro cercano en beneficios económicos tanto para la compañía como para los propios concesionarios.

3. Debe considerarse que la utilización de SQL Server como Administrador de la Base de Datos puede ser temporal según se vaya incrementando la data y los requerimientos de seguridad sobre la misma, en un futuro cercano puede optarse por ejemplo por trabajar con Oracle, que a nuestro criterio es uno de los manejadores de datos más robustos que existen en la actualidad.

4. Nuestra solución Cliente/Servidor garantiza la seguridad de los datos y a la vez permite que los usuarios estén en condiciones de acceder en forma

directa a la información necesaria y en el momento oportuno para desempeñar sus funciones.

5. Con la Arquitectura Cliente/Servidor, las aplicaciones de la compañía podrán funcionar en una amplia variedad de plataformas de sistemas. Y sobretodo su presentación ante el usuario será más amigable.
6. Gracias a la solución Cliente/Servidor se verá aumentada la capacidad administrativa de la compañía, con lo cual se incrementará la automatización de normas o reglas de contabilidad, ventas, pagos, etc.
7. Las aplicaciones basadas en el modelo Cliente/Servidor permitirán aprovechar los avances en el diseño de la interfaz y las ayudas en línea al usuario para facilitar su aprendizaje y uso.
8. Si en un futuro cercano la compañía tuviera que redimensionarse, pasando a utilizar un servidor mas potente u otros Administradores de bases de datos, va a poder seguir utilizando las mismas aplicaciones en Visual Basic 5 que se desarrollen durante este proyecto.
9. Consideramos que poco a poco todos los aplicativos de la compañía que están bajo AS/400 serán desarrollados en Visual Basic 5.0, así como también la data será migrada en su totalidad al SQL Server 6.5, estimamos que puede hacerse todo esto en el lapso de un año y medio.

10. El Sistema de Información Integral que se proyecta desarrollar permitirá una mejor distribución en la carga de trabajo de cada Cliente y a la vez una mejora notable en la carga de trabajo sobre la red.

RECOMENDACIONES

- a) Consideramos que en la actualidad todos necesitamos tomar ventaja de la arquitectura Cliente/Servidor y en general de todas las tecnologías de punta que se abren a nuestros pasos para de esta manera extender los límites de los sistemas computarizados en beneficio de las compañías donde laboremos.

- b) Actualmente el sistema AS/400 tiene mucho éxito en el ambiente Cliente/Servidor porque cuenta con una sofisticada tecnología para el ambiente comercial y su facilidad de uso, pues este requiere de poco entrenamiento. Como hemos podido investigar a lo largo de nuestra tesis, el AS/400 permite que existan aplicaciones basadas en archivos planos y coexistan con aplicaciones de bases de datos relacionales, y aun más, las aplicaciones que requieren procesamiento interactivo y batch son actualmente más fáciles de programar en AS/400 con el lenguaje RPGLE.

- c) Creemos que toda inversión que se haga en la capacitación al personal, creará una Cultura Informática Competitiva y explotará mejor la implementación de esta nueva tecnología Cliente / Servidor.

- d) De acuerdo a estudios realizados en Estados Unidos, los ingresos registrados en 1995 por las ventas de bases de datos SQL fue de \$5.9 billones de dólares, creciendo en un rango de 30% anualmente. Como detallamos en páginas anteriores sólo un 10% de la data en el mundo está depositada en bases de datos

relacionales, tal que la oportunidad de crecimiento de las bases de datos relacionales se incrementará en los próximos años.

- e) Las principales compañías de investigación de mercados en Estados Unidos predicen que para el año 2000, los Cinco Grandes vendedores de bases de datos serán: IBM, Informix, Microsoft, Oracle y Sybase.
- f) Hoy en día SQL SERVER está considerado como un manejador de base de datos potente y con un alto grado de interoperabilidad con otras plataformas.
- g) Consideramos importante señalar que las aplicaciones Cliente/ Servidor se pueden diseñar con características que no son Orientadas a Objetos; de la misma manera, una aplicación Orientada a Objetos puede no tener ningún atributo Cliente/Servidor. Sin embargo, es importante que resaltemos que el Análisis y Programación Orientados a Objetos son tecnologías de punta que no se pueden ignorar; una infraestructura Cliente/Servidor y el entorno de desarrollo de Aplicaciones deben ser capaces de incorporar a las tecnologías Orientadas a Objetos en un futuro próximo, aunque no se utilicen inicialmente.

BIBLIOGRAFIA

1. Orfali Robert, Harkey Dan, Edwards Jeri ; “The Essential Client/Server Survival Guide”, Segunda Edición, USA, 1997.
2. Centro Europeo de Soluciones Cliente/Servidor de IBM ; “Fundamentos Cliente / Servidor”, Segunda edición, España, 1995
3. Pantajja Jim; “The Microsoft SQL Server survival guide”, 1º Edición, USA, 1996.
4. Microsoft; “Manual para un Desarrollo Cliente / Servidor con Visual Basic”
5. Salemi Joe; “Client/Server computing with Sybase SQL Server”, 2º Edición, USA, 1994.
6. Gates Bill; “The Road Ahead”, USA, 2º Edición, USA, 1996
7. Alope Nath; “The guide to SQL Server”, 1º Edición, USA,1995.
8. Microsoft Education and Certification Delivery Guide (Course Number : 756); “Systems Administration for Microsoft SQL 6.5”
9. Kast Fremont E., Rosenzweig James E.; “Administración en las Organizaciones, Cuarta Edición, USA, 1988.

GLOSARIO

Protocolo, es un conjunto de estándares que permite que computadoras o dispositivos puedan intercambiar información

API (Interfaz de Programa de Aplicación) , es un lenguaje y formato de mensaje utilizados por un programa para activar e interactuar con las funciones de otro programa o de hardware. Por ejemplo el API de conectividad a las Bases de Datos de Microsoft es el ODBC.

File Server, es un computador, generalmente una PC, que provee los recursos compartidos primeramente sobre una LAN y que pueden ser usados por todas las Workstations o nodos sobre la LAN. Los recursos compartidos generalmente incluyen espacio en Disco e Impresoras.

Protocolos de Comunicación LAN Común

Los protocolos usados para la comunicación entre hosts puede tomar muchas formas. Uno de los mas comunes es el llamado Protocolo Internet (IP). Otros dos ejemplos de protocolos de redes son IPX y DECnet.

Red física.- es una colección de enlaces de comunicación, cables, routers y conexiones que trabajan juntos. Las redes de hoy consisten de una combinación de LANs, incluyendo Token Rings y Ethernets; WANs incluyendo bridges, routers y gateways que proveen la interconectividad.

Bridges .- son computadores o dispositivos que interconectan las LANs usando capas de enlace que llevan la información y las direcciones físicas.

Routers .-interconectan LANs usando protocolos dependientes que llevan la información. Son usados con protocolos que soportan una capa de red como: TCP/IP, IPX/SPX, Apple Talk.

Gateways .-son dispositivos que ejecutan traducciones entre protocolos. Son usados en situaciones en que la estructura de la red soporta sólo un protocolo y todos los otros protocolos son trasladados a éste.

Local Area Network (LAN) , es la red que provee una conectividad de alta velocidad para una pequeña área geográfica, menos de 3 Km. de distancia, por ejemplo en un edificio, permite acceder servidores de archivos, de impresoras, hosts, etc.

Principales Arquitecturas LAN, las principales Arquitecturas LAN son:

1. **Ethernet.**- protocolo estándar de 10Mbps, aparece en 1981.
2. **Token Ring.**- fue desarrollada por los laboratorios de IBM en Zurich, Suiza. Los primeros productos Token Ring aparecieron en 1986. Es ahora tan popular como Ethernet.

Metropolitan Area Network (MAN), es la red que provee conectividad de alta velocidad, más de 100 Mbps para un área geográfica de una magnitud, menor a 50 Km. y permite a los usuarios los mismos recursos que una LAN.

Wide Area Network (WAN) , es la red que abarca más grandes distancias que una MAN, tales como ciudades, regiones, o tal vez países, no hay límite de distancias, sin embargo la velocidad de la WAN es menor que de la MAN.

SNA, es la arquitectura de comunicaciones de IBM a través de la cual se encuentran conectados millones de computadores personales a sistemas medios y grandes.

Artículo, Es la unidad básica de la replicación. Se refiere a una tabla dentro de una publicación.

Publicación, Es un grupo de tablas marcadas para la replicación. Una publicación puede contener uno o más artículos. Se pueden publicar:

- Grupo de Tablas
- Tablas enteras
- Tablas particionadas verticalmente
- Tablas particionadas horizontalmente

Lo que no pueden ser publicados son:

- Las tablas de las bases de datos *model*, *tempdb*, *msdb* y las tablas del sistema en la base de datos *master*.
- Tablas sin primary key, excepto en replicación snapshot.

HTML (Hypertext Markup Language).- Es un lenguaje basado en un texto simple que permite enlazar la información en muchas sites de Internet.

Proveedor de servicios.- Es una compañía o institución que proporciona conexión directa o indirecta a Internet. Por ejemplo: Telefónica del Perú, Red Científica Peruana, Ameritech, etc.

ANEXO A

Stored Procedures

EJEMPLO 1.- Este Procedimiento del Servidor, realiza el cálculo de AFP para la Tabla de planilla_afp segun el monto ganado por el personal.

```
-- *****  
PROCEDURE sp_reporte_afp  
( aSPID      number,  
  aPERIODO   varchar2,  
  aPLANILLA  varchar2,  
  aAFP       varchar2  
)  
IS  
cPERSONAL    varchar2(10);  
cPERSONAL_PREVIO varchar2(10);  
nORDEN_AFP   integer;  
rMONTO       real;  
rCONCEPTO01 real;  
rCONCEPTO02 real;  
rCONCEPTO03 real;  
rCONCEPTO04 real;  
rCONCEPTO05 real;  
rCONCEPTO06 real;  
CURSOR CURSOR_PLANILLA IS  
  select personal, orden_afp , monto  
  from planilla_afp  
  where spid = aSPID  
  order by personal;  
BEGIN  
-----  
  delete from operador;  
  insert into operador (spid,planilla,periodo) values (aSPID, aPLANILLA,  
aPERIODO);  
-----  
rCONCEPTO01 := 0;  
rCONCEPTO02 := 0;  
rCONCEPTO03 := 0;  
rCONCEPTO04 := 0;  
rCONCEPTO05 := 0;  
rCONCEPTO06 := 0;  
open CURSOR_PLANILLA;
```

```

fetch CURSOR_PLANILLA into cPERSONAL, nORDEN_AFP , rMONTO;
if not(CURSOR_PLANILLA%FOUND) then
  close CURSOR_PLANILLA;
  return;
end if;
cPERSONAL_PREVIO := cPERSONAL;
DELETE FROM AFP_REPORTE WHERE spid = aSPID;
LOOP
  if nORDEN_AFP = 1 then
    rCONCEPTO01 := rCONCEPTO01 + rMONTO;
  end if;
  if nORDEN_AFP = 2 then
    rCONCEPTO02 := rCONCEPTO02 + rMONTO;
  end if;
  if nORDEN_AFP = 3 then
    rCONCEPTO03 := rCONCEPTO03 + rMONTO;
  end if;
  if nORDEN_AFP = 4 then
    rCONCEPTO04 := rCONCEPTO04 + rMONTO;
  end if;
  if nORDEN_AFP = 5 then
    rCONCEPTO05 := rCONCEPTO05 + rMONTO;
  end if;
fetch CURSOR_PLANILLA into cPERSONAL, nORDEN_AFP , rMONTO;
  if rtrim(cPERSONAL) <> rtrim(cPERSONAL_PREVIO) or
not(CURSOR_PLANILLA%FOUND) THEN
    rCONCEPTO06 := rCONCEPTO01 + rCONCEPTO02 + rCONCEPTO03 +
rCONCEPTO04 + rCONCEPTO05;
INSERT INTO AFP_REPORTE ( SPID , PERSONAL , CONCEPTO01,
CONCEPTO02,
    CONCEPTO03, CONCEPTO04, CONCEPTO05, CONCEPTO06)
    VALUES (aSPID ,cPERSONAL_PREVIO,
rCONCEPTO01,rCONCEPTO02,
    rCONCEPTO03, rCONCEPTO04, rCONCEPTO05, rCONCEPTO06);
rCONCEPTO01 := 0;
rCONCEPTO02 := 0;
rCONCEPTO03 := 0;
rCONCEPTO04 := 0;
rCONCEPTO05 := 0;
rCONCEPTO06 := 0;
if nORDEN_AFP = 1 then
  rCONCEPTO01 := rMONTO;
end if;
if nORDEN_AFP = 2 then
  rCONCEPTO02 := rMONTO;
end if;

```

```
if nORDEN_AFP = 3 then
  rCONCEPTO03 := rMONTO;
end if;
if nORDEN_AFP = 4 then
  rCONCEPTO04 := rMONTO;
end if;
if nORDEN_AFP = 5 then
  rCONCEPTO05 := rMONTO;
end if;
end if;
cPERSONAL_PREVIO := cPERSONAL;
if not(CURSOR_PLANILLA%FOUND) then
  EXIT;
end if;
END LOOP;
close CURSOR_PLANILLA;
COMMIT;
END;
```

EJEMPLO 2.- Este Procedimiento del Servidor, realiza el cálculo de las remuneraciones según los datos de la tabla **personal_select** para actualizar la tabla **asistencia_periodo**.

```
-- *****
PROCEDURE SP_PR02_xASISTENCIA_BASE
(aSPID number,
 aPERIODO varchar2
)
IS
F_FIN_PERIODO date;
F_INI_PERIODO date;
F_INI_ANNO date;
F_FIN_ANNO date;
F_INI_ESCOLAR date;
F_FIN_ESCOLAR date;
F_INI_GRATIFICA date;
F_FIN_GRATIFICA date;
ANNO varchar(04);
ANNOE varchar(04);
MES number(2,0);
BEGIN
-----
MES := to_number(SUBSTR(aPERIODO,5,2));
ANNO := SUBSTR(aPERIODO,1,4);
-----
F_INI_ANNO := TO_DATE('01-01-' || ANNO, 'DD-MM-YYYY');
F_FIN_ANNO := TO_DATE('31-12-' || ANNO, 'DD-MM-YYYY');
-----
F_INI_PERIODO := fp_ut02_periodo_to_fecha_ini(aPERIODO);
F_FIN_PERIODO := fp_ut03_periodo_to_fecha_fin(aPERIODO);
-----
if MES < '07' then
    F_INI_GRATIFICA := TO_DATE('01-01-' || ANNO, 'DD-MM-YYYY');
    F_FIN_GRATIFICA := TO_DATE('30-06-' || ANNO, 'DD-MM-YYYY');
else
    F_INI_GRATIFICA := TO_DATE('01-07-' || ANNO, 'DD-MM-YYYY');
    F_FIN_GRATIFICA := TO_DATE('31-12-' || ANNO, 'DD-MM-YYYY');
end if;
-----
ANNOE := to_char(to_number(ANNO) -1);
F_INI_ESCOLAR := TO_DATE('02-12-' || ANNOE, 'DD-MM-YYYY');
F_FIN_ESCOLAR := TO_DATE('02-03-' || ANNO, 'DD-MM-YYYY');
-----
INSERT INTO asistencia_periodo (
```

```

    periodo, personal)
SELECT aPERIODO, ps.personal
FROM personal_select ps
WHERE ps.spid = aSPID and
    ps.personal NOT IN (SELECT sp1.personal
                        FROM asistencia_periodo sp1
                        WHERE sp1.periodo = aPERIODO);

```

```

update asistencia_periodo
set meses_trabajados =
( SELECT nvl((F_FIN_PERIODO - pp3.fecha_ingreso) * 12 / 365,0)
  from personal pp3
  where pp3.personal = asistencia_periodo.personal), en_cese_id =
( SELECT nvl(1,0) from personal pp4
  where pp4.personal = asistencia_periodo.personal and
    pp4.fecha_cese >= F_INI_PERIODO and
    pp4.fecha_cese <= F_FIN_PERIODO),
  en_ingreso_id =
( SELECT nvl(1,0) from personal pp5
  where pp5.personal = asistencia_periodo.personal and
    pp5.fecha_ingreso >= F_INI_PERIODO and
    pp5.fecha_ingreso <= F_FIN_PERIODO),
  meses_gratificacion =
( SELECT fp_pp11_meses_validos (pp6.fecha_ingreso,
  F_INI_GRATIFICA, F_FIN_GRATIFICA)
  from personal pp6
  where pp6.personal = asistencia_periodo.personal),
  meses_proyeccion_quinta =
( SELECT fp_pp11_meses_validos (F_INI_ANNO, F_INI_ANNO,
  nvl.fecha_cese,F_FIN_ANNO))
  from personal pp6
  where pp6.personal = asistencia_periodo.personal),
  dias_periodo =
( SELECT fp_pp05_dias_periodo ( pp8.fecha_ingreso,
  pp8.fecha_reingreso, pp8.fecha_cese, F_INI_PERIODO,
  F_FIN_PERIODO)
  from personal pp8
  where pp8.personal = asistencia_periodo.personal),
  meses_escolaridad =
( SELECT fp_pp11_meses_validos (pp6.fecha_ingreso,
  F_INI_ESCOLAR, F_FIN_ESCOLAR)
  from personal pp6
  where pp6.personal = asistencia_periodo.personal)
  where periodo = aPERIODO and
  personal IN (select personal_select.personal
  from personal_select

```


where spid = aSPID);

commit;

END;

-- *****

A N E X O B

Triggers

EJEMPLO 1:

Este trigger inserta datos actualizados en la Tabla de Personal, en función a todos los datos.

```
-- *****
```

DECLARE

```
fFECHA date;  
filas integer;  
nSPID integer;
```

BEGIN

```
:new.cargo_actual := fnv13(:new.cargo, :new.cargo_encargado,  
:new.cargo_designado);  
:new.categoria_actual := fnv13(:new.categoria, :new.categoria_encargada,  
:new.categoria_designada);  
  
if fp_ut05_fecha_ok(:new.fecha_cargo_encargado) = 1 then  
:new.fecha_cargo_actual := :new.fecha_cargo_encargado;  
else  
:new.fecha_cargo_actual := :new.fecha_cargo;  
end if;  
  
ffecha := fp_ut04_fecha(sysdate);  
sp_pp70_asigna_sueldo_personal  
(:new.personal,:new.categoria,'C',ffecha);  
sp_pp70_asigna_sueldo_personal  
(:new.personal,:new.categoria_encargada,'E',ffecha);  
sp_pp70_asigna_sueldo_personal  
(:new.personal,:new.categoria_designada,'D',ffecha);  
  
if rtrim(ltrim(:new.promocion_ena)) = " then  
:new.promocion_ena := '0';  
end if;  
:new.fecha_creacion := sysdate;  
:new.fecha_modificacion := sysdate;  
:new.usuario := fp_ut24_lee_usuario;
```

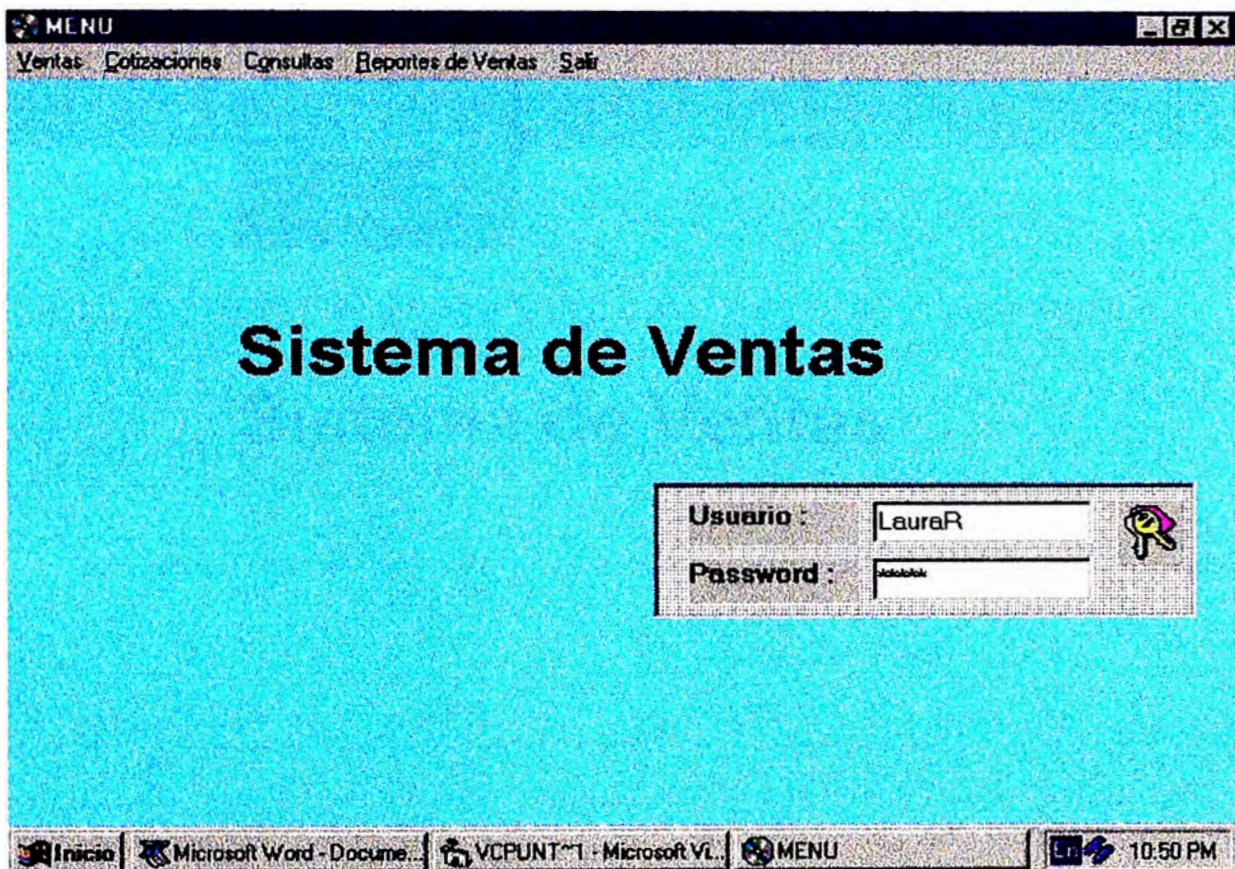
```

nSPID := fp_ut00_spid;
insert into personal
(operacion, fecha_movimiento, spid, personal, nombre,apellido, apellido_esposo,
ruta_bmp, ubicacion_direccion, direccion, telefono, fecha_nacimiento,
ubicacion_nacimiento, lugar_nacimiento,estado_civil, sexo, grupo_sanguineo, ipss,
le, ruc, carnet_extranjeria, pasaporte, categoria, cargo, nominativo, area,
ubicacion_personal, categoria_encargada, cargo_encargado, categoria_actual,
cargo_actual, encargatura, area_evaluacion, ubicacion_destaque,
direccion_destaque, regimen_laboral, planilla, centro_costo, condicion_laboral,
grupo_evaluacion, tipo_trabajador, legajo, resguardo_id, smf_id, en_afp_id, afp,
codigo_afp, en_cts_id, activo_id, ena_id, anno_ena, promocion_ena, banco_cts,
moneda_cts, cuenta_cts, banco_pago, moneda_pago, cuenta_pago,
fecha_primer_contrato,fecha_ingreso, fecha_reingreso, fecha_cese,
fecha_regimen, fecha_cargo, fecha_cargo_encargado,
fecha_cargo_actual, fecha_ingreso_maritima, fecha_cts, numero_tarjeta,
autoriza_justificacion_id, titulo_ena, indicado_id, fecha_creacion,
fecha_modificacion,usuario, ubicacion_detaque, activo_juvenil_id, funcion_juvenil )
values
('ins', sysdate, nspid, :new.personal,:new.nombre,:new.apellido,
new.apellido_esposo, :new.ruta_bmp, :new.ubicacion_direccion, :new.direccion,
:new.telefono, :new.fecha_nacimiento, :new.ubicacion_nacimiento,
:new.lugar_nacimiento,:new.estado_civil, :new.sexo, :new.grupo_sanguineo,
:new.ipss,:new.le, :new.lm, :new.ruc, :new.carnet_extranjeria,
:new.pasaporte, :new.categoria, :new.cargo, :new.nominativo, new.area,
:new.ubicacion_personal, :new.categoria_encargada, :new.cargo_encargado,
:new.categoria_actual, :new.cargo_actual, :new.encargatura, :new.area_evaluacion,
:new.ubicacion_destaque, :new.direccion_destaque, :new.regimen_laboral,
:new.planilla, :new.centro_costo, :new.condicion_laboral,
:new.grupo_evaluacion, :new.tipo_trabajador,:new.legajo,
:new.resguardo_id, :new.smf_id, :new.en_afp_id,
:new.afp, :new.codigo_afp, :new.en_cts_id, :new.activo_id, :new.ena_id,
:new.anno_ena, :new.promocion_ena, :new.banco_cts, :new.moneda_cts,
:new.cuenta_cts, :new.banco_pago, :new.moneda_pago, :new.cuenta_pago,
:new.fecha_primer_contrato,:new.fecha_ingreso, :new.fecha_reingreso,
:new.fecha_cese, :new.fecha_regimen, :new.fecha_cargo,
:new.fecha_cargo_encargado, :new.fecha_cargo_actual,
:new.fecha_ingreso_maritima, :new.fecha_cts,
:new.numero_tarjeta, :new.autoriza_justificacion_id, :new.titulo_ena,
:new.indicado_id, :new.fecha_creacion, :new.fecha_modificacion,:new.usuario,
:new.ubicacion_detaque, :new.activo_juvenil_id, :new.funcion_juvenil);
END;

```


ANEXO C

En esta sección de anexos presentamos algunas pantallas prototipo que han sido desarrolladas en Visual Basic. Estas pantallas pertenecen al Sistema Integral de Ventas.

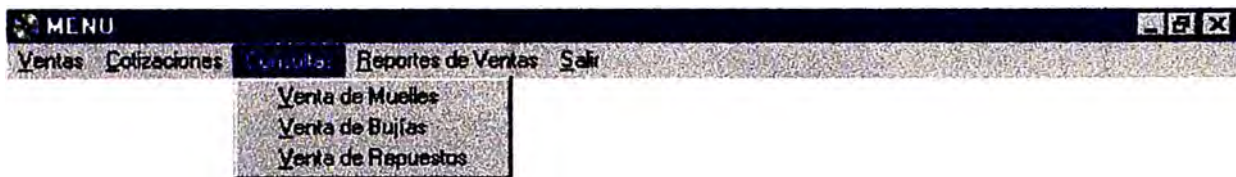


Esta es la pantalla de ingreso al Sistema de Ventas, para el acceso a este sistema deberá ingresarse el Login del Usuario y el Password correspondiente por medidas de seguridad.

Menú del Sistema de Ventas

El Menú de este sistema se dividirá cuatro menús que serán:

- Ventas, aquí se registran las ventas diarias y la emisión de facturas y boletas
- Cotizaciones, registra las cotizaciones valorizadas realizadas por los Concesionarios.
- Consultas, presenta los registros de ventas por Muelles, Bujías y Repuestos.
- Reporte de Ventas, emite todos los tipos de reportes generados por las ventas, pedidos y cotizaciones, mensuales y diarias.



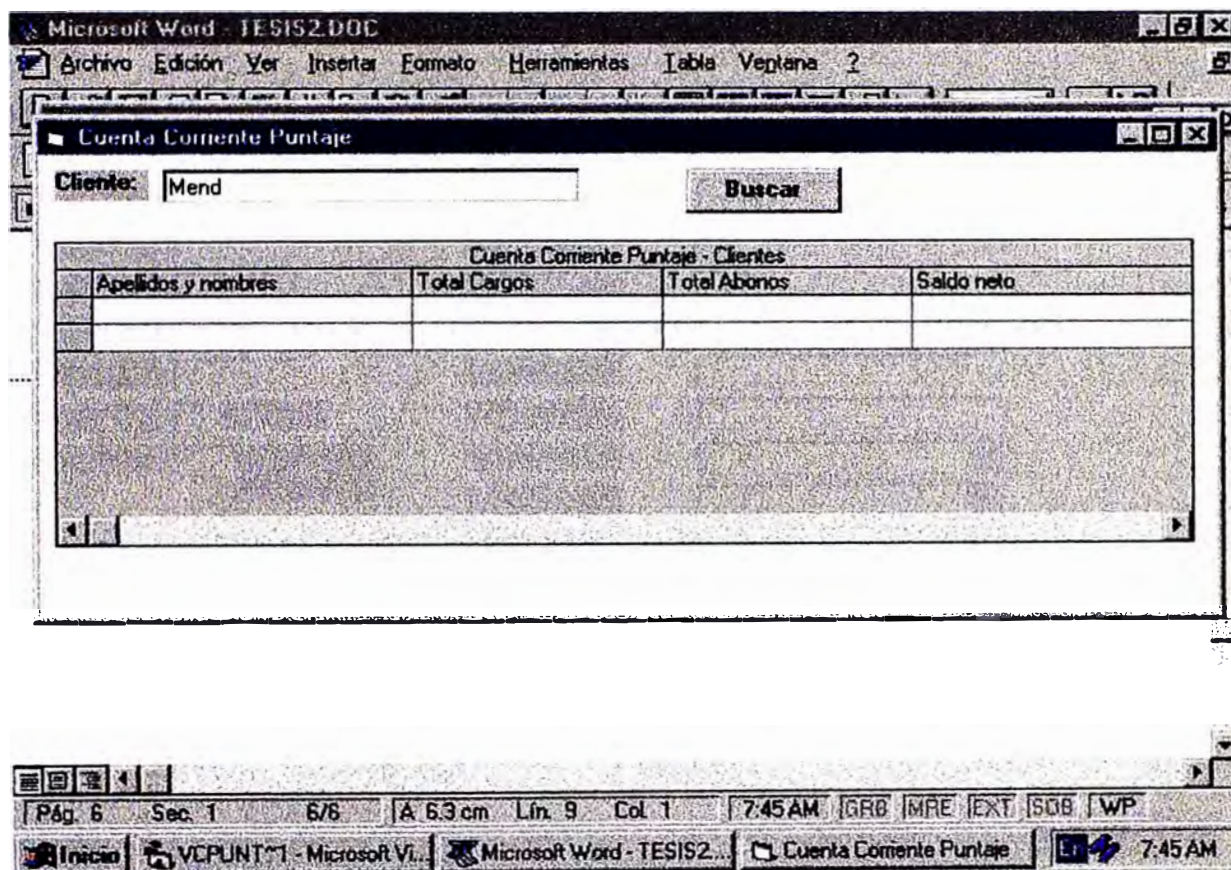
Sistema de Ventas



Pantallas de Cuentas Corrientes

Las pantallas de Cuentas Corrientes también otorgarán la facilidad de hacer las búsquedas por nombres de Concesionarios o clientes.

En una “grilla” por ejemplo, se podrán displayar los datos correspondientes al Total Cargos, Total Abonos y al Saldo Neto por cada Concesionario.



Pantalla de Ingreso de Datos de los Cientes y Concesionarios

Aqui se registran los datos personales de los clientes y concesionarios, tales como Código, Nombre, Dirección, etc.

Datos del Cliente

Código	<input type="text"/>	Fecha de Nacimiento	<input type="text"/>	Sexo	<input type="checkbox"/> (F.M)
Nombre	<input type="text"/>	Profesión	<input type="text"/>		
Dirección	<input type="text"/>	Ocupación	<input type="text"/>		
Distrito	<input type="text"/>	Empresa	<input type="text"/>		
Teléfono 1	<input type="text"/>	Teléfono 2	<input type="text"/>		
Fax	<input type="text"/>	Tarjeta de Crédito	<input type="text"/>		
RUC/LE	<input type="text"/>	Lugar visitado	<input type="text"/>		
Estado Civil	<input type="checkbox"/> (S.C.D.V)	Lugar por visitar	<input type="text"/>		
Número de hijos	<input type="text"/> ENTER para actualizar	Hobbie 1	<input type="text"/>		
		Hobbie 2	<input type="text"/>		
		Hobbie 3	<input type="text"/>		

Inicio Microsoft Word - TESIS2... VCPUNT~1 - Microsoft Vi... Datos del Cliente 11:15 PM