

25



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGON**

**SISTEMA DE RECAUDACION
E INFORMACION PREDIAL, CLIENTE-SERVIDOR**

**TESIS
QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION**

PRESENTA:
HECTOR BERNARDO ORTEGA GINES

ASESOR:
ING. JUAN GASTALDI PEREZ

MEXICO

287402

2000



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
CAMPUS ARAGÓN

SECRETARÍA ACADÉMICA

UNIVERSIDAD NACIONAL
AVENIDA LEONARDO
MEXICO

Ing. JESÚS DÍAZ BARRIGA ARCEO
Jefe de la Carrera de Ingeniería en Computación,
Presente.

En atención a la solicitud de fecha 9 de mayo del año en curso, por la que se comunica que el alumno HÉCTOR BERNARDO ORTEGA GINES, de la carrera de Ingeniero en Computación, ha concluido su trabajo de investigación intitulado "SISTEMA DE RECAUDACIÓN E INFORMACIÓN PREDIAL, CLIENTE-SERVIDOR", y como el mismo ha sido revisado y aprobado por usted, se autoriza su impresión; así como la iniciación de los trámites correspondientes para la celebración del Examen Profesional.

Sin otro particular, reitero a usted las seguridades de mi atenta consideración.

Atentamente
"POR MI RAZA HABLARÁ EL ESPÍRITU"
San Juan de Aragón, México, 10 de mayo del 2000
EL SECRETARIO

Lic. ALBERTO IBARRA ROSAS

C p Asesor de Tesis.
C p Interesado. ✓

AIR/VSR/vr

Recibido
14/05/2000

Recibido
14/05/2000



LIBERTAD NACIONAL
AYER MAH
MAYE

ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ARAGÓN

JEFATURA DE INGENIERÍA EN
COMPUTACIÓN

OFICIO: ENAR/JAC/210/00/A

ASUNTO: Asignación de Jurado

ENEP - MEXICO
MAY 9 2000
RECIBIDO

LIC. ALBERTO IBARRA ROSAS
Secretario Académico
Presente.

Por este conducto me permito presentar a usted, nombre de los profesores que sugiero integren el Síndico del Examen Profesional del alumno HÉCTOR BERNARDO ORTEGA GINES, que presenta el tema de tesis: "SISTEMA DE RECAUDACIÓN E INFORMACIÓN PREDIAL, CLIENTE-SERVIDOR".

- PRESIDENTE: ING. JOSÉ GONZÁLEZ BEDOLLA
- VOCAL: ING. JUAN GASTALDI PÉREZ
- SECRETARIO: ING. ROBERTO BLANCO BAUTISTA
- SUPLENTE: LIC. ALBERTO IBARRA ROSAS
- SUPLENTE: LIC. MA. DEL PILAR GARCÍA VILLANUEVA

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

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

A T E N T A M E N T E
"POR MI RAZA HABLARÁ EL ESPÍRITU"
San Juan de Aragón, Edo. de México, Mayo 09 del 2000.
EL JEFE DE CARRERA

ING. JESÚS DÍAZ BARRIGA ARCEO

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

JDBA/gga.

DEDICATORIA

*En donde quiera que estés, es donde
tienes que empezar. La diferencia del
mañana radica en el esfuerzo de hoy.*

Dedico este trabajo de tesis a la Sra. Aurora Gines, mi madre, que con trabajo constante me proporciono lo necesario para realizar una profesión.

A mi esposa Alicia, que fue la motivadora para iniciar por segunda ocasión la carrera, así como su ayuda y apoyo continuo.

A mi hijo Marco Antonio, quien no me hace olvidar la chispa de la vida.

Y por supuesto a Dios, que me ha permitido recuperarme.

AGRADECIMIENTOS

No está por demás dar las gracias a todas las personas que permitieron dar inicio y fin a este trabajo de tesis. Comenzando con Marco Antonio Pacheco, jefe de unidad de Redes Locales en la Secretaría de Finanzas quien me recomendó para el desarrollo de este proyecto. A Eduardo Maldonado, Gerardo Ruiz y Malena Ríos, jefes de departamento y de oficina que laboran actualmente en la Secretaría de Finanzas, otorgando las facilidades necesarias para realizar la filmación del sistema, operando en la infraestructura requerida para su funcionamiento.

Alicia, gracias por el apoyo que tengo de ti para con la elaboración del presente trabajo. A mis amigos, por el apoyo para conmigo durante la estadia en la Universidad.

Gracias al Ing. Juan Gastaldi, por la dedicacion y direccion para realizar este trabajo de tesis. Y sobre todo por brindar el apoyo y colaboración incondicional en los diferentes proyectos de la ENEP Aragón, que le han permitido a la escuela de ingeniería tener la proyeccion necesaria.

INDICE

<i>Contenido</i>	<i>Página</i>
INTRODUCCION	6
Planteamiento del problema	7
Hipótesis	8
Propuesta del sistema	8
Restricciones de software y hardware, para el desarrollo del sistema	8
Metodología para el análisis y diseño	9
CAPITULO 1. Infraestructura Existente	11
Elementos de la red LAN	11
Proceso centralizado	11
Proceso de red	12
Proceso distribuido	12
Servidores	12
Servidor de comunicaciones	12
Estación de trabajo	13
Sistema operativo de red	13
Tarjeta de interfase	13
Paquetes	13
Acceso a la red	14
Protocolo de transporte	14
Protocolo Cliente/Servidor	14
Arquitectura abierta	15
Plataformas de servidor abiertas	15
Simulación de una terminal	15
Pantallas de simulación	15
Gateways	16
NetWare Services for SAA	17
SNA. System network architecture	17
Conceptos del esquema corporativo SNA	17
Servicios a las sesiones en la red	18
Usuario final (EU: End User)	18
Unidades lógicas (LU: Logical Units)	18
Sesiones entre unidades lógicas	18

Servicios a los usuarios finales (Transaccionales)	19
Activación de una sesión	19
Desactivación de una sesión	20
Flujo de datos en sesiones LU-LU	20
Tipos de unidades lógicas	20
LUT 0 (Unidad lógica tipo cero)	20
LUT 1 (Unidad lógica tipo uno)	20
LUT 2 (Unidad lógica tipo dos)	20
LUT 3 (Unidad lógica tipo tres)	21
LUT 4 (Unidad lógica tipo cuatro)	21
LUT 6 (Unidad lógica tipo seis)	21
LUT 6.2 (Alias LU "C")	21
Unidades físicas (PU: Physical Units)	21
Tipos de unidades físicas	22
PUT5	22
PUT4	22
PUT2	22
PUT1	22
Equipo SNA instalado	22
Cuadro demostrativo del equipo	22
Unidad central de proceso IBM 4381	23
Unidad central de proceso IBM 303	24
Controlador de unidades de almacenamiento directo IBM 3880	24
Unidad de almacenamiento IBM 3380	24
Disco de estado sólido STC 4305	25
Controlador de cintas STC 3800	25
Sistema de impresión de tecnología laser 3800	26
Impresora de impacto STC 1500	26
Estación de despliegue	27
CAPITULO 2. Conectividad SNA de IBM	28
NetWare para SAA	29
Acceso completo de aplicación AS/400	30
Soporte a múltiples conexiones a host	30
Soporte a múltiples máquinas de escritorio	30
Manejo de red mejorado	30
Ambiente abierto	31
Utilerías	31
Status	31
Seguimiento para el diagnóstico	31
Configuración por menú	31
Auditoría	31
Ventaja del NetWare Name Service	31
Seguridad específica para comunicaciones	31
Implementación	32
Requerimientos de hardware	32
Software necesario	32
NetWare SNA gateway	32
Coaxial	32
Coaxmux	33
Remoto	33
Token-Ring	33
Características	34
Implementación	34
Requerimientos de hardware	34
Conexión coaxial	34

<i>Contenido</i>	<i>Página</i>
Conexión coaxial mux	34
Conexión remota	34
Token Ring	34
Software necesario	35
Especificaciones	35
NetWare SNA gateway ELS	35
NetWare 3270 LAN workstation para DOS	35
Características	36
Implementación	36
Requerimiento de hardware	36
Software necesario	36
Especificaciones	37
Instalación	37
NetWare 3270 LAN Workstation para Windows	37
Características	38
Implementación	38
Requerimiento de hardware	38
Software necesario	38
Especificaciones	38
LU (Logical Unit; unidad lógica)	38
LU tipo 0	38
LU tipo 2	38
LU tipo 3	38
LU tipo 4	39
LU tipo 6.0	39
LU tipo 6.1	39
LU tipo 6.2	39
CAPITULO 3. Empleo del Protocolo DDE con Extra for Windows	40
Particularidades de Windows	40
Formato de los mensajes	41
Bucle de mensajes	42
Intercambio dinámico de datos	42
DDE de Microsoft Visual FoxPro	43
Intercambio de datos (DDE)	43
DDE y Extra! for Windows	43
Iniciando una liga DDE	44
Nombre de servicio	44
Nombre de tema	44
Empleando mensajes de ejecución DDE	44
Allow emulator updates	44
Block emulator updates	44
Open layout	45
Pause	45
Send file	45
Recive file	45
Run macro	45
Start emulator	45
Stop emulator	45
Terminal On	45
Terminal Off	45
Empleando mensajes de petición DDE	46
Columns	46
Configuration name	46
Configuration topics	46
Cursor	46

Directory name	46
Emulator	46
File transfer	46
Keyboard	46
Keystroke	46
Layout name	47
Model	47
OIA	47
Power	47
Ps	47
Rows	47
Search	47
Sysitems	47
Timeout	48
Topic	48
Topics	48
Envío de datos a la emulación	48
Cursor	48
Keystroke	48
Ps	48
Search	48
Timeout	48
Crear vínculos entre Visual FoxPro y EXTRA!	49
Determinar errores de funciones DDE	49
Terminar una conversación con EXTRA!	50
Funciones DDE de EXTRA! para Visual FoxPro	50
CAPITULO 4. Diseño Aplicación Cliente Servidor	52
Diseño de la interse gráfica	52
Análisis de información	53
Método tradicional	53
Análisis de información	54
Axioma ANALIM	54
Axioma INSYGRAM	55
Metodología para el desarrollo de sistemas	55
Toda aplicación se genera para dar una solución a la necesidad o por iniciativa del área de computo	56
Involucrarse en la información de la aplicación	56
Funcionamiento actual	56
Si es necesario solicitar una muestra	57
Solicitar formatos de salida, con sus respectivas restricciones	57
Análisis de cada formato	58
Oraciones compuestas	58
Oraciones elementales	58
Tablas de población	59
Diagrama ENALIM	60
Diagrama ENALIM FINAL	60
Diseño de tablas, bases de datos y/o archivos	60
Eliminar campos redundantes	60
Estandarizar tipos y nombres de campos así como su longitud	61
Diccionario de datos	61
Modulación	62
Diseño de pantallas	62
Descripción de programas	67
Métodos que llaman a las clases (pantallas) del modulo servicios y procesos. formulario asiste02	67

<i>Contenido</i>	<i>Página</i>
Clases (pantallas) del módulo servicios y procesos. formulario asiste02	69
Asignación de programas para captura y aplicación	72
Pruebas parciales y en paralelo	76
Liberación	76
Mantenimiento	76
CONCLUSIONES	77
APENDICE A	79
Tecnología cliente servidor: RPC	
APENDICE B	89
Recibo emitido de adeudo predial	
GLOSARIO	90
BIBLIOGRAFIA	95

INTRODUCCION

La Secretaría de Finanzas, Tesorería, es un órgano perteneciente al gobierno del Distrito Federal, que tiene por objetivo el del coordinar, planear y administrar la recaudación de los impuestos en sus diferentes servicios y procesos, como lo son:

- Predial. Se puede consultar información de cómo inscribirse al padrón, obtener constancias de adeudos, aclarar asuntos referentes a la cuenta predial, hacer correcciones o cambios de propietario.
- Nomina. Se puede contar con información de trámites relacionados con los impuestos debidos al pago de salarios y demás prestaciones que se deriven de una relación laboral.
- Agua. Tramites relacionados con el pago de los derechos por el suministro del agua que provee el Distrito Federal.
- Tenencia y control vehicular. Referente al pago del impuesto por pertenencia y uso de vehículos.
- Otros. Trámites adicionales que puede realizar dentro de la Tesorería como los relacionados con: Espectáculos públicos; Loterías, rifas, sorteos y concursos; Impuestos de adquisición de inmuebles entre otros.

Para realizar los trámites la Tesorería cuenta con cuarenta puntos de recaudación, distribuidas en toda la ciudad que tienen por nombre Administraciones Tributarias de la Tesorería del Distrito Federal (abreviado Administraciones Tributarias, AT's). La distribución se realiza en las 16 delegaciones políticas de la ciudad de México, y donde la mayoría de los contribuyentes realizan sus pagos y/o trámites. Además, de la recaudación se encarga de administrar la distribución de los impuestos para realizar obras, como:

- Transporte público.
- Entrega de libros de texto gratuitos.
- Desazolve para evitar inundaciones.
- Y otros.

Debido a esto la Secretaría de Finanzas, Tesorería, desea impulsar un intercambio en la cultura fiscal, ofreciendo al contribuyente transparencia, seguridad, confianza y ante todo un cambio de actitud del servidor público hacia la ciudadanía. Reduciendo así, el tiempo de atención y trámites de pagos, que conllevara la eliminación de papeleo, trabas; por lo tanto, los trámites serán sencillos en los siguientes trámites:

- Inscripción al padrón
- Devoluciones
- Pago con cheque
- Aclaraciones de pagos
- Corrección del nombre de propietario
- Entre otros

Para ayudar a la administración y control de recaudación de los impuestos y/o trámites relacionados la Tesorería cuenta además de las AT's, personal capacitado; de áreas tecnológicas, como:

- Departamento de Bases de Datos. Se encarga de mantener a punto la información de los estados de pagos de los contribuyentes en sus diferentes modalidades. Así como, de la información personalizada para cada caso; por ejemplo: Agua, dirección del predio, cantidad consumida en metros cúbicos, etc.
- Redes Locales. Da mantenimiento e instala nuevo equipo dentro de la red local de la Tesorería y las redes también locales de las AT's. Así como, de la instalación del hardware y software con licencia.
- Comunicaciones y Enlaces. Se encarga de enlazar por medio de modems los cuarenta puntos de recaudación con el modulo central. También da mantenimiento a los módulos de comunicaciones.
- Proceso Central. Se encarga de vigilar la entrada y salida de datos, asignar derechos de usuario, administrar los recursos de CPU, entre otros del IBM SNA host.

Dentro del Departamento de Redes Locales, se encuentra el área de Mantenimiento y Desarrollo de Sistemas, la cual cuenta con el personal encargado de analizar, diseñar y programar los sistemas internos y externos que requiere la Tesorería, debido a esto se tienen sistemas desarrollados en la Tesorería sin tener que contratar servicios de desarrollo externos; como sistemas en desarrollo y en funcionamiento, se tiene:

- Sistema de Tenencia. Captura los datos y proporciona los talones de pago sobre la tenencia vehicular, y da mantenimiento de la información registrada. Todo esto es tiempo de ejecución.
- Sistema de Cajas. Es el sistema encargado de realizar las recaudaciones de los impuestos en sus diferentes rubros. El sistema propiamente es conocido por los contribuyentes que acuden a realizar sus pagos en las cajas de las AT's. Este pago se registra de forma local y posteriormente se envía a través de la red al modulo central para su registro, y almacenamiento.
- Registro de Folios. Asigna números de folio a los engomados de tenencia de acuerdo a diferentes algoritmos, además de brindar la administración necesaria para su conteo de asignación.
- Sistema de Corte. Se tiene en dos versiones, el de las AT's y del modulo central. El primero se pone en funcionamiento al realizar los cortes de caja y cierre de cajas en las AT's durante el día y al final, para reportar las cantidades recaudadas. El segundo modulo se pone en funcionamiento al otro día por la mañana para obtener los reportes de la recaudación en cada uno de los cuarenta puntos de pago de impuestos.
- Otros. Sistema de Nómina, administra el pago de tiempo extra y excedente; Sistema de Quejas y Denuncias, recaba las diferentes quejas y denuncias de los contribuyentes; Sistema de Reportes y Fallas, captura las diferentes fallas en red, comunicaciones y fallas de los sistemas para darles solución en las áreas correspondientes.

Para su desarrollo se tienen varios proyectos entre los cuales y para el desarrollo de la tesis en cuestión se tiene el proyecto de Kiosco de Consultas de Adeudos, que servirá de vinculo y proporcionará un acercamiento entre contribuyente y la Secretaría de Finanzas, para alcanzar la modernización administrativa en todos y cada uno de los rubros, en beneficio de la ciudadanía. De este modo se invitará al ciudadano a participar, en el esfuerzo por mejorar la calidad del servicio que se le ofrece. Creando una base de consultas, donde el contribuyente pueda contar con los elementos de información necesarios de su estado de adeudo actual. Este proyecto de Kiosco de Consultas de Adeudos debe dar soporte a los elementos de información y atención posibles, para que el contribuyente realice sus trámites.

PLANTEAMIENTO DEL PROBLEMA

La Tesorería, tiene actualmente procesos de atención al ciudadano; tales se encuentran en operación en las AT's. En estudios recientes por personal interno y además de consultas realizadas a los contribuyentes, se llega a la conclusión que el ciudadano promedio que asiste a realizar sus pagos se encuentra descontento por el largo proceso para realizar una queja, reclamación y pago de impuestos. Por lo tanto, Tesorería se encuentra preocupada por esto y ha propuesto varias soluciones para combatir estas problemáticas. La que más nos interesa para el caso de esta tesis, es el pago de los impuestos en los rubros de predial, agua y tenencia. Ya que actualmente para realizar un pago de alguno de estos rubros (o cualquier otro), hay que realizar largas filas y en periodo de vencimiento la problemática se agrava. A continuación detallo antecedentes de cómo se realiza un proceso manual para consultar los adeudos y saldos de predial:

1. Actualmente, se requiere que el contribuyente lleve su recibo anterior a la ventanilla correspondiente.
2. Enseguida la persona a cargo teclea el número de cuenta impreso en el recibo anterior, y de esta forma puede consultar los adeudos del contribuyente,
3. A continuación se imprime el recibo de adeudos con la especificación de reposición de recibo, comprobante de no adeudo, o impresión de adeudo, según sea el caso.
4. Si el recibo impreso es de adeudo, la persona puede pasar con él a las cajas y saldar su cuenta.
5. En alguna de las oficinas también se puede consultar su saldo a través de búsquedas por nombre.
6. Cada operación lleva un número de folio o número de operación la cual se graba en la base de datos local.

Para el predial es un formato impreso en impresora de matriz porque cuenta con original y dos copias. El número de impresoras en cada sucursal varía dependiendo de la carga de trabajo y del número de nodos que existan.

De lo descrito anteriormente, la Tesorería tiene la necesidad de agilizar y simplificar los trámites de consultas de adeudos de predial y finalmente de tenencia, se requiere de un medio de automatización que pueda reducir el tiempo requerido para consultar saldos adeudos de pagos especialmente en periodos de término de los pagos, en los cuales la demanda de atención a los usuarios es mucho mayor.

Por lo tanto, la Tesorería tiene un gran interés de ofrecer un mejor servicio a sus contribuyentes, con menos complicaciones, efectividad y rapidez; además tratando de integrar la participación ciudadana para colaborar conjuntamente al buen funcionamiento de la ciudad. Esperando como resultados finales el proceso automatizado que describe:

1. En la terminal, emular una terminal VT 3270.
2. Presentar en pantalla un ambiente amigable, ocultando de pantalla la emulación al host central en tiempo de ejecución.
3. Llevar a cabo procesos locales. Grabar en el servidor local para que el contribuyente pase a cajas y ahí se marque como pagado. Posteriormente se llevarán a cabo procesos de actualización Batch para llevar la información al host central.

HIPOTESIS

Se ofrecerá al usuario una interfaz multimedia simple, fácil y entendible; permitiendo consultas en línea de registros de adeudos y saldos, evitando largos trámites burocráticos. Alcanzando la modernización administrativa en todos y cada uno de los rubros, en beneficio de la ciudadanía. Reduciendo el tiempo de atención y trámite de pagos, eliminando papeleo, trabas e inconvenientes; impulsando un cambio en la cultura fiscal, ofreciendo al contribuyente transparencia, seguridad, y otra imagen del servidor público hacia la ciudadanía.

PROPUESTA DEL SISTEMA

El objetivo principal de la realización de un Kiosco de Consultas de Adeudos es simplificar los trámites administrativos para llevar a cabo consultas de saldos de predial y posteriormente de tenencia; de forma que el ciudadano pueda realizar desde este módulo dichas consultas y evitarse trámites burocráticos. Tal proceso debe valerse de un medio automatizado, en este caso de un sistema informático que sea capaz de explotar la infraestructura existente de software y hardware.

Por lo anterior, el diseño y análisis del sistema se encomienda a la Unidad Departamental de Atención a Redes Locales, de la Secretaría de Finanzas; al que se le denominará Jefe de Sistemas; el a su vez encomienda la elaboración de la aplicación al que llamare programador.

La función principal del Jefe de Sistemas en el área de sistemas es controlar los tiempos de entrega y de observar que el producto del sistema se encuentre dentro de los lineamientos de la Secretaría de Finanzas. El Jefe de Sistemas delega el diseño y análisis del sistema al programador, el cual se encargará de analizar, diseñar y programar la aplicación. El nombre del sistema propuesto por el programador es: Sistema de Recaudación e Información Predial, Cliente-Servidor; que al término deberá ser un sistema con una interfaz simple, y sencilla; con la cuál los contribuyentes se sientan invitados a integrarse a esta tecnología.

Restricciones de software y hardware para el desarrollo del sistema

Sistema de Recaudación e Información Predial, Cliente-Servidor, es el nombre del sistema. El por qué usar el nombre cliente-servidor?, esto se debe a que se planea a mediano plazo (segunda etapa) que el sistema utilice las bondades de la arquitectura cliente-servidor. Por lo tanto, la aplicación en la etapa de diseño y prueba (primera etapa) se ejecutará en modo de red. En los siguientes renglones se explica el motivo.

Para el desarrollo de este proyecto debe aprovecharse el conjunto integrado de software con licencia, sistema operativo de red NetWare 4.1, infraestructura de redes locales, redes de comunicación, equipo (estaciones de trabajo PC, servidores, y un mainframe de la serie 308x) y hardware; ya existente en la Secretaría de Finanzas. Toda esta infraestructura se encuentra instalada, en los cuarenta puntos de recaudación (Administraciones Tributarias Locales, ATL) y el módulo central (Dirección General de Informática, DGI; donde se encuentra el sistema central).

De la infraestructura descrita, se aprovecharán los productos de Novell de donde es posible establecer conectividad con soluciones que soportan computación en red.

En el ambiente NetWare se puede realizar la conectividad al protocolo IBM SNA, Systems NetWare Architecture; estrategia principal de IBM para el uso de redes, donde SNA esta compuesta por una gran variedad de productos de hardware y software que interactuan entre sí. El objetivo de esta es proveer a la aplicación cliente (sistema) una conexión desde una estación de trabajo en la red hacia el host (equipo IBM SNA mainframe, que es la computadora central en un entorno de procesamiento en tiempo compartido o distribuido), este nodo puede trabajar con múltiples sesiones al host emulando una terminal nativa de éste. La terminal emulada es un dispositivo 3270 que pertenece a la familia de sistemas grandes de IBM. Además, el Sistema de Recaudación e Información Predial, Cliente-Servidor, en este caso la aplicación cliente, llevará a cabo procesos locales, los cuales se grabaran en el servidor local, el contribuyente podrá pasar a cajas, ahí se marcan sus registros como pagados. Al final del día se llevará a cabo procesos de actualización Batch para enviar la información guardada localmente, al host central.

El sistema debe ser desarrollado como un front end que explotara una emulación de una terminal 3270 para obtener los servicios del host central. La programación del front end será en Visual Fox Pro que utilizara el protocolo de comunicaciones DDE de Windows para el intercambio dinámico de datos entre dos aplicaciones.

La aplicación cliente en este caso el sistema de recaudación e información predial cliente-servidor llevará a cabo procesos locales, los cuales se grabaran en el servidor local, el contribuyente podrá pasar a cajas y ahí se marcan sus registros como pagados. Al finalizar el día se arrancaran procesos de actualización batch para enviar las bases de datos generadas por la aplicación cliente al host central. De lo anterior, la información generada podrá ser explotada por otros sistemas orientados al análisis de datos, brindando soporte a la planeación y la toma de decisiones a corto y mediano plazo. Los elementos de arquitectura son: sistemas, archivos fuente, bases de datos destino, software de administración de bases de datos, metadatos como un modelo de datos y herramientas de consulta residentes en estaciones de trabajo clientes. Además de una infraestructura existente de arquitectura de red, que administran los dispositivos periféricos, y computadoras. La red es de tipo LAN (Red de Area Local), en donde la comunicación está diseñada en un área geográfica limitada. Brindando los siguientes beneficios: conexión a diversidad de equipo, compartir recursos de alto costo o difícil acceso, flexibilidad de crecimiento, conexiones externas, acceso a librerías de software, control centralizado o distribuido y compartir información.

La arquitectura cliente-servidor se explica con más detalle en el apéndice A; es un proceso computarizado caracterizado por hacer referencia a una red basada en un servidor, donde la aplicación, nodo (estación de trabajo PC), conocida como cliente y otra aplicación o nodo, conocida como servidor, trabajan y se comunican a través de la red. Donde también los componentes del cliente y el servidor pueden operar en plataformas separadas. El cliente como el servidor pueden crecer en forma independiente. El servidor puede atender múltiples clientes. Se aprovecharan los productos de Novell para obtener conexiones en red. La plataforma NetWare soporta la Arquitectura de Sistemas de Red (SNA). Protocolos de IBM para la especificación de conexiones terminal a macrocomputadora. SNA es la estrategia principal de IBM para el uso de redes, compuesta por una gran variedad de productos de hardware y software que interactuan entre sí. Permitiendo diseñar la aplicación cliente con un front end en Visual Fox Pro conectado al emulador 3270, que trabajara en una estación de trabajo en red, el equipo se conectará al host (equipo IBM SNA mainframe), teniendo la posibilidad de tener varias sesiones emulando una terminal nativa de éste, en este caso una terminal 3270.

Para realizar tal conexión es necesario contar con NetWare for SAA en el servidor. Utilizándolo se conecta el mainframe a las redes manteniendo la seguridad, desempeño, servicios y administración integrados al NetWare. Permitiendo el acceso de estaciones de DOS, Windows y Unix, además soporta hasta 506 sesiones de impresión y despliegue en múltiples anfitriones (host). Además del acceso transparente a sesiones AS/400, NetWare for SAA es implementado en un NLM (NetWare Loadable Module) Este NLM es cargado en el servidor NetWare 3.12 y/o 4.0 que funciona como servidor de archivos o como servidor dedicado exclusivamente a las comunicaciones. Soporta conexiones Token-Ring (usando controladores 317x, procesadores 37x5 con controladores Token-Ring en sistemas 3370 y AS400), SDLC y QLLC/X.25 al host.

METODOLOGIA PARA EL ANALISIS Y DISEÑO

La metodología para lograr una óptima solución del proyecto se dará en los siguientes:

- El objetivo es realizar un kiosco de consultas de adeudos de predial, tal proceso debe de ser un medio automatizado, en este caso un sistema informático que contenga una interfaz simple de manipular y que invite al contribuyente (persona común, que no tenga conocimientos de aspecto informático) a participar en esta tecnología; este sistema por lo tanto, debe de considerar un diseño de tipo multimedia con sonido, imágenes, presentación y otras de sus bondades.
- El modelado de las bases de datos dependerá prácticamente de la estructura actual del almacén de datos contenido en el mainframe para el cobro del predial; educandolo al análisis obtenido conforme a las vistas y reportes, que necesite el sistema-para generar la información adecuada. Definiendo las entidades en la fase de

aceptación de requerimientos, posteriormente utilizando diagramas Enalim para facilitar el proceso de diseño de las entidades y las relaciones entre ellas, así como los atributos de cada una.

- De lo que se refiere a la infraestructura existente, se busca la combinación de los elementos existentes y los adicionales. En el presente proyecto se explica las configuraciones y conexiones de red y/o comunicaciones existentes en la Secretaría de Finanzas y sus cuarenta puntos de recaudación, para brindar el conocimiento del proceso que logra la acción de computo cliente-servidor, acceso al host (capítulo uno y dos).

INFRAESTRUCTURA EXISTENTE

En este capítulo se describe la infraestructura existente dentro de la Dirección General de Informática abreviado DGI y los cuarenta puntos de recaudación que las llamare ATL por sus siglas Administración Tributaria Local, ya que considero importante la tecnología instalada en donde trabajara el software de este tema de tesis. El Sistema de Recaudación e Información Predial, (cliente/servidor), trabaja en una red LAN con emulaciones al Host central por medio del producto Extra! for Windows Versión 5.0 de Attachmate, además se cuenta con otro producto de Novell el cual establecerá conectividad a red IBM SNA (este se explica con más detalle en el capítulo dos).

Con la llegada de la tecnología se lograron crear dispositivos que permitían almacenar grandes cantidades de información, capacidades que iban desde 1 Megabyte hasta 100 Megabytes. Una desventaja de esta tecnología era el alto costo que significaba la adquisición de un disco duro. Pero, la necesidad era inmediata, los usuarios tenían la necesidad de compartir información y programas en forma simultánea, así como el poder compartir recursos de relativa baja utilización y alto costo llevó a diversos fabricantes y desarrolladores a la idea de las redes locales.

Las redes locales LAN (Local Area Netware) habían nacido. Las primeras redes locales estaban basadas en Servidores de Disco (Disk Servers). Esto equipos permitían a cada usuario el mismo acceso a todas las partes del disco, causando obvios problemas de seguridad y de integridad de datos.

La compañía de Novell Inc. Fue la primera en introducir un Servidor de Archivos (File Server) en el que todos los usuarios pueden tener acceso a la misma información, compartiendo archivos y contando con niveles de seguridad, lo que permite la integridad de la información no sea violada.

Novell Inc. Basó su investigación y desarrollo en la idea de que es el software de la red, no el hardware, el que hace la diferencia en la operación de la red. En la actualidad Novell soporta más de 100 tipos de redes. Las tendencias indican una orientación hacia la conectividad de datos. No solo en el envío de la información de una computadora a otra, sino sobre todo en la distribución del procesamiento a lo largo de grandes redes.

Novell Inc. Fue el pionero en 1986, una vez más, al lanzar la tecnología de protocolo abierto para ser una arquitectura universal de conectividad bajo NetWare.

ELEMENTOS DE LA RED LAN

El propósito de tener una red es para compartir servicios. Compartir servicios son funciones de una computadora, que son utilizados en forma simultánea por muchos clientes en cualquier lugar de la red.

Algunos de los ejemplos de servicios básicos de red son: almacenamiento de archivos, administración central y distribuido, seguridad, impresión, protección de datos, respaldos, comunicación, conectividad.

Proceso centralizado

El proceso centralizado es aquel en que la información reside en una sola máquina, a la que se encuentran conectadas un conjunto de terminales tontas (sin procesador ni memoria), todas las peticiones y modificaciones se llevan a cabo en la computadora central.

Proceso de red

Es un conjunto de computadoras que comparten recursos entre si, cuentan con un lugar común de almacenamiento de datos el cual les permite intercambiar información entre los diferentes usuarios que tengan asignada una cuenta en la red.

Toda la información esta almacenada en un sólo lugar, en una computadora denominada servidor de archivos (File Server), al que todos los usuarios de la red tienen acceso desde su propia estación de trabajo (Workstation). El servidor de archivos se encarga de administrar y responder a las peticiones de las estaciones de trabajo. El lugar donde se almacena la información de todos los integrantes de la red es en disco duro del servidor de archivos.

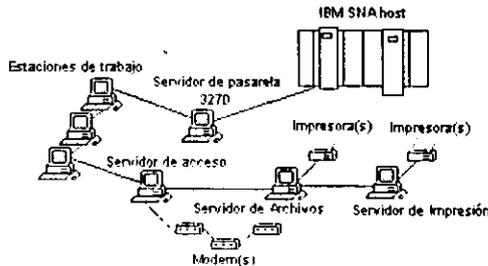
Proceso distribuido

El proceso distribuido es aquel en que la información reside en una máquina, a la que se encuentran conectadas un conjunto de terminales inteligentes (tienen procesador y memoria), todas las modificaciones y procesos se llevan a cabo en cada terminal y sólo será requerida la computadora principal para borrar y guardar archivos. Este tipo de operación de procesamiento distribuido es la que se realiza dentro de la DGI y las ATLS, empleando una red LAN remota y una red SNA geográficamente distanciadas y enlazada por telecomunicaciones

Servidores

Es la computadora central que nos permite compartir recursos y es donde se encuentra alojado el sistema operativo de la red. Además provee de acceso controlado a los archivos, permitiendo compartir impresoras y otros recursos de la red. Actualmente en la DGI y en las ATLS se tienen computadoras con procesadores 80486/120, 64 Mb en RAM, dos discos duros de 2 Mb. A pesar de esta configuración comparada con los nuevos microprocesadores hasta el advenimiento de nuevas necesidades, los servidores cuentan con suficiente capacidad de procesamiento para llevar las tareas de red.

Normalmente, se piensa que el servidor de la red es como una pieza de hardware, el término servidor es actualmente la descripción funcional de un papel que está determinado por el software.



Según se muestra en la figura anterior, la ejecución del software en una computadora conectada a la red, permite que la computadora actúe como un servidor de archivos, un servidor de impresora o un servidor de comunicaciones. Como su nombre indica, los servidores de archivos permiten a los PC clientes, compartir archivos almacenados en sus discos duros de la red. El sistema operativo NetWare de Novell, le permite combinar las funciones de servidor de archivos e impresora en el mismo PC, o establecer los servidores de impresora separados. Actuando los PC como servidor de archivos no pueden ejecutar aplicaciones, pero el software del servidor de impresora puede residir en un PC que ejecuta programas de aplicación. En NetWare, el software del servidor de comunicaciones, normalmente se ejecuta en uno separado, un PC dedicado. Hay dos tipos de servidores de comunicaciones, un servidor de acceso y un servidor pasarela (gateway), que proporcionan las conexiones a las computadoras fuera de la red local.

Servidor de comunicaciones

Dentro de las ATLS, el servidor de archivos, servidor de impresoras y el servidor de comunicaciones son el mismo en un PC, claramente desgastando el poder de proceso. El servidor de comunicaciones puede colocarse donde exista una línea telefónica en este caso de un módem y/o un radio módem. Este servidor proporciona conexiones en tiempo real

entre PC clientes y los canales de comunicación. El término "servidor de comunicaciones" cubre una variedad de software que puede realizar diferentes tareas.

Los servidores de comunicaciones como mencione antes pueden actuar como gateways (pasarelas) del mainframe (computadora central), permitiendo a los PC clientes compartir el canal de comunicaciones del mainframe. El PC gateway contiene adaptadores internos y software que hacen posible el traslado entre esquemas de comunicaciones del mainframe, 3270 SNA de IBM y la red LAN. El software que se ejecuta en el gateway de la estación conecta la tarjeta de comunicaciones del gateway de la LAN y el software de simulación de terminal ejecutándose en cada PC de la red LAN. El software de simulación y el software de gateway se comunican a través del protocolo IPX/SPX, que es el administrador de la red Novell.

El producto añadido de las estaciones PC, es el simulador Extra de Attachmate; que permite actuar a los PC como terminales del mainframe de IBM.

Estación de trabajo

Las estaciones de trabajo son microcomputadoras interconectadas por una tarjeta de interfase. Ellas compartirán los recursos del servidor y realizarán un proceso distribuido. El procesamiento de datos en una red es distribuido, por lo tanto el desempeño de la estación de trabajo se debe definir en función a la aplicación que se estará manejando en ella. Analizar el tipo de aplicaciones que se estarán manejando en la red es de suma importancia para lograr que la estación trabaje de forma optima.

Existen algunas reglas generales que hay que tomar en cuenta al escoger una estación de trabajo. Esta debe contar al menos con 640 Kb en memoria para aplicaciones comerciales DOS y de un mínimo de 1 Mb de memoria para aplicaciones Windows. Además, deben de tener posibilidades de crecimiento tanto en ranuras de expansión como en capacidad para colocar un disco duro o una unidad de disco. Dentro de las ATLS se tienen estaciones de trabajo con procesador 80486/60, memoria RAM de 4 Mb, monitor blanco y negro VGA, tarjeta de interfase EtherLink, y disco duro de 1 Mb.

Sistema operativo de red

Es el software que se encarga de administrar los recursos que se estarán compartiendo (discos duros, impresoras, etc.) y a los usuarios. El sistema operativo de la red se escoge según las necesidades de control de nuestra información. Existen algunas consideraciones como son: el tipo de información que se estará compartiendo, los programas que se utilizarán, quien tendrá acceso a cierta información, etc. El sistema operativo de red escogido nos debe dar toda la seguridad que se requiera dentro de la red. Esta debe ir desde que máquina se pueda usar, a que hora se puede entrar a la red y que día se puede trabajar, clave de acceso que se tendrá, los archivos que se podrán compartir a los programas que se ejecutaran.

El sistema operativo de red para la DGI y las ATLS es NetWare 4.0, este se encarga de controlar la comunicación, recursos y servicio de la red. Este sistema operativo reside en el servidor de archivos. El sistema operativo de las estaciones de trabajo es MSDOS y el sistema gráfico donde coren la mayoría de las aplicaciones de la Tesorería es Windows 3.1; los cuales se encargan de controlar la comunicación con todos los elementos internos de la estación de trabajo.

Tarjeta de interfase

Las tarjetas de interfase nos permitirán el enlace entre microcomputadoras. Existe en el mercado una gran cantidad de tarjetas de interfase. La mayoría de estas tarjetas son Ethernet, Arcnet y Token Ring. Predominando estas Ethernet y Token Ring, mientras que Arcnet tiende a desaparecer. Dentro de las ATLS se utiliza la tarjeta de interfase Fast EtherLink XL 10/100 de 3Com con Parallel Tasking II que permite mayor cantidad de datos a través de un bus PCI, obteniendo mayor capacidad de transmisión con la menor utilización del CPU.

Paquetes

La información se envía por paquetes, de la mismo forma que el correo se envía en cartas dentro de sobres. Si se ha de enviar una gran cantidad de información, se usará más de un paquete. A cada paquete se le añaden las instrucciones de direccionamiento y reensamblaje. Normalmente el tamaño de los paquetes ronda por los 618 y los 4202 bytes. Los fabricantes de las placas de interfaz de red deben suministrar los controladores que permitan los distintos tamaños de paquetes; al arrancar el sistema operativo se debe de especificar un tamaño de paquete. Puede parecer que se envían pocos paquetes por la red, pero en realidad, se envían cientos o miles de paquetes por segundo, teniendo éstos diversas direcciones de destino. Cada estación de trabajo receptora debe clasificar los paquetes dirigidos a ella y devolver o

reenviar el resto. Una red es como un sistema postal a alta velocidad en el cual se les pusiera la dirección a los paquetes, se procesaran, se enviaran y se recibirán en microsegundos con exactitud

Acceso a la red

Cuando los paquetes que se van a enviar alcanzan la placa de interfaz de la red, deben de convertirse en señales eléctricas que se puedan enviar por el cable. Cada placa de red posee un método de control de acceso, que utiliza para transferir los paquetes de la memoria al sistema de cableado. El método de acceso se puede basar en un esquema de detección de portadora, como el usado en las redes Ethernet (que es el instalado en las ATLS y la DGI), el paquete se difunde a todas las estaciones de trabajo como, si fuera una transmisión radiofónica. La estación con la detección adecuada aceptará el paquete. Las redes con paso a testigo, como ARCNET y Token Ring, envían paquetes de estación a estación, pero sólo cuando poseen el testigo.

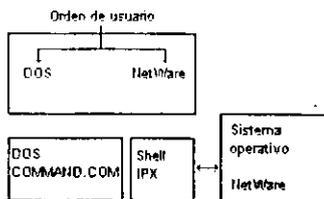
Los métodos de control de acceso funcionan en el nivel de enlace del protocolo por capas y pueden estar estandarizados por el Institute of Electrical and Electronics Engineers (IEEE). El archivo IPX.COM que se ha de ejecutar en cada estación de trabajo, debe ser personalizado para tener en cuenta el método de acceso y otras características de la placa de interfaz de red. Esto se hace enlazando el código ejecutable del controlador que viene con la placa de interfaz con el archivo IPX.COM utilizando el programa SHGEN de NetWare. Si todas las estaciones de trabajo utilizan el mismo tipo de tarjeta, sólo se debe generar un único archivo IPX.COM, pero si las estaciones de trabajo utilizan diversos tipos de placas, debe de crearse un archivo IPX para cada tipo. Esto hace recomendable utilizar placas idénticas para todas las estaciones. Sin embargo, la estrategia de independencia del medio de Novell se basa en el hecho de que se pueda enlazar cualquier placa y su controlador al archivo IPX.

Como indique anteriormente los paquetes contienen la dirección de remitente y de destino. Las direcciones están divididas en tres partes: la red, el nodo y el conector (socket). La parte de la red de la dirección determina el cable de red físico al que está conectada la estación de trabajo. Una red de redes puede consistir en dos o más redes físicamente separadas conectadas entre sí; por lo tanto, la dirección de la red es importante para dirigir el paquete a su lugar adecuado. La dirección de nodo identifica la estación de trabajo exacta. El socket conector se utiliza para identificar un proceso software concreto de destino.

Protocolo de transporte

Los protocolos de transporte definen las reglas para transportar paquetes o información de una estación de trabajo a otra. Los protocolos de transporte son un tipo de comunicaciones peer-to-peer (de igual a igual) entre dos sistemas. El protocolo de transporte de NetWare es el Internetwork Packet Exchange (IPX). La independencia de protocolos es una etapa importante en el desarrollo de NetWare, y su alcance es tan importante como la introducción de la independencia del medio. Permitiendo que en una red se utilicen varios protocolos de comunicaciones sobre la misma placa, ofreciendo una mayor flexibilidad a la hora de diseñar una red.

Si un usuario da una orden para NetWare, entra en acción el programa IPX establecerá, mantendrá, y terminará una sesión de comunicación en red.



El shell de NetWare redirige las peticiones al sistema operativo de red al servidor a través del cableado

Protocolo Cliente/Servidor

Los protocolos cliente/servidor definen las reglas que se aplican cuando una estación de trabajo (el cliente) hace una petición al servidor. Los clientes envían peticiones y los servidores las atienden.

Existen protocolos cliente/servidor, como el "Apple Talk Filing Protocol" de Apple para equipos Macintosh, el "Network File System" (NFS) de Sun Microsystems para clientes UNIX. Novell soporta estos y otros protocolos cliente/servidor para las versiones de NetWare 3.1 en adelante, para asegurar que el sistema operativo que se ejecute en

una estación pueda acceder a los servicios de un servidor NetWare. Esta estrategia ofrece a los diseñadores de redes la flexibilidad de poder mantener o incorporar estaciones que utilicen sistemas operativos distintos del DOS.

ARQUITECTURA ABIERTA

Un sistema de computadoras con una arquitectura abierta es aquella que potencia el desarrollo de aplicaciones por parte de terceros. Novell soporta el desarrollo de aplicaciones por parte de otras empresas mediante los sistemas abiertos NetWare.

Plataformas de servidor abiertas

A medida que se incrementa la potencia de procesamiento de un servidor, las empresas de software se interesan cada vez más por utilizar parte de dicha potencia. NetWare permite que las aplicaciones tales como programas de gestión de bases de datos funcionen en el servidor en el que puedan aprovechar mejor los recursos de la red para incrementar el rendimiento.

Generalmente, cuando una estación de trabajo ejecuta una aplicación de un servidor, recoge la aplicación completa y el archivo de datos; a continuación, ejecuta la aplicación sobre su propia memoria. Aunque éste es un aspecto importante de las redes, a menudo resulta ineficiente transferir todo un programa o archivo de datos a la estación de trabajo. Sería más eficiente enviar exclusivamente las partes de la aplicación y el archivo de datos necesarias en ese momento.

Una aplicación distribuida puede dividir su carga de trabajo entre el servidor de archivos y las estaciones de trabajo. El nivel front-end (frontal) de esta configuración es la estación de trabajo, en donde se gestionan las actividades de pantalla y teclado. El nivel back-end (de base) está constituido por el servidor, en donde se realizan las tareas compartidas, como entrada/salida de disco. Al interconectarse la potencia del procesamiento del servidor, puede aumentar también el procesamiento back-end realizado por éste.

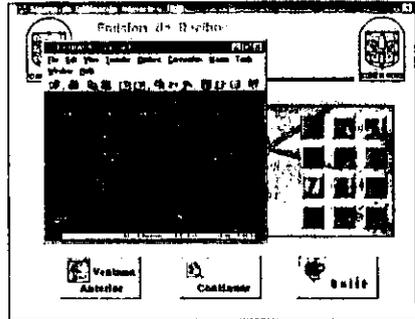
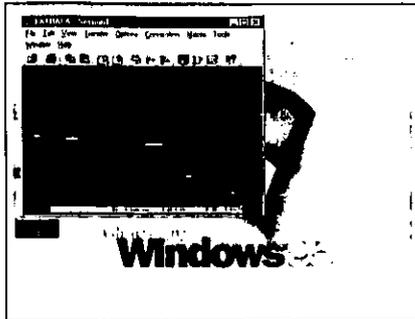
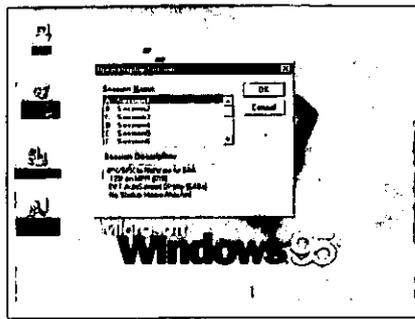
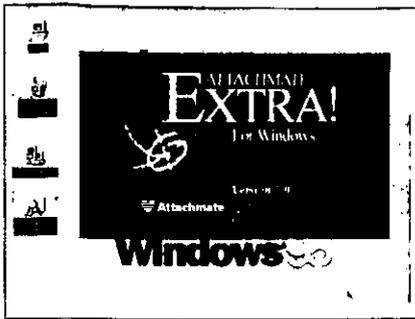
SIMULACION DE UNA TERMINAL

Los PC que actúan como terminales operan en uno de estos modos. Un CUT (Terminal de Unidad de Control) puede tener una sola sesión con el mainframe. En modo DFT (Terminal de Función Distribuida), la terminal 3270 puede tener hasta cinco sesiones coincidentes con el mainframe. IBM tiene incluso un modo llamado MLT (Terminal Lógico Múltiple) que permite múltiples sesiones con las terminales en modo CUT a través de controladores terminales de grupo IBM 3174.

Todos los productos de simulación de terminales y en este caso Extra de Attachmate tienen la posibilidad de grabar y usar macros, conjuntos de teclas de control que están almacenados y listos para responder. Las macros facilitan el arranque de aplicaciones que normalmente necesitan muchos conjuntos de teclas de control para inicializarse. Los programas de simulación de terminal pueden memorizar los keystrokes (Teclas de Control) que utilice y almacenarlos como una macro a la cual puede acceder fácilmente. Extra de Attachmate permite incluso realizar pausas en la transmisión de la macro por la entrada de teclado; de esta forma, puede crear fácilmente macros que harán una pausa en espera de entradas de teclado, y después continúan. Esto puede resultar interesante cuando necesite introducir datos, una clave o cualquier otra información.

Pantallas de simulación

En máquinas DOS, el entorno de carácter obliga los programas de simulación a producir pantallas estériles pero funcionales. Normalmente cuando se ejecute sesiones múltiples, se debe paginar a través de las pantallas llenas que aparecen. Extra de Attachmate (un simulador basado en Windows), suministra múltiples pantallas de sesiones activas.



Esté simulador de terminal 3270, como se muestra en las figuras de arriba, se puede comprimir la ventana mostrada en una sesión 3270 a un tamaño adecuado y además proporcionar una visión práctica y legible. De este modo se puede controlar la actividad de varias sesiones de mainframe hasta el punto que sea necesario y además usar otras aplicaciones locales.

Con Windows, los simuladores 3270 pueden crear imágenes para lanzar aplicaciones de mainframe y mostrar su actividad. Además, los simuladores 3270 permiten a nosotros los programadores usar las funciones internas de Windows para conectar el mainframe y las aplicaciones de Windows. Para realizar comunicación entre la aplicación de predial y el simulador 3270 empleare el protocolo de comunicaciones de Windows, DDE. El cual establece un canal de intercambio dinámico de datos (DDE) entre Visual FoxPro (en este caso el sistema predial) y otra aplicación basada en Microsoft Windows (Extra! for Windows). El manejo del protocolo DDE (dentro de la programación del sistema lo explico a detalle en el capítulo tres).

Gateways

Los gateways permiten conexiones con entornos basados en minicomputadoras y grandes sistemas (a partir de ahora llamados sistemas centrales). A través de gateways cualquier usuario de la red puede acceder al sistema central utilizando el enlace establecido con la red. Los gateways de NetWare son gestionados por los productos de servicios de comunicaciones de NetWare Communications Services, que se adquieren por separado. Estos productos se han diseñado para funcionar en un servidor NetWare, como un módulo cargable NetWare. Los NetWare Communications Services, están diseñados para admitir cualquier comunicación de servicio de acceso red-sistema central, red-red o red remota. A través de los Communications Services los usuarios de las estaciones de trabajo DOS, Windows, OS/DOS y UNIX, pueden acceder a los recursos de los sistemas centrales de la empresa y a redes de gran alcance. El producto posee un diseño modular, consiste en una plataforma a la que se le añaden otros productos. La plataforma básica se denomina NetWare Communication Executive. Este admite el siguiente módulo adicional que fue el comprado por la DGI para utilizarse como gateway; NetWare Services for SAA. Ofrece un gateway con los sistemas centrales IBM.

Los Communications Services de NetWare son una plataforma expandible. Un motor de comunicaciones funciona sobre el servidor con cualquier combinación de módulos cargables de NetWare (NLM = Loadable Modules).

Los Communications Services de NetWare admiten cualquier combinación de adaptadores Ethernet, ARCNET o Token Ring en un mismo servidor. También admite combinaciones con los protocolos de transporte de red IPX/SPX.

TCP/IP o Apple Talk. Cada protocolo se encuentra implementado como un NLM, pudiendo cargarse asignado a un adaptador de red permanente, o ser cargado y asignado dinámicamente cuando sea necesario.

NetWare Services for SAA

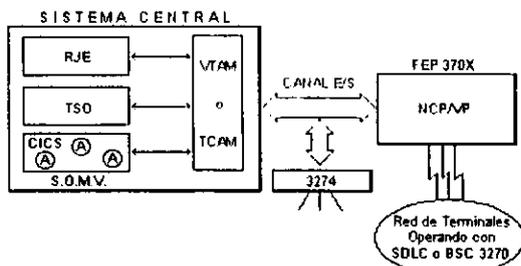
Los Services for SAA de NetWare constituyen un módulo añadido a los NetWare Communications Services (se explica con mayor detalle en el capítulo dos). Ofrecen las siguientes características:

- Ofrecen conectividad red-sistema central IBM, y dan acceso a aplicaciones como Office Vision, NetView, DB2, CICS, TSO y CMS.
- Funciona con los entornos de aplicaciones distribuidas APPC y LU 6.2 de IBM.
- Ofrece conectividad con los AS/400.
- Funciona con las estaciones de trabajo DOS, Windows de Microsoft, Apple de Macintosh, OS/DOS y UNIX.
- Admite hasta mil sesiones de computadora central desde un único servidor.
- Admite las conexiones con computadoras centrales SDLC y Token Ring.
- Funciona con otras utilidades de las Communications Services de NetWare y módulos que permiten la gestión, seguridad y el asignar direcciones.

SNA. SYSTEM NETWORK ARCHITECTURE

SNA (System NetWork Architecture) es un esquema corporativo de IBM orientado al procesamiento distribuido y a la administración de las comunicaciones. Representa un conjunto común de estándares de interconexión, para que una familia de productos de hardware y software se comuniquen. Es también una filosofía de comercialización, que tuvo entre sus objetivos iniciales, la solución a problemas de compatibilidad dentro de la amplia línea de producto de procesamiento y comunicaciones de datos de IBM. Responde a los requerimientos de los clientes de proveer nuevas capacidades. Haciendo un uso efectivo de nuevas tecnologías, SNA tiene como objetivo proveer:

- Un mecanismo de distribución de funciones, que mueva algunas de las tareas del computador central, hacia los periféricos del sistema y equipos remotos.
- Independencia de conexión de forma tal, que diferentes tipos de equipos pueden conectarse al mismo enlace, usando un protocolo común, SDLC.
- Independencia del dispositivo, a efecto de que las aplicaciones sean escritas sin tener en cuenta las características específicas del dispositivo a ser usado.
- Flexibilidad de configuración, para que se pueda cambiar fácilmente la disposición de la red.



Red versión SNA2, se detallan distintas piezas de hardware y software (RJE: Remote Job Entry; TSO: Time Sharing Option, CICS: Customer Information Control System, A: Aplicaciones, SOMV: Sistema Operativo de Memorial Virtual)

CONCEPTOS DEL ESQUEMA CORPORATIVO SNA

SNA define las responsabilidades funcionales de cada componente de red y las reglas de comunicación entre los mismos, proveyendo una estructura de red coherente que pueda acomodar diferentes configuraciones de la red y aplicaciones del usuario.

El objetivo final de una red SNA, como de cualquier otra, es servir a sus usuarios. Por lo tanto, comenzaremos explicando el concepto de usuario.

Servicios a las sesiones en la red

Estos servicios se encuentran ubicados en los puntos de control de los servicios del sistema SSCP, en las unidades lógicas LU y en las unidades físicas PU, estos conceptos los aclaro más adelante.

- *Servicios de configuración*, Responsables de controlar los recursos asociados con la configuración física de la red SNA. Incluyen la activación y desactivación de enlaces entre nodos, Permiten que el operador de la red altere la configuración de la misma.
- *Servicios al operador de la red*, Facilitan la comunicación entre el punto de control SSCP y los operadores de red. Proveen los medios para ejecutar comandos para arrancar y detener la red SNA, activando y desactivando los recursos y llevando bitácoras de errores en los nodos.

Usuario final (EU: End User)

El término usuario final (EU: End User) hace referencia a una terminal de computador, al operador de la misma o a un programa de aplicación. En los tres casos, estamos haciendo referencia a una entidad que interactúa con la red, que usa los servicios de la red con un propósito definido, principalmente, el intercambio eficiente de datos con otro usuario final. El usuario final no forma parte de la red sino que se sirve de ella; es emisor y receptor de datos que fluyen por la red.

Unidades lógicas (LU: Logical Units)

Los usuarios finales no son parte de la red y por lo tanto no están identificados en la red. Sin embargo, debe existir algún punto de conexión o contacto entre la red y los usuarios. A esta puerta de acceso, se le llama Unidad Lógica (LU).

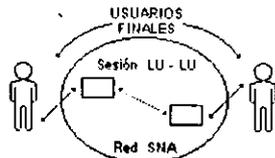
Una LU es una pieza de software (programa o microcódigo) que permite que un usuario se conecte a la red para usar sus servicios y envíe y reciba datos por la red.

Cada usuario está representado por una LU en SNA, aunque un LU puede representar a varios usuarios.

Un LU puede ser un subsistema de aplicación (CICS, IMS) o la programación o lógica asociada con un dispositivo independiente o un subsistema terminal.

Sesiones entre unidades lógicas

Un usuario final accede a la red para efectos de comunicarse con otro a través de la red. Para que ello sea posible, existe un mecanismo de conexión de LUs entre sí, llamado sesión o sesión LU-LU.



Los usuarios finales se conectan con la red a través de las unidades lógicas

Una sesión es una conexión o relación temporal, que permite el intercambio de datos entre las unidades lógicas, o en un sentido más general, entre entidades interlocutoras en la red.

Cuando se activa una sesión LU-LU, la red pone recursos, tales como la capacidad del procesador y la memoria, a disposición de las partes, mientras dure la misma.

En el proceso de establecimiento de una sesión, las unidades lógicas especifican un cierto número de reglas que equivalen a los términos de un contrato entre los usuarios. Se especifica cosas tales como, el formato de los datos, cuántos datos se van a enviar antes de una respuesta, y que acciones se tomarán en caso de error. Las LUs se identifican por un nombre (o nombre de red) que está asociado con una dirección (o dirección de red, NA: Network Address).

Los usuarios trabajan a nivel de nombres lógicos y no de direcciones físicas.

- *Servicios de sesión* Ayudan a activar y desactivar sesiones cuando así se solicita. Una de las principales funciones es convertir los nombres de los elementos que inician una sesión en direcciones de red.
- *Servicios de gerencia y mantenimiento* Permiten que un punto de control (SSC/P) ejecute varias pruebas para determinar si un enlace o un nodo han fallado y por que razón. Ayudan además a llevar estadísticas de error ocurridos en los nodos.

Servicios a los usuarios finales (Transaccionales)

Los servicios NAU usados en el intercambio de datos entre usuarios finales provistos por las unidades lógicas (LU), se llaman servicios a usuarios finales. Se dividen en dos categorías: de *presentación* y de *aplicación a aplicación*.

- *Servicios de presentación (PS: Presentación Services)*. Define el puerto de acceso a la red SNA por un usuario final, en términos de:
 - Requerimientos de traducción de códigos y comandos.
 - Formato de pantalla, atributos de vídeo, etc.
 - Comprensión y Compactación de datos.

Se encargan de que los mensajes sean compatibles con las características del usuario final de destino. Las funciones PS pueden ser realizadas en el nodo fuente, destino en una combinación de ambos. Aquellas terminales que sean tontas, tendrán muy limitados servicios de presentación, relegando esta función a la inteligencia de la cual dependen. Las terminales inteligentes en cambio, tendrán un PS más significativo.

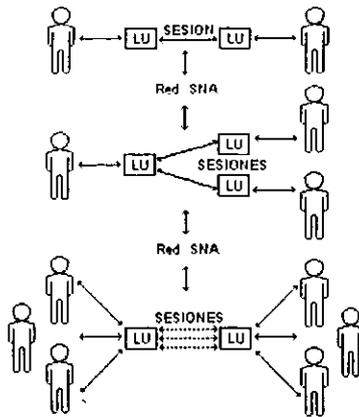
- *Servicios de aplicación a aplicación*. Estos son servicios definidos para aquellas sesiones que vinculan dos sistemas de procesamiento transaccional (tal como CICS/VS). Se acceden desde los programas de aplicación y permiten que éstos, en diferentes nodos, se comuniquen entre sí, sin tener en cuenta otros detalles protocolares de la red. En otros casos, permiten que un programa de aplicación obtenga acceso a una base de datos, sin saber donde se encuentra ésta en la red. Otro servicio, puede ser la forma para que los dos o más programas de aplicación sincronicen sus actividades (por ejemplo la actualización simultánea de registros relacionados, en una base de datos distribuida).

Activación de una sesión

Una sesión entre dos unidades lógicas pueden ser iniciadas por una de las dos LUs involucradas, por una LU diferente, por el operador de la red o por un procedimiento predefinido. El primer paso a realizar para activar una sesión consiste en enviar un requerimiento de activación. Estas se realizan satisfactoriamente, si se cumplen ciertas condiciones:

- Existe un cambio disponible entre las LUs.
- Ambas LUs cumplen las necesidades de los usuarios.
- Hay una autorización para la conexión.

La red puede proveer diferentes niveles de servicios a las sesiones dependiendo de los requerimientos y las opciones (velocidades controles de seguridad).



Relaciones entre los usuarios finales y las unidades lógicas. Un LU puede comunicarse concurrentemente con múltiples LUs. Puede haber múltiples sesiones concurrentes en LUs, donde un LU representa varios usuarios finales.

Desactivación de una sesión

Una sesión LU-LU se desactiva a requerimientos de una de las partes involucradas, o por la causa de un evento ajeno a la sesión, tal como una falla. Siempre se trata que, de ser posible, la transferencia de datos sea completada antes de la culminación de la sesión.

Flujo de datos en sesiones LU-LU

La corriente de datos transmitidos en una sesión LU-LU puede viajar entre:

- Un programa y una terminal, por ejemplo, en una consulta de una base de datos.
- Dos programas, que residen, por ejemplo, en dos sistemas centrales remotos.
- Dos terminales, como en el caso de conmutación de mensajes en una cinta.

Tipos de unidades lógicas

El tipo de unidades lógicas define un subconjunto de protocolos de capas y opciones SNA, soportadas por programas de aplicación durante una sesión. A continuación veremos mediante esquemas, cuáles son las características de los tipos de LUs.

LUT0 (Unidad lógica tipo cero)

La unidad lógica tipo cero se llama de extremo abierto. Es definida por el producto que se usa. Vincula dos programas entre sí.

Los siguientes productos de IBM se aplican como ejemplo: 3600 (finanzas), 3630 (manufactura) y 3650/60 (minoristas).

LUT1 (Unidad lógica tipo uno)

Se refiere al flujo de datos entre una terminal y un programa. Ejemplos de productos son: Impresoras 3270, Terminales RJE.

LUT2 (Unidad lógica tipo dos)

Vincula un programa con una terminal, pero esta vez usando una corriente de datos 3270. Ejemplo de productos son: CRT 3270 y cualquier activación interactiva.

LUT 3 (Unidad lógica tipo tres)

Hace referencia al flujo de datos entre un programa y una terminal, pero este caso, se trata de una impresora simple. Ejemplo de producto: impresoras 3270.

LUT 4 (Unidad lógica tipo cuatro)

La sesión LU-LU tipo 4, define un flujo de datos entre dos terminales o un programa y una terminal, en lo que se llama una corriente de datos SNA.

En este caso no existe una relación primaria/secundaria. Tampoco se requiere un elemento SSCP, del cual hablaremos más adelante. Es posible definir destinos alternativos en este caso. Los productos destinados a la autorización de oficinas son un ejemplo de LUT4.

LUT 6 (Unidad lógica tipo seis)

En este tipo se referencia el flujo de datos (complejo) entre programas. Se refieren a la comunicación entre sistemas usando CICS o IMS.

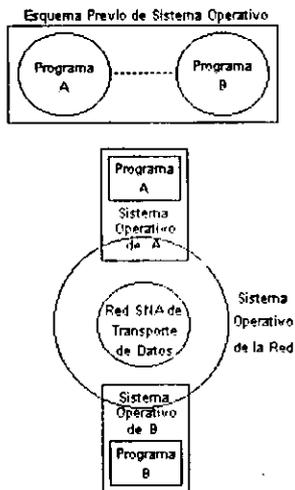
LUT 6.2 (Alias LU "C")

A la LUT 6.2 se le conoce como *Unidad Lógica Tipo C* diseñada para comunicaciones avanzadas entre programas de aplicación.

Algunos de los beneficios de esta LU son: la estandarización, las comunicaciones distribuidas y el sistema operativo distribuidos. La idea es enfocar a SNA desde subconjuntos incompatibles (LUs) hacia subconjuntos convergentes de un único tipo de LU (6.2).

Se busca en los productos LUT 6.2, que todos se comuniquen con todos y que las características avanzadas de los LUT 6.2 pueden ser usadas por cualquier par que soporte funciones avanzadas.

Junto con esta LUT 6.2 surge el concepto de *Sistema Operativo de la Red*.

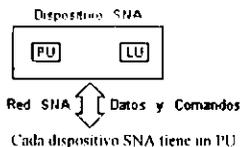


Concepto de Sistema Operativo de la Red Sistema Operativo de la UCP

Unidades físicas (PU: Physical Units)

La unidad física representa las propiedades físicas que tiene el producto respecto de la red. Una unidad física no es un dispositivo físico, sino que es un conjunto de componentes SNA que provee servicios usados para controlar enlaces. terminales tiene una unidad física que es en la red, la representación de ese dispositivo.

La PU administra los recursos físicos de un nodo, activando y desactivando a la propia máquina y a cada enlace proveyendo acceso a otros nodos SNA.



Tipos de unidades físicas

El tipo de PU define la clase de nodos que representa en la red. Un tipo de PU determina el rol de ese nodo dentro de la red, su tamaño y capacidades, tipos de encabezamientos que se usarán en la transmisión. Existen cuatro tipos de unidades físicas (1, 2, 4, y 5) en SNA actualmente.

PUT5

Una unidad física tipo 5 representa un nodo central (Host) conteniendo una PU, una LU y un SSCP. Como ejemplo de producto podemos mencionar: un IBM 3031 con MVS y VTAM o un 4331 con SSC/VTAME.

PUT4

Hace referencia a un nodo que contiene software de control de encaminamiento (PC), una PU y LUs opcionalmente. En general, un FEP va a contener una PUT4. Por ejemplo, un 3705 corriendo software ACF/NCP/VS.

El PUT3 no ha sido definido, por lo cual seguimos con el PUT2.

PUT2

Se refiere a un nodo final con funciones de ruteo limitadas, que contienen una PU y LUs. Generalmente es un dispositivo que controla un grupo de terminales. Ejemplos son: un IBM 3274, 3600, 36x0, s/34/38, 4700.

PUT1

Los dispositivos de dirección única contienen una PUT y opcionalmente una LU. Son dispositivos simples, de bajo costo o dispositivos pre-SNA. Por ejemplo, un 3271 (operando en BSC o SDLC) o un 2741 asincrónico.

EQUIPO SNA INSTALADO

En el desarrollo tecnológico en electrónica esta llegando a niveles muy altos de sofisticación, esto a dada pauta a que máquinas, que en la década pasada fueron grandes en cuanto a capacidad y tamaño, hoy en día sean remplazadas por máquinas de mayor capacidad de procesamiento y en las que se ha reducido significativamente su tamaño, un vivo ejemplo de esto son los procesadores 3031 y 4381.

Otro componente que podríamos mencionar y que juega un papel muy importante en el procesamiento de datos, son las unidades de almacenamiento de acceso directo (discos), cuyo controlador ya incluye una memoria de alta velocidad (cache) y no se diga de los mismos discos, que actualmente ya se fabrican con una densidad de grabación que alcanza una capacidad de almacenamiento de hasta 5000 mega bytes.

Como los ejemplos mencionados, existe toda una gama de dispositivos que han invadido el mercado de la computación, y que sería un poco difícil describir.

El objetivo de este tema es comentar el equipo SNA instalado dentro de la DGI, así como las funciones y características sobresalientes del equipo con que se cuenta actualmente.

Cuadro demostrativo del equipo

Máquina	Modelo	Marca	Cantidad	Descripción
4381	M02	IBM	2	Unidad Central de proceso (CPU)
3031	002	IBM	2	Unidad Central de proceso (CPU)
3880	B13	IBM	2	U. De control de DASD
3880	D23	IBM	2	U. De control de DASD

3880	AA4	IBM	6	U. De control de DASD
3880	B04	IBM	6	U. De control de DASD
3880	AD4	IBM	2	U. De control de DASD
3880	BE4	IBM	2	U. De control de DASD
4305	64	STCDM	2	Disco de estado solido
4800		STCDM	1	Acelerador de cintas magnéticas
3800	004	STCDM	3	U. de CTRL. de U. de cintas
3670		STCDM	12	U. de cintas magnéticas
3800	001	IBM	3	Subsistema de impresión de rayo láser
1500	001	STCDM	4	U. de impresión por impacto
3289		IBM		U. de impresión
3268	002	IBM		U. de impresión
3725	001	IBM	2	U. de CTRL. De comunicaciones
3274	41*	IBM	6	U. de CTRL. De terminales
3274	1B	IBM	2	U. de CTRL. De terminales
3274	31C	IBM		U. de CTRL. De terminales
3274	41D	IBM		U. de CTRL. De terminales
3274	51C	IBM	8	U. de CTRL. de terminales
3274	61C	IBM		U. de CTRL. De terminales
3276		IBM		U. de CTRL. De terminales
3036	001	IBM	2	Video para consola maestra
3279	02C	IBM	7	Video para consola maestra
3278	02*	IBM	7	Video para consola maestra
3279	02C	IBM	9	Estación de despliegue
3278	002	IBM	84	Estación de despliegue
3178	002	IBM	165	Estación de despliegue
3814	A4	IBM	1	U. de conmutación
3540	B01	IBM	2	U. lectora de disquetes

Unidad central de proceso IBM 4381

De la familia de los procesadores 43XX, este procesador es considerado como de capacidad media. Diseñado como procesador de propósitos generales, es decir para aplicaciones administrativas, científicas, ingenieriles como centro de información.

Como cualquier otro procesador, también consta de una unidad de procesamiento de instrucciones: un sistema de memoria que consta de memoria real, una cache y un swap buffer, un subsistema de canales, y un subsistema de servicio de mantenimiento que incluye un microprocesador de 16 bits, siendo independiente del sistema.

El bus de la 4381 consiste de dos grosos, uno que comunica al CPU con los canales y es de 8 bytes en paralelo, mientras que, existe otro que comunica con la memoria real y la cache y es de 16 bytes.

CARACTERÍSTICAS GENERALES		
Ciclo de máquina	68	N S
Memoria principal	16	M Bytes
Cache	32	K Bytes
Número de canales	12	
1 byte multiplexor	40-75	K B/S
4 Block multiplexor data streaming	3	M B/S
3 Block multiplexor	2	M B/S
4 Block multiplexor	1	M B/S
Bus	8	Bytes

Traducción dinámica de direcciones de memoria (DAT) por hardware

Unidad central de proceso IBM 303

Perteneciendo a la familia de los 30XX, la 3031 es un procesador de propósito general, ya sea para aplicaciones comerciales o científicas.

El procesador 3031 consiste de las siguientes partes: procesador, consola de monitoreo y una unidad de energía.

El procesador a su vez consta de las siguientes partes: un procesador de instrucciones, memoria la cual está dividida en dos niveles: memoria real y una cache, un grupo de canales controlados por un procesador microprogramado y que además incluye su propia memoria.

Esta máquina puede operar bajo los siguientes sistemas operativos: DOS/VS, DOS/VSE, OS/VSL, MVS/SP, VM/370. Teniendo implementada como característica estándar funciones del supervisor de vs1 en el hardware.

CARACTERISTICAS GENERALES		
Ciclo de máquina	115	NS
Memoria principal	6 y 8	M Bytes
High speed buffer (cache)	32	K Bytes
No. de canales	6	
1 byte multiplexor	40-75	K B/S
5 Block multiplexor	1.5	M B/S
2 Canales data streaming (opcional)	3	M B/S

Traducción dinámica direcciones de memoria (DAT) por hardware

Controlador de unidades de almacenamiento directo IBM 3880

La unidad de control IBM 3880 es una interfase entre el procesador y disco IBM 3380, formando entre la unidad de control y discos 3380, el subsistema de acceso y almacenamiento de la información. Un 3880 consiste de los siguientes componentes:

- Subsystem storage.
- Cache storage directors (CSD).

A su vez el subsystem storage contiene un directorio y una memoria cache. El cache contiene copias de datos de los discos 3380 que son dinámicamente manejados de acuerdo al algoritmo que mantiene las pistas recientemente referenciados de datos accesibles en el cache, con el objeto de incrementar la cantidad de acceso que puedan ser resueltos en el cache y así evitar retardos mecánicos. Para preservar la integridad de los datos, la operación de escritura se realiza tanto en el disco como en el cache. La transferencia de datos hacia y desde el cache es manejada por el microcódigo del director de memoria cache (CSD), siendo así transparente al usuario.

CARACTERISTICAS GENERALES	
Conexión al CPU	1 canal por 'CSD'
Director de memoria cache	2 por unidad
Conexión a discos	2 trayectorias independientes
Modelo B 13	AA4, A04, B04
Modelo D 23	AD4, BE4, AE4, BE4
Dispositivo por trayectoria	4 unidades de disco 3380
Capacidad de la memoria cache	
Modelo B 13	4 M bytes
Modelo D 23	8 M bytes
Velocidad de transferencia al canal	3 M bytes/seg.
Formato de datos soportados	Bloque fijo (512 bytes)
Corrección de errores	1, 2 y 3 bits

Unidad de almacenamiento IBM 3380

Los discos IBM son dispositivos de almacenamiento directo, estos discos se cuenta con 4 modelos, los AA4, AD4 y BE4.

Los AA4 y BE4 cuentan con la circuitería capaz de controlar a las demás unidades que forman una cadena, por lo que se les llama cabeza de string, mientras que los BO4 y BE4 sólo cumplen con la tarea de almacenar información.

Cada unidad de disco viene ensamblada con 2 discos que se encuentran permanentemente montados, y cuyo ensamblaje puede sólo ser removido por un ingeniero de servicios. Cada disco contiene 2 mecanismos de acceso, un mecanismo atiende a una mitad del disco y el otro mecanismo sirve a la otra mitad.

CARACTERÍSTICAS GENERALES				
Discos ensamblados por unidad	2			
Mecanismos de acceso por disco	2			
Cabeza por mecanismo de acceso	15		Datos y uno de servicio	
Cilindros por mecanismo de acceso	AA4, B04, AD4		BE4	
Datos	885		1770	
Alternos	1		2	
Ingeniería de servicios	1		1	
Pistas por cilindros	15		15	
Pistas por mecanismo de acceso				
Datos	13275		26550 Bytes	
Alternos	15		30 Bytes	
Capacidad máxima de datos				
Por pista	47476		Bytes	
Por cilindro	712140		Bytes	
Por mecanismo de acceso	630.2		1260 M Bytes	
Por unidad	2.52		5.04 G Bytes	
Tiempo de búsqueda	AA4, B04	AD4	BE4	
Mínimo	3.0	3.0	3.0 M Segundos	
Promedio	16.0	15.0	17.0 M Segundos	
Máximo	30.0	28.0	31.0 M Segundos	
Tiempo de rotación			8.3 M Segundos	
Velocidad de transferencia de datos			3.0 M B/S	

Disco de estado sólido STC 4305

El 4305 es un dispositivo con memoria volátil (RAM), esto es, la pérdida de energía eléctrica causa la pérdida de datos almacenados en el dispositivo, diseñado para ser dispositivo de paginación virtual aunque también puede ser utilizado para almacenar información que requiere transferencias de alta velocidad de E/S.

El sistema de almacenamiento 4305 consiste de las siguientes partes:

- Unidad de control STC 4000, el cual consiste de 2 controladores de memoria y su fuente de poder, una unidad para floppy disk, un panel de operador, y conectores de canal de F/S compatibles con IBM.
- Unidad de memoria STC 4305 consiste de módulos de memoria y su fuente de poder, interfase para la unidad STC 4000, y un subsistema de enfriamiento. Cada módulo consiste de 2 arreglos de memoria y un módulo de control, el cual direcciona y sincroniza al arreglo de memoria.

CARACTERÍSTICAS GENERALES		
Capacidad de memoria	11,258,880	bytes/módulo
Tiempo de rotación	0.0	Mili segundos
Bytes por pista	14,660	
Tiempo de acceso	0.6	Mili segundos
Velocidad de transferencia	1.5 o 3.0	M bytes/seg.
Capacidad en slots para MVS	2470	Slots/módulo
Cilindros por módulo	96	
1 Arreglo	12	Tarjetas con chips de memoria RAM de 16 K
12 Tarjetas	6	M bytes

Controlador de cintas STC 3800

La unidad de control STC 3800 cumple la función de interfase entre la unidad de cinta y el canal. El canal tiene control de todas las operaciones de E/S ya que es el controlador no tiene interfase con el CPU, mientras que el

controlador: controla y monitorea las operaciones ejecutadas por la unidad de cintas (CU). Algunas de las funciones del controlador son:

- Control de movimiento de la cinta.
- Monitoreo de errores y status.
- Selecciona unidad de cintas.
- Corrige errores durante operaciones de lectura.
- Controla la interfase controlador unidad de cinta.
- Controla la interfase de E/S.
- Almacena la interferencia de datos entre el canal y cu.

CARACTERÍSTICAS GENERALES	
Conexión a CPU	2 canales
U. de cintas soportadas	8
Velocidad de transferencia	1250 K bytes
Métodos de grabación	6250 BPI (GCR)
	1600 BPI (PE)
Interfase	Radial
Recuperación de errores	Automática

Subsistema de impresión de tecnología láser 3800

El subsistema de impresión 3800, es una impresora de alta velocidad que imprime sin impacto y utiliza una técnica electrofotográfica con un láser de bajo poder para que imprima el papel.

Los datos a imprimir son transmitidos de líneas en líneas del procesador a la 3800 y son almacenados en un buffer de paginas interno. A medida que se forma una página, esta es expuesta al rayo modulado de un cilindro rotatorio creando así una imagen latente de la pagina a imprimir. La imagen es cubierta con polvo magnético (toner) y en la estación de transferencia del cilindro al papel, el papel es pasa a través del fijador (fuser) el cual fija la imagen en el papel, mientras tanto, la superficie del cilindro se limpia y reacondiciona para exposiciones subsecuentes.

Si hay suficientes datos en el buffer para imprimir otra página, la impresión continua sin que se detenga el papel. Pueden imprimirse formas con datos fijos, exponiendo la imagen de un negativo de un patrón de formas (overlay) sobre el cilindro. A continuación de muestran las características generales:

- Velocidad de impresión de hasta 20,000 líneas por minuto.
- Electrónica de control con buffer de pagina de 52 KB.
- Proceso electrofotográfico.
- Almacenamiento de generación de caracteres de 128 símbolos.
- 20 Conjuntos de caracteres: incluyendo 6 conjuntos de subrayado.
- Capacidad para imprimir a 10, 12 y 15 caracteres por pulgada (pitch) separados o entremezclados.
- Espaciado vertical de 6, 8, 12 líneas por pulgada.
- Uso de caracteres gráficos o diseñados por el usuario.
- Utilizando un negativo previamente preparado para imprimir formas (overlay), logotipos o diseñar mientras se imprime la información.
- Modificación de copias que permite la impresión de información predefinida de copias especificadas de todas las paginas de un conjunto de datos.
- Marcado de separación de JOB o copias.

Impresora de impacto STC 1500

La impresora STC 1500 es una impresora de impacto de bajo costo la cual contiene propia fuente de poder y electrónica de control. Dos microprocesadores Intel 8085 son usados como parte de la electrónica de control para recibir, interpretar y ejecutar comandos desde el CPU.

Es una impresora de impacto en la cual los martillos golpean las formas y las cintas contra los caracteres de una banda de impresión en constante movimiento. Conforme la banda de impresión se mueve horizontalmente, a una velocidad constante, los caracteres pasan frente a una hilera de martillos de impresión oprimiendo sucesivamente cada carácter presentando en la banda para cada martillo. La unidad de control, utilizando señales de tiempo de la banda rotatoria, identifica cada carácter antes de que este se le alinee con el martillo para determinar si este es el carácter que se

va a imprimir en esa posición. Conforme un carácter se alinea en la posición donde este debe imprimir, la unidad de control envía una señal al martillo de impresión apropiada.

CARACTERÍSTICAS GENERALES		
Velocidad de impresión	1500	LPM
Densidad de línea	6	LPI
	8	LPI
Longitud de la línea	132	caracteres
Control de avance vertical (FCB)	Si	
Cinta de naylon de		
Largo	20 A 25	yardas
Ancho	15	pulgadas

Banda de impresión intercambiable (UCS):

Comercial	9 arreglos de 48 caracteres
Científica	9 arreglos de 48 caracteres
OCR A	9 arreglos de 48 caracteres
PLI 108	4 arreglos de 108 caracteres
ASCII	4 arreglos de 108 caracteres
TEST	5 arreglos de 87 caracteres

Tamaño del papel:

Longitud	3 24	yardas
Ancho	3 1875	pulgadas

Estación de despliegue

Conocida como terminal, es un dispositivo de E/S donde el dispositivo de entrada es el teclado y el dispositivo de salida es la pantalla, video o TRC (tubo de rayos catódicos).

Los datos desplegados en la pantalla son almacenados en forma de códigos en un buffer; el buffer contiene tantas localidades como posiciones en la pantalla. El dato puede ser cargado desde el CPU por el programa de aplicación (CICS ; ROSCOE ; ETC..) o desde el teclado a la pantalla. A continuación se detallan las características generales:

Terminal 3178.

- 24 Líneas de despliegue visual de 90 caracteres cada una.
- Angulo de rotación de 180 grados.

Terminal 3279 y 3278.

- 24 Líneas de despliegue visual de 80 caracteres cada una.
- Teclado especial para funcionar como principal del sistema.
- Monocromático.
- Facilidades de pruebas de barrido.
- Control de brillantez.
- Teclado tipo "date entry".
- Alarma audible.
- Conexión via cable coaxial.
- Códigos manejables ASCII, EBCDIC.
- Colores base blanco, rojo, azul y verde.

CONECTIVIDAD SNA DE IBM

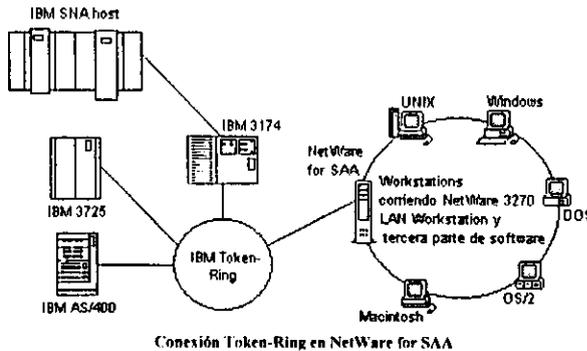
A través del uso de productos Novell es posible establecer conectividad con soluciones que soportan la computación en red. En el ambiente NetWare se puede hablar de tres grupos básicos de conectividad:

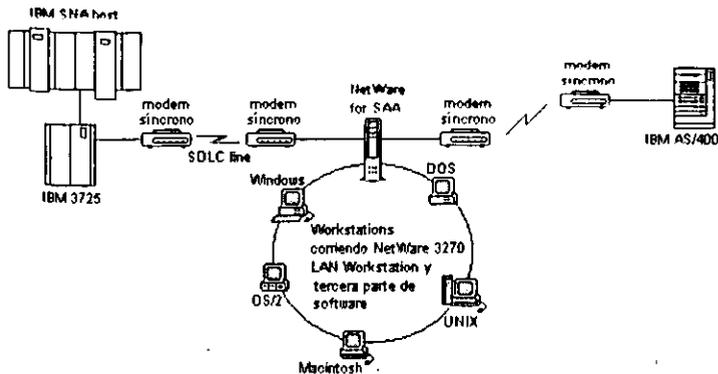
- Conectividad IBM SNA.
- Conectividad PC a HOST.
- Conectividad de área amplia.

En los tres casos la conexión debe ser simple, rápida y eficiente, pero manteniendo un alto grado de funcionalidad y confiabilidad.

En esta sección se revisara la conectividad a IBM SNA. El objetivo de esta es proveer al usuario una conexión desde su estación de trabajo en la red hacia el host (equipo IBM SNA mainframe o similar) permitiendo el trabajo con secciones en el host emulando una terminal nativa de éste.

Para poder realizar la conexión es necesario contar con varios elementos: NetWare for SAA en el servidor, NetWare 3270 for Dos o for Windows en la estación de trabajo y netWare SNA Gateway para establecer la comunicación entre el servidor y el host.



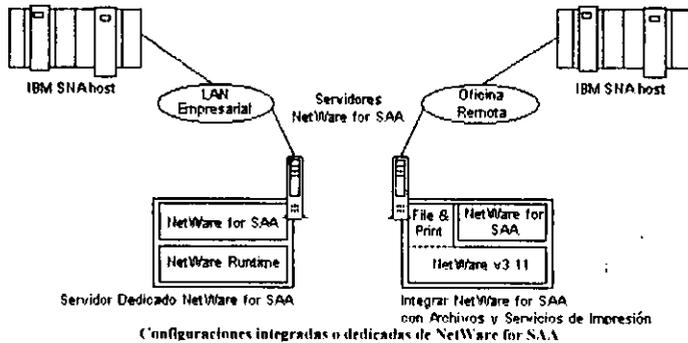


Conexión SDLC en NetWare for SAA

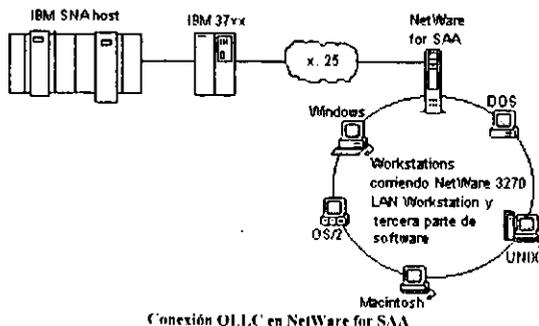
NETWARE PARA SAA

La conectividad entre la red y un equipo IBM SNA se logra a través del NetWare for SAA. Utilizándolo se conectan los mainframes a las redes manteniendo la seguridad, desempeño, servicios de nombre y administración integrados al NetWare. Permite el acceso de estaciones DOS, Windows y UNIX, además soporta hasta 506 sesiones de impresión y despliegue en múltiples anfitriones (host). Permite el acceso transparente a sesiones de AS/400.

NetWare para SAA es implementado en un NLM (NetWare Loadable Module). Este NLM es cargado en el servidor NetWare 3.12 que funciona como servidor de archivos o como un servidor dedicado exclusivamente a las comunicaciones.



Configuraciones integradas o dedicadas de NetWare for SAA

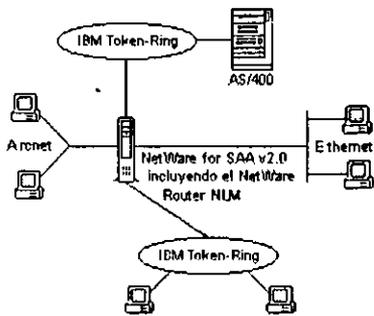


Conexión QLLC en NetWare for SAA

NetWare for SAA soporta conexiones Token-Ring, SDLC y QLLC/X.25 al host. Las conexiones Token-Ring se pueden realizar mediante los controladores 317x, procesadores 37x5 con controladores Token-Ring en los sistemas 9370 y AS/400. Las conexiones SDL utilizan modems asincronos QLLC pueden hacerse a través de una red X.25 utilizando NetWare Packet-Switching Interface (NPSI).

Acceso completo de aplicación AS/400

La integración con IBM AS/400 es completa a través de cliente PC sin importar la topología. Utilizando el PC support software, los clientes de NetWare accesan aplicaciones AS/400 desde cualquier lugar de la red.



Cientes NetWare corriendo PC Support. Acceso para aplicaciones IBM AS/400 a través de NetWare for SAA

Soporte a múltiples conexiones a host

NetWare for SAA soporta hasta 2 PUs (conexiones a host) por servidor. Hasta 254 sesiones se puede configurar por cada PU. Las sesiones host pueden ser cualquier combinación de pantalla, impresión o comunicaciones. NetWare for SAA emula cualquiera de los tipos, PU2.1, cuando accesa las aplicaciones del host.

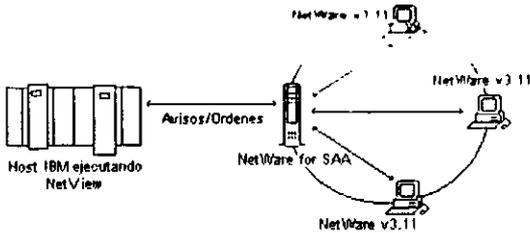
Soporte a múltiples máquinas de escritorio

NetWare for SAA soporta plataformas: Dos, Windows, Macintosh, OS/2 y UNIX. Los usuarios pueden accesar los servicios NetWare for SAA utilizado el NetWare 3270 LAN Workstation for DOS, NetWare 3270 LAN Workstation for Macintosh de Novell. Existen tambien otros emuladores.

Manejo de red mejorado

NetWare for SAA incluye una interfase a NetView management system (Sistema de Administración NetView) de IBM para que los supervisores de red y/o el departamento de sistemas pueden monitorear y controlar los servidores desde el host. Implementando como NetWare Entry Point, NetWare for SAA genera mensajes de alerta sobre errores detectados en un archivo o impresión de NetWare, además de otros en las capas liga de datos y protocolos.

NetWare for SAA nos permite enviar comandos a las servidores de NetWare a través de la consola que corre el NetView en el host. Los comandos que se pueden utilizar incluyen los que cargan y descargan NLMs, los de manejo de directorios y volúmenes, además de recoger mensajes de alerta. Múltiples servidores pueden recibir comandos cuando comparten el mismo host a través de la misma liga facilitando enormemente el trabajo de administración.



Un servidor NetWare for SAA puede funcionar como colector de alertas para el NetWare

Ambiente abierto

NetWare for SAA permite trabajar con el host en forma transparente. Los usuarios no están limitados a acceder los servidores con los emuladores para 3270. Pueden hacer uso de las aplicaciones interconstruidas en el NetWare for SAA que incluye: bases de datos, correo electrónico y de distribución de software del host hacia la red.

Las APLs (Application Programming Interfaces; Interfases de Programación de Aplicaciones) diseñadas para NetWare for SAA proveen un ambiente de desarrollo se puedan montar servicios y aplicaciones, bases de datos, correo electrónico, de una manera muy sencilla.

UTILERIAS

Status

El CSSTATUS provee la información en tiempo real de las sesiones del servidor, identificación de las ligas y de las estaciones de trabajo además, de proveer con sesiones de control de éstas.

Seguimiento para el diagnóstico

NetWare for SAA incluye una utilería que nos permite hacer diagnósticos. Con ella el supervisor puede rastrear las peticiones que se hacen a las ligas. El reporte se puede escribir en disco en ASCII para poder revisarlo posteriormente.

Configuración por menú

A través de la utilería CSCON se puede configurar, ligas, y los accesos a grupos y usuarios del NetWare for SAA desde cualquier lugar de la red.

Auditoría

Esta utilería nos permite conocer el desenvolvimiento histórico de un usuario incluyendo su uso de las conexiones al host y que aplicaciones utilizó. Los datos auditados se pueden extraer en ASCII para su control.

Ventaja del NetWare Name Service

NetWare for SAA extiende el uso de NetWare Name Service para direccionar los recursos del servidor y sus ligas, por Servicios de Nombre. Utilizando un Nombre el usuario accesa los recursos de comunicaciones sin saber la ruta que tomó la información.

Seguridad específica para comunicaciones

Además de la seguridad intrínseca al NetWare los supervisores pueden implementar seguridad específica a través de los ACLs (Access Control Lists; Lista de Control para el Acceso). Los ACLs son listas que permiten saber que usuarios o grupos están autorizados para utilizar los recursos de comunicaciones como son conexiones al host. Esta seguridad del NetWare for SAA está integrada al bindery de NetWare de tal manera que los archivos de seguridad son comunes.

Implementación

Antes de que el servidor con el NetWare for SAA sea conectado al mainframe, el VTAM (Virtual Telecommunications Access Method: Método de Acceso para Telecomunicaciones Virtuales) en el host, que controla la red del host, debe estar configurada para reconocer a el servidor y las sesiones con el NetWare for SAA. De la misma manera la información de configuración del host debe ser registrada en la utilería de configuración del NetWare for SAA.

El administrador de la red y los del mainframe o AS/400 deben trabajar en conjunto para intercambiar la información necesaria para la configuración. Para facilitar el trabajo, se incluye en el producto hojas de trabajo con la información que se requiere.

Requerimientos de hardware

- Procesador: 386 ó 486
- Memoria: Mínimo 6 Mb para 16 sesiones
8 Mb para 64 sesiones
12 Mb para 254 sesiones

El servidor debe incluir tarjeta de red y el host una tarjeta para comunicaciones.

Software necesario

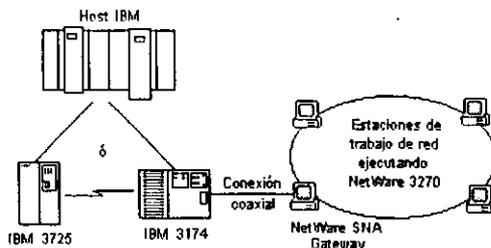
Además de NetWare for SAA, los usuarios que requieren acceso a los hosts de IBM necesitan el NetWare 3270 LAN Workstation for DOS, NetWare 3270 LAN Workstation for Windows, NetWare 3270 LAN Workstation for Macintosh o algún otro software de emulación corriendo en sus terminales. Todos estos productos permiten emular las terminales 3270 y transferir archivos de y hacia el host. Los productos para Windows y DOS permiten también a las estaciones emular las impresoras 3287 y correr programas de transacción del tipo LU6.2.*

NETWARE SNA GATEWAY

Este software convierte una microcomputadora en un servidor SNA gateway. Dependiendo del tipo de conexión al host el NetWare SNA Gateway permite hasta 97 microcomputadoras comunicarse con un mainframe SNA o compatible. El NetWare SNA Gateway corre sobre redes NetWare SPX o IBM NetBIOS y soportar cuatro diferentes conexiones al host: coaxial, CoaxMux, remota o Token-Ring. Estas opciones permiten al supervisor y a los usuarios seleccionar el tipo de hardware de comunicaciones que más les convenga según su presupuesto o necesidades.

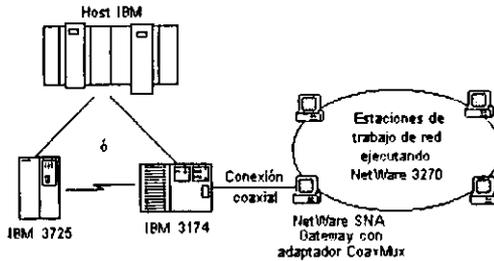
Coaxial

La conexión coaxial soporta hasta cinco sesiones entre el adaptador en el servidor del gateway y el puerto DFT del SNA en el cluster controlador 3x74.



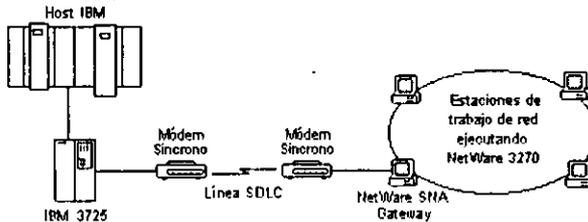
CoaxMux

Esta opción soporta hasta cuarenta sesiones. El adaptador está conectado por cable coaxial al puerto multiplexor 299 en el cluster controlador 3x74. Este tipo de conexión permite al gateway multiplexar 85 líneas de sesión sobre un solo cable coaxial.



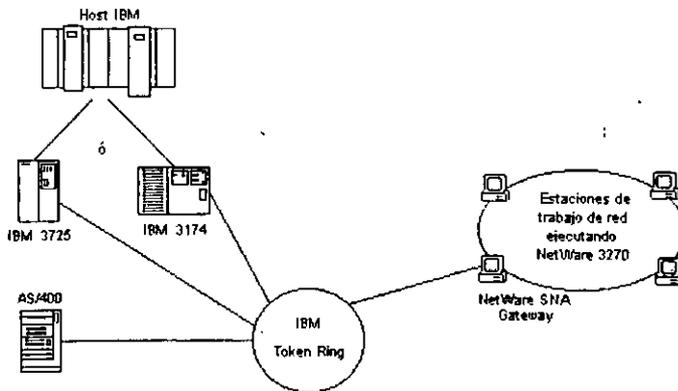
Remoto

Esta opción soporta hasta 128 sesiones en 97 estaciones de trabajo utilizando un módem síncrono conectado al procesador front-end del 37xx a velocidades de 64kbits/s.



Token-Ring

Esta disponible con Token Ring hasta 128 sesiones en 97 estaciones de trabajo. La conexión al host se realiza a través de equipo capaces de utilizar los IBM 3174, 3720, 3725, 3745, 9370 o AS-400.



Características

Incluyen archivos de teclado, plantillas y documentación que facilitan la conexión al AS/400 a través de TokenRing.

La utelería Gateway Status nos provee información actualizada sobre las sesiones efectuadas en: el host, las ligas, identificadores y estaciones de trabajo.

Las herramientas para administración incluyen una utelería para seguimiento interno que nos permite, diagnosticar y hacer autoexámenes para detectar y aislar problemas entre el gateway y las estaciones del trabajo.

El software puede ser configurado para hacer conexiones a múltiples host, permitiéndonos cambiar la conexión cuando la instalación crece. Esta característica también nos provee herramientas de recuperación para después de un desastre.

La instalación es simple debido a que existe una Guía para un Inicio Rápido, documentación y el menú de configuración.

Es posible instalar múltiples gateways en una sola red, permitiendo a las estaciones de trabajo acceder múltiples hosts. El NetWare SNA Gateway nos permite implementar gateways de respaldo como equipo de redundancia en caso de fallas o para repartir la carga de trabajo.

La instalación de Token Ring provee acceso de alta velocidad a los host IBM pero también nos permite conectar redes con otras topologías, como Ethernet, utilizando tarjetas token ring y Ethernet en la máquina que funcionará como gateway.

El gateway puede correr como servidor dedicado o no dedicado. Novell recomienda que sea dedicado.

Implementación

NetWare SNA Gateway permite a una máquina en la red emular un nodo tipo 2, como sería un cluster controlador. Antes de conectar el gateway al maiframe, se debe definir en el VTAM, que corre en el host. El supervisor que instale NetWare SNA Gateway probablemente necesitará ayuda de los operadores de los equipos grandes para la configuración.

Se debe considerar cuidadosamente el tipo de conexión que se realizará. La decisión debe considerar lo que tenga más sentido al medio ambiente: remota, coaxial o TokenRing.

En general, se utiliza la conexión remota para host que no se encuentran en el mismo lugar físico de la red. La conexión coaxial se utiliza localmente, es bastante más eficiente que la remota y cuesta menos que el Token Ring. El Token Ring se utiliza cuando la velocidad es crítica pero su costo es alto.

Requerimientos de hardware

Independientemente de la configuración, se requiere una máquina PC o compatible: Desde 202 hasta 502kb de RAM, dependiendo de la configuración y el número de sesiones que se utilicen.

Conexión coaxial

Este tipo de conexión requiere de un adaptador (tarjeta) coaxial. El adaptador se conecta a un controlador cluster 3x74 o compatible a través del cable coaxial. Con esta configuración se puede operar la estación de trabajo en modo no dedicado, permitiéndonos que la estación tenga otros usos. Esta solución es la más eficiente costo beneficio pero es muy probable que el gateway deje de funcionar al ser reinicializada la estación o cuando ésta se congela.

Conexión Coaxial Mux

Para operar esta solución es necesario adquirir un adaptador CoaxMux. Este se conecta vía cable coaxial al puerto 3299 del SNA 3174 o 3724. Esta solución se debe correr en modo dedicado para mantener su confiabilidad y rendimiento.

Conexión remota

Requiere de una tarjeta para comunicaciones remotas, que conecta en forma sincrónica módems extremos sobre líneas SNA/SDLC a un procesador front-end del IBM 37xx o equivalente., Dependiendo del tipo de tarjeta la conexión al módem se realiza con una RS- 232 o con un cable V.35.

Esta configuración puede operar en modo no dedicado, pero corremos el riesgo de perder la conexión. Dependiendo del tipo de tarjeta, se puede transmitir a velocidades entre 19.2kbit/s hasta 64 kbit/s. Para mejorar su desempeño es importante definir un número realista de conexiones que se necesitarán en el host, utilizar al límite el gateway con 97 estaciones de trabajo y 123 sesiones, disminuye su desempeño.

Token Ring

Para establecer esta conexión es necesario contar con un adaptador para Token Ring. Este adaptador requiere cableado de par telefónico blindado.

Algún host debe estar conectado al Token Ring. Se debe considerar una máquina 386 dedicada cuando la conexión tiene mucho uso. Para mejorar su desempeño es importante definir un número realista de conexiones que se necesitarán en el host, utilizar al límite el gateway con 97 estaciones de trabajo y 123 sesiones, disminuye su desempeño.

Software necesario

Además del SNA Gateway, el usuario necesita el NetWare 3270 LAN Workstation for DOS software. Es necesario utilizar las versiones de DOS 3.1 o superior.

Especificaciones

Número máximo de sesiones al host

Coaxial	5
CoaxMux	40
Remoto	128
Token Ring	128

Número máximo de usuarios

97

Velocidad máxima de las líneas

64 kbits/s en remoto
2.3 Mbits/s en coaxial
4/16 Mbits/s en Token Ring

Protocolos soportados

IEEE 802.2
SPX
NetBios

Sesiones del Host

LU Tipo 2 terminales de despliegue emulando 3270
LU Tipos 1 ó 3 sesiones de impresión 3287
LU Tipo 6.2 sesiones punto a punto

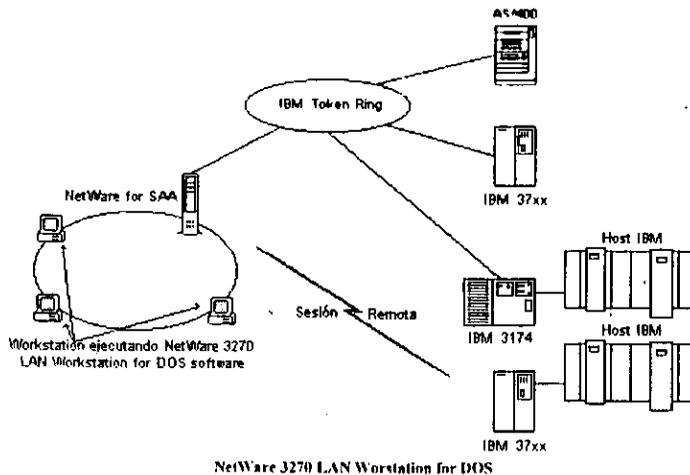
NETWARE SNA GATEWAY ELS

Es una versión recortada del SNA Gateway. Sólo funciona para conexiones remotas o por coaxial. Soporta 5 sesiones al host en coaxial y en forma remota 16 sesiones. El número máximo de usuario es de 16.

NETWARE 3270 LAN WORKSTATION PARA DOS

NetWare 3270 LAN Workstation for Dos es un emulador de terminal y de impresora de IBM utilizado por el NetWare for SAA. Este software de emulación soporta hasta cinco sesiones con el host por estación de trabajo. Las sesiones pueden ser combinación de despliegue, impresión, comunicaciones, sesión DOS, transferencia de archivos y cambio de teclado.

La transferencia de archivos se realiza utilizando las utilerías SEND/RECIVE o TRANSFER, que son incluidas en el software. La utilería SEN/RECIVE baja y sube archivos desde y hacia el host.



Características

- Utilería para su instalación.
- Utilería para la conversión de archivos.
- Una aplicación DOS puede ejecutarse en el fondo, aun cuando tengamos una sesión activa de terminal o impresora.
- El reattach automático restablece la conexión entre la estación de trabajo y el servidor.
- Emula terminales IBM 3270 e impresoras 3287.
- La personalización de la sesión de impresión permite al usuario asignar una impresora a la red o una impresora local, especificar el número de líneas por página y configurar el software para que automáticamente inicie la sesión de impresión en el host cuando sea cargado el software.
- Soporta el redireccionamiento de impresión de pantallas y de trabajos a las impresoras de red.
- La utilización de memoria puede ser reducida al levantar el software en la memoria alta de la estación de trabajo.
- NetWare 3270 LAN Workstation for DOS puede ser dado de baja de la memoria residente sin necesidad de reinicializar la estación, permitiendo al usuario trabajar con otras aplicaciones que necesitan esa memoria.
- Provee diagnóstico interno de problemas.
- El remapeo del teclado nos muestra ayuda en línea, permitiéndonos reasignar las teclas, incluyendo las críticas. Soporta hasta 30 teclados de diferentes lenguajes.
- Encripta passwords en la red.

Implementación

NetWare 3270 LAN Workstation for DOS emula hasta cinco sesiones de despliega (LU2), impresoras (LU16 LU3) y comunicaciones (LU6.2) por estación de trabajo. En el ambiente del mainframe los LUs deben de estar definidos en el VTAM software. Para configurarlo se debe seguir los mismos pasos que para el NetWare for SAA.

Requerimiento de hardware

IBM PC, PS/2 compatible con 2.5 Mb de espacio en disco, tarjeta de red y conexión a la misma.

Software necesario

Es necesario contar con el NetWare for SAA para establecer la estación con el host. En la estación de trabajo es necesario tener DOS 3.3 o superior.

Es necesario por lo menos 135 Kb de memoria por cada sesión en el servidor NetWare for SAA. Más memorias se requiere para soportar sesiones adicionales y que sea funcional.

Especificaciones

- Número máximo de sesiones en la estación de trabajo:
5.
- Protocolos Soportados:
SPX de Novell.
- Sesiones del host soportados:
LU Tipo 2 terminales de despliegue emulando 3270.
LU Tipos 1 ó 3 sesiones de impresión emulando 3270 model 2.
LU Tipo 6.2 sesiones punto a punto.

Instalación

Se inserta en la unidad A el disco denominado NetWare 3270 LAN Workstation for DOS Disk 1.

Estando en la unidad A se tecllea

Install

Al ejecutarse el programa de instalación aparece en pantalla una ventana donde debemos indicar en que directorio y unidad de disco se almacenara la instalación.

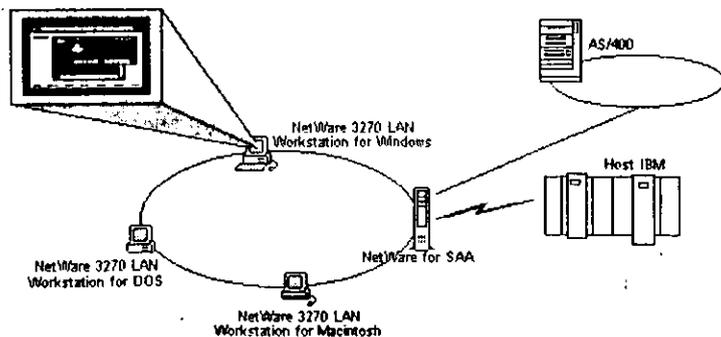
Comienza a ejecutar la instalación y nos pide que insertemos el Disco 2.

NETWARE 3270 LAN WORKSTATION PARA WINDOWS

NetWare 3270 LAN Workstation for Windows es un emulador de terminal y de impresora utilizado por el NetWare for SAA para proveer a los usuarios Windows conectividad al host IBM. Los usuarios de Windows obtienen un acceso a los mainframes SNA y a las minicomputadoras AS/400 de manera muy eficiente, permitiéndoles acceder aplicaciones en el host como son: NetView, DB2, TSO, CMS, CICS, PROFS y Office Visión.

Como cualquier aplicación Windows, NetWare 3270 LAN Workstation for Windows está completamente integrada al ambiente Windows. Los usuarios pueden utilizar la interfase tan amigable de Windows, incluyendo el uso extensivo del ratón, copia y pegado, fuentes, teclado y esquemas de color.

NetWare 3270 LAN Workstation for Windows soporta múltiples sesiones concurrentes de despliegue e impresión del host por cada estación de trabajo. Con un clic del ratón, los usuarios pueden cambiar de aplicaciones Windows a sesiones del host. Si poseen los derechos, los usuarios podrán mantener sesiones con múltiples hosts SNA. Un menú de transferencia de archivo, les permite mover información desde y hacia el host.



NetWare 3270 LAN Workstation for Windows provee a los usuarios de red un poderoso acceso a los host IBM a través de NetWare for SAA

Características

- Interfase gráfica para el usuario utilizando Windows 3.1 o superior.
- Emula terminales IBM 3270 e impresoras 3287.
- Soporta múltiples sesiones concurrentes de despliegue e impresión en múltiples hosts, cada una en su propia ventana.
- Permite a los usuarios personalizar colores y fuentes en la sesión del host.
- Soporta al ratón para reposicionamiento del cursor, copiado y pegado del texto desde y hacia el host.
- Muestra el reposicionamiento del teclado gráficamente.
- Muestra en pantalla un teclado para utilizando el ratón activar teclas críticas (del host).
- Soporta el redireccionamiento del impresión de pantalla e impresión 3287 a las impresoras de red.
- Provee con ayuda sensitiva al contexto.

Implementación

NetWare 3270 LAN Workstation for Windows emula múltiples sesiones concurrentes de despliegue (LA2), impresión (LUI o LU3) y comunicaciones (LU6.2) por estación de trabajo. En el medio ambiente de los LUs deben estar definidos en el VTAM, que corre en el host y que lo controla antes de conectarse a la red. En el ambiente AS/400. El supervisor que instale NetWare 3270 LAN Workstation for Windows probablemente necesitará ayuda de los operadores de los equipos grandes para la configuración.

Requerimientos de hardware

NetWare 3270 LAN Workstation for Windows software requiere un procesador 386 ó 486, por lo menos 2 Mb memoria y uno de los monitores requeridos por Windows.

Software necesario

En el servidor debe estar instalado el NetWare for SAA. En la estación de trabajo debe estar instalado Windows 3.0 o superior.

Especificaciones

- Protocolos de Transporte Soportados:
IPX/SPX de Novell.
- Sesiones del host soportadas:
LU Tipo 2 de despliegue emulando terminales 3270.
LU Tipos 1 ó 3 sesiones de impresoras emulando impresoras 3287 Modelo 2.
LU Tipo 6.2 sesiones punto a punto.

LU (Logical unit; unidad lógica)

Una unidad lógica puede ser una terminal, un programa de emulación de terminal o una aplicación. Una LU puede comunicarse con un SSCP (System Services Control Point; Punto Central de Servicios del Sistema), con aplicaciones del host (en los tipos 0, 2, 3,4 y 7) o con LUs del mismo tipo (Tipo 6.0, 6.1, y 6.2 solamente).

LU Tipo 0

Esta unidad lógica es definida por el usuario. Se utiliza para soportar tipos de terminal y características que no están definidas por IBM. Requiere un código específico escrito por el usuario en el host.

LU Tipo 2

Terminal con teclado y monitor compatible con equipo 3270. Por ejemplo, la IBM 3278.

LU Tipo 3

Impresora compatible con la 3270, pero sin juego de caracteres SNA.

LU Tipo 4

Terminal tipo 5250 con caracteres SNA

LU Tipo 6.0

Unidad lógica de comunicaciones aplicación-a-aplicación del mismo tipo en el mismo host.

LU Tipo 6.1

Unidad lógica de comunicaciones aplicación-a-aplicación entre dos tipos de aplicación en el mismo host en diferentes hosts.

LU Tipo 6.2

Unidad lógica para comunicaciones avanzadas IBM de aplicación-a-aplicación/computadora personal. Se implementa en redes punto-a-punto bajo el esquema SNA.

EMPLEANDO PROTOCOLO DDE CON EXTRA! FOR WINDOWS

Windows no es un sistema operativo, es un shell o interfase que permite acceder con mayor facilidad los comandos del sistema operativo, pero que además incorpora comandos propios al DOS para hacerlo más poderoso y extender sus capacidades. Windows facilita la interfase con el usuario, es decir, interfases gráficas conocidas como GUI (Graphic User Interface). Este tipo de interfaces facilitan en gran medida el uso de las computadoras, y por lo tanto, el consumo de gran variedad de software de aplicaciones y desarrollo.

Windows es un producto que exige hardware para ser ejecutado, para Windows 3.0 en adelante se necesita por lo menos un procesador 286, un mínimo de 1 Mb de memoria RAM, monitor de alta resolución, suficiente espacio disponible para almacenamiento de información; de todo esto se desprende que actualmente es un shell estándar, y se le denomina sistema operativo.

Esto implica además, que las empresas y otros que desarrollan software hayan tenido que reescribir sus aplicaciones para aprovechar la interfase gráfica, el acceso a más memoria, el ambiente multitareas, etc., que ofrece Windows. Microsoft va a la cabeza en número de aplicaciones desarrolladas para Windows (el ambiente Windows fue desarrollado por Microsoft), pero no sólo ellos tienen herramientas de productividad desarrolladas en este ambiente, empresas como Lotus, Borland, Aldus, sólo por mencionar algunas, tienen productos que trabajan bajo este ambiente.

A partir de Windows 3.1, da un nuevo soporte para el desarrollo de aplicaciones en el ambiente multiusuario con tecnologías como la de Cliente-Servidor, el último grito de la moda en el manejo centralizado de bases de datos utilizando PC y una conectividad de red

PARTICULARIDADES DE WINDOWS

Windows se pone en marcha de la misma forma que cualquier aplicación que se ejecuta en modo DOS. Sin embargo, cuando aparece la venta del administrador de programas, Windows se convierte en un sistema operativo. Windows carga el teclado, ratón, video, puertos series y paralelos, memoria y la ejecución de programas; tomando el control del hardware, Windows puede presentar la pantalla en ventanas separadas. Gestionando el hardware Windows puede interpretar toda entrada del usuario, y distribuir dicha entrada como mensaje a las aplicaciones apropiadas; sólo gracias al control total del hardware, puede asegurar que una aplicación puede combinarse con una variedad de dispositivos de salida, creando de este modo una máquina virtual.

Con Windows una aplicación no escribe directamente en la pantalla ni procesa una interrupción hardware, ni genera salida directamente en la impresora, en lugar de esto, la aplicación usa las funciones aplicadas de Windows o espera que se entregue un mensaje apropiado. El desarrollo de aplicaciones Windows debe ahora considerar el procedimiento realizado por la aplicación y el punto de vista del usuario a la misma, a través de Windows. Desde el punto de vista de la aplicación un mensaje es una notificación de que se ha producido un suceso de interés; un suceso que puede o no requerir una acción. Estos puede haberlos iniciado el usuario haciendo clic, moviendo el ratón, modificando el tamaño de una ventana, o realizando una selección de menú: los sucesos señalados también podrían haber sido generados por la aplicación.

La aplicación debe estar ahora totalmente orientada al procesamiento de mensajes. Debe ser capaz de iniciar y mantener la acción adecuada en función del tipo de mensaje recibido, realizar la acción hasta que termine. Las aplicaciones Windows son bastante diferentes a su analogas para DOS. Windows ofrece a una aplicación el acceso a unas 1500 llamadas a funciones, generalmente estas llamadas a función son gestionados por tres módulos principales: KERNEL, GDI (Interfaz de Dispositivos Gráficos) y USER. El módulo KERNEL tiene unos 50 Kb y es el planificador. El GDI es de unos 100 Kb y contiene todas las rutinas necesarias para mostrar y crear gráficos. El módulo USER, de unos 150 Kb se preocupa de las demás necesidades de las aplicaciones.

Formato de los mensajes

El propósito de un mensaje es notificar a su programa que se ha producido un suceso de interés. Técnicamente un mensaje no es de interés para la aplicación, sino para una ventana específica dentro de dicha aplicación. Todos los mensajes están destinados a una ventana. Windows sólo tiene una cola de mensajes de sistema, para cada programa que se ejecuta concurrentemente con Windows dispone de su propia cola de mensajes. Todos los mensajes de la cola deberán ser transferidos eventualmente por el módulo USER a la cola de mensajes de un programa. La cola de mensajes de un programa almacena todos los mensajes destinados a todas las ventanas de mensajes.

Los parámetros siguientes están asociados a todos los mensajes independientes de su tipo:

- Un manipulador de ventana (palabra de 16 bits).
- Un tipo de mensaje (palabra de 16 bits).
- Un parámetro Word (palabra de 16 bits).
- Un parámetro Long (palabra de 32 bits).

El primer parámetro especificado en un mensaje de ventana en el manipulador de la ventana a la que esta dirigido el mensaje en un entorno a programación orientado a objetos, un manipulador es justamente el identificador de un objeto, que para la sintaxis actual es el identificador de una ventana en particular a la que esta dirigido el mensaje. Un manipulador es un número de 16 bits sin signo, frecuentemente, este manipulador hará referencia a un objeto (definido como una estructura de datos) que esta localizado en una parte movable de la memoria. Es importante darse cuenta de que, aunque dicha porción de memoria puede ser movida, el manipulador sigue siendo el mismo. Esto permite a Windows gestionar la memoria de forma más eficiente, haciendo que la reubicación sea invisible para la aplicación.

Debido que puede crear múltiples ventanas usadas en la misma clase de ventanas, una única función de ventanas podría procesar los mensajes de más de una ventana en un único programa. Bajo estas circunstancias, la aplicación podría utilizar el manipulador para determinar qué ventana recibe el mensaje.

El segundo parámetro de un mensaje es el tipo de mensaje. Este es uno de los identificadores específicos en el archivo de cabecera de Windows.h; con Windows cada tipo de mensajes se representa mediante un mnemónico de todos los caracteres, seguido de un carácter subrayado y por último, un descriptor para complementar el mensaje. El tipo de mensaje más común que puede procesar una acción es el mensaje de ventana, WM_CREATE, WM_PAINT, WM_CLOSE, WM_COPY, WM_PASTE, son algunos ejemplos de mensaje de ventana. Otros tipos de mensaje de ventana (BM_), los mensajes de control de edición (EM_) y los mensajes de cuadros de listado (LB_).

Cuando es necesario una aplicación puede crear y declarar sus propios tipos de mensaje, lo que posibilita el uso de tipos de mensaje privados. El uso de los dos últimos parámetros Word y Long es ofrecer información adicional para interpretar el mensaje. El tamaño de Word y Long cambiara en Win32 (Windows NT).

Windows es capaz de realizar multitareas debido a su estructura subyacente de mensajes, todos los mensajes deben ser procesados por Windows. Básicamente, existen cuatro fuentes de mensajes, una aplicación puede recibir un mensaje del usuario, del mismo Windows, del propio programa de aplicación o de otra aplicación.

Los mensajes provenientes del usuario incluyen información sobre el cambio de las teclas pulsadas, movimientos del ratón, coordenadas de las selecciones realizadas con el ratón, las selecciones de los menús, la posición de los cuadrados de desplazamiento en las barras de desplazamiento. Su programa de aplicación dedicara una gran parte del tiempo a procesar mensajes del usuario. Los mensajes originados por el usuario indican que la persona que a ejecutado el programa desea cambiar o ver la aplicación. Windows origina un mensaje a una aplicación siempre que se efectúe un cambio de estado, por ejemplo, cuando el usuario hace un clic en una aplicación para activarla. En este caso Windows informa a la aplicación que se esta abriendo su ventana principal, que su tamaño y posición se están modificando y así sucesivamente. Dependiendo del estado actual de una aplicación los mensajes originados por Windows pueden ser atendidos o ignorados. Los programas de Windows se descomponen en procedimientos específicos, cada uno de los cuales procesa un tipo de mensaje en particular por cada ventana concreta. El origen del mensaje es la propia aplicación.

Para facilitar el uso de tipo de mensaje, Microsoft a desarrollado el *protocolo de mensajes DDE* (Intercambio Dinámico de Datos), que permite que los programas de aplicación pidan o intercambien los datos automáticamente. Un programa en una ventana puede interrogar a un programa en otra ventana utilizando el protocolo de mensajes DDE.

Una aplicación de Windows tendrá un procedimiento para procesar cada tipo de mensaje que sea de interés para alguna de sus ventanas. Distintas ventanas normalmente responderán de modo diferente a mensajes del mismo tipo. Una aplicación puede tener dos ventanas que responden a distinta forma a la pulsación del ratón. La primera ventana puede responder modificando el color del fondo; la segunda ventana puede responder situando un gráfico en una hoja de cálculo. Puesto que el mismo mensaje puede ser interpretado por distinta forma por diferentes ventanas, Windows envía cada uno de los mensajes a una ventana específica dentro de una aplicación. La aplicación no sólo tendrá un procedimiento diferente para gestionar cada tipo de mensaje sino que necesitará un procedimiento para gestionar cada tipo de mensaje para cada ventana. El procedimiento de ventana se utiliza para agrupar todos los procedimientos de tipos de mensaje de una aplicación Windows.

Bucle de mensajes

Un componente fundamental de todas las aplicaciones de Windows es el bucle de procesamiento de mensajes. Todas las aplicaciones contienen procedimientos para crear e inicializar ventanas, seguidas del bucle de procesamiento de mensaje y por último, algún bucle de código necesario. El bucle de procesamiento de mensajes es el responsable de procesar un mensaje enviado por Windows al cuerpo principal del programa. Entonces el programa acusa de recibo del mensaje y solicita a Windows que lo envíe al procesamiento de la ventana apropiado para su procedimiento. Hecho esto, el procesamiento de ventana ejecuta la acción deseada. La cola de mensajes y la prioridad del servicio son dos factores que pueden influir en la secuencia de procesamiento. Los mensajes se pueden enviar desde una de las dos colas posibles ó desde la cola de sistema, a la cola de mensaje de la aplicación. Este modo dual hace posible que Windows pueda ejecutar la pista a todos los mensajes, permitiendo a la vez que cada aplicación siga la pista a los mensajes que sean de su incumbencia. Normalmente, Windows ubica la mayoría de sus mensajes en las colas de las formas que cabría esperar, según el orden "el primero en entrar es el primero en salir" (FIFO). Estas son las denominados mensajes sincrónicos. La mayor parte de la aplicación de Windows utiliza ese tipo de método de servicio. Sin embargo, existen ocasiones en las que Windows mantiene algunos mensajes al final de la cola, evitando de este modo que sean atendidos. Los mensajes de ese tipo se denominan mensajes asincrónicos que altere la secuencia normal de procedimiento de un programa.

Existen tres tipos de mensajes asincrónicos: pintar, temporizar y salir, un mensaje de temporización, hace que una determinada acción surta efecto en un determinado instante. Por lo tanto, un mensaje de temporización hará que los restantes mensajes de la cola sean desplazados del principio de la misma.

Windows también dispone de mensajes asincrónicos que puede enviar a otras aplicaciones. Esta situación es excepcional, en tanto la aplicación que recibe el mensaje no lo coloca en su propia cola. En lugar de esto, el mensaje llama inmediatamente a la función de ventana apropiada de la aplicación receptora, donde se ejecuta dicho mensaje. Windows despacha los mensajes que están pendientes en varias aplicaciones a la vez. Windows resuelve esta cuestión de dos formas. Uno de los métodos de procedimientos de mensajes se denomina prioridad de servicio. Cada vez que Windows carga una aplicación ajusta la prioridad de dicha aplicación a cero. Sin embargo, una vez que se esta ejecutando la aplicación puede ajustar la prioridad entre los valores -15 y 15. En igualdad de condiciones, Windows soluciona cualquier contienda por el servicio enviando mensajes a la aplicación de mayor prioridad.

Una aplicación de comunicación de datos es un ejemplo de programa que podría necesitar elevar su nivel de prioridad. Si su sistema estuviese conectado a una computadora central distante.

Puesto que entrometerse en el nivel de prioridad en una aplicación es algo poco común, Windows debe disponer de otros métodos para despachar los mensajes de aplicación concurrentes que tengan el mismo nivel de prioridad. Además de procesar los mensajes de la cola, siempre que Windows observa que una aplicación determinada tiene una sobrecarga de mensajes no procesados, desecha nuevos mensajes determinados a dicha aplicación mientras continúa despachando restantes para las restantes aplicaciones.

INTERCAMBIO DINÁMICO DE DATOS

El intercambio dinámico de datos (Dynamic Data Exchange, abreviado DDE) es un mecanismo soportado por algunos lenguajes de programación bajo el ambiente Windows, por ejemplo, Visual Basic, Delphi, C++, FoxPro (a partir de la versión 2.5). Estos permiten que una aplicación hable continuamente con otra aplicación desarrollada con ambiente Windows para intercambiarse datos automáticamente. DDE automatiza el intercambio manual entre aplicaciones que se realizan a través del portapapeles.

El DDE es similar a una conversación entre dos personas. La aplicación que inicia la conversación se denomina aplicación destino; la aplicación que responde al destino se denomina aplicación fuente. Una aplicación puede estar ocupada en varias conversaciones simultáneamente, actuando en alguna aplicación como destino y en otras como fuente.

Cuando una aplicación destino inicie una conversación DDE, debe especificar dos cosas:

- El nombre de la aplicación fuente con la que quiere hablar (aplicación que responde).
- El tema de la conversación (datos a intercambiar).

Cuando una aplicación fuente recibe una petición de conversación referente a un tema, la reconoce, y en caso afirmativo inicia la conversación. En general, es la aplicación fuente la que envía datos a la aplicación destino.

Cada aplicación que puede actuar como fuente tiene un único nombre, que normalmente coincide con el nombre del fichero ejecutable sin extensión. Si es una aplicación en Visual FoxPro, su nombre es dado al fichero ejecutable correspondiente a la misma; lo mismo es para Visual Basic, en WinWord (no hay diferencia entre utilizar mayúsculas y minúsculas). Para otras aplicaciones Windows, hay que verificar la documentación correspondiente a las mismas.

DDE DE MICROSOFT VISUAL FOXPRO

Intercambio de datos (DDE)

El intercambio dinámico de datos (DDE) es una característica del ambiente Windows que puede ser accedido a través de algunas aplicaciones Windows que utilicen DDE. El DDE es un protocolo que a través de aplicaciones puede intercambiar datos en una base de tiempo real. Para ejecutar tal cambio, alguna de las dos aplicaciones participantes deberán iniciar la conversación. La aplicación que inicia la conversación será la aplicación cliente, y la aplicación que responde a tal será la aplicación servidora.

DDE y EXTRA ! for Windows

Cuando una aplicación quiere realizar una conversación con EXTRA !, esta inicia una conversación. Por ejemplo, cuando Word para Windows se conecta con EXTRA !, esta inicia una conversación; y lo mismo se aplica a una aplicación desarrollada en Visual FoxPro. La aplicación que inicia la conversación (en este caso el Sistema de Recaudación e Información Predial) es la aplicación cliente. La aplicación que es llamada para iniciar (en este caso EXTRA !) es la aplicación servidora.

La aplicación servidora habla con la aplicación cliente para mandarle los mensajes DDE desde ella hacia el cliente. Para ejecutar los mensajes y funciones DDE es necesario escribirlos dentro de la aplicación cliente para que pueda emplearlos.

Cuando la conversación se inicia el cliente identifica a EXTRA ! como la aplicación servidora y/o en una de las sesiones de terminal de EXTRA ! o como un sistema de conversación típica (información que se define del sujeto de una conversación DDE y representa algunas unidades de datos que significan que ha iniciado una conversación servidora). La aplicación servidora y el tema son especificados en un mensaje llamado INITIATE. Esta especificación nunca podrá ser cambiada después de iniciar una conversación, ya que devuelve un número entero consecutivo para cada comunicación.

Cuando la conversación continua, el cliente y la aplicación servidora hablan entre ambas para intercambiarse mensajes entre una y otra. Cada uno de estos mensajes es asociado con un tema especificado en el mensaje INITIATE. Alguno de estos mensajes pueden también contener un ítem (una referencia hacia un bloque de datos, por ejemplo, un entero, una cadena, un rango de celdas en una hoja de trabajo, un gráfico, o un bitmap; que puede pasar entre dos aplicaciones ocupado en una conversación DDE). Este ítem es el nombre definido por EXTRA !, o un bloque de datos. Por ejemplo, la posición del cursor, nombre de esquema, y número de modelo son todos los ítems definidos por EXTRA !.

En una conversación DDE, los datos que pueden ser desplegados en una pantalla mainframe son referenciados por una localización en la sesión de terminal de espacio de presentación (es una representación lineal que puede ser desplegada en la pantalla del mainframe). Esta inicia en la esquina superior izquierda de la pantalla del mainframe, a cada posición se le da un número. Este número es llamado un posición de espacio de presentación (PSP), en la terminal de sesión de espacio de presentación. El número de PSPs en un espacio de presentación depende del tamaño de la pantalla del mainframe ó su terminal de sesión emulada. Por ejemplo, si la sesión de terminal es emulada en una pantalla con 24 renglones y 80 columnas hay 1920 PSPs ($24 \times 80 = 1920$) en el espacio de presentación de la sesión, iniciando del 1 hasta 1920.

Existe otro camino para referenciar una localización en el espacio de representación y es por un campo. La aplicación mainframe tiene campos protegidos y desprotegidos en la pantalla. Estos campos pueden referenciarse directamente por mensajes DDE, permitiendo insertar o quitar datos de ellos.

Iniciando una liga DDE

Para iniciar una conversación, se necesita enviar un mensaje INITIAL hacia EXTRA!. En este mensaje, se especifica a EXTRA! como una aplicación servidora y una configuración de una sesión de terminal como el tema de conversación. EXTRA! regresará información que la identificará como la aplicación servidora y la sesión de terminal como el tema de la conversación. Algunos programas de Windows permiten un juego de parámetros de tiempo de espera (determinará al programa que tiempo esperará para recibir una respuesta de un mensaje DDE). EXTRA! tiene un tiempo predeterminado de 3 minutos, el cual se puede cambiar. Este caso no funciona para Visual FoxPro, pero por descripción se ejemplifica para Visual Basic.

En Visual Basic se emplea en una caja de texto en la propiedad LinkTimeout, que determina la cantidad de tiempo que el un control (destino, aplicación en Visual Basic) espera para establecer la conversación con la fuente. Se mide en décimas de segundo, y el valor por defecto es 50. Si el tiempo de respuesta de la fuente (aplicación EXTRA!) supera este valor, se genera un mensaje de error. Un valor de -1 significa esperar lo máximo (65535 décimas de segundo). En este caso, se puede finalizar la espera pulsando la tecla ESC.

A continuación se detalla la sintaxis para iniciar una liga, especificando a EXTRA! como la aplicación servidora:
DDEInitiate(cNombreServicio, cNombreTema)
iChannel = DDEInitiate('EXTRA!', 'A')

La cual establece un canal de intercambio dinámico de datos (DDE) entre Visual FoxPro y otra aplicación basada en Microsoft Windows.

cNombreServicio. Especifica el nombre de servicio de la aplicación servidora que, en la mayoría de los casos, es el nombre del archivo ejecutable sin su extensión. El nombre de servicio predeterminado para Visual FoxPro es Visual FoxPro. Si va a establecer un canal con EXTRA!, cNombreServicio será EXTRA!.

cNombreTema. Especifica el nombre de tema. El tema es específico de la aplicación y debe ser comprensible para la aplicación. Por ejemplo, un tema suministrado por la mayoría de servidores DDE es el tema System. Consulte en la documentación de la aplicación los nombres de servicio y de tema proporcionados por la aplicación. El tipo devuelto es numérico. DDEInitiate() establece un canal DDE entre Visual FoxPro y la aplicación servidora DDE. Una vez que el canal se ha establecido, Visual FoxPro puede pedir datos del servidor haciendo referencia al canal en las funciones DDE subsequentes. Visual FoxPro actúa como el cliente, pidiendo datos de la aplicación servidora a través del canal.

Si logra establecerse el canal, DDEInitiate() devolverá el número de canal. Los canales tienen números no negativos, y el número de canales que puede establecer está limitado únicamente por los recursos de su sistema.

DDEInitiate() devolverá -1 si no puede establecerse el canal. Si la aplicación servidora no está abierta, Visual FoxPro le preguntará si desea abrirla. Si elige "Sí", Visual FoxPro intentará abrir la aplicación. (Puede utilizar DDELastError() para determinar por qué no puede establecerse un canal).

Para evitar que se le pregunte si desea abrir la aplicación, establezca la opción SAFETY de DDESetOption(). También puede utilizar RUN con la opción/N para iniciar la aplicación. Un canal puede cerrarse mediante DDETerminate().

Empleando mensajes de ejecución DDE

Para indicarle a EXTRA! que ejecute una tarea, se necesita indicarle el asunto a tratar un mensaje EXECUTE. Por ejemplo, se puede decir a EXTRA! que abra la sesión de una terminal, transferir un archivo, o ejecutar alguna de estas macros.

A continuación se detalla la relación de temas o tópicos:

Sintaxis: Para enviar un mensaje EXECUTE en una aplicación DDE o una función EXECUTE en una macro, se emplea la siguiente sintaxis que especifica que tarea se quiere para que la realice EXTRA!.

[comando]

o

[comando (opciones)]

= DDEExecute(iChannel, "[run macro (logon)]")

Allow Emulator Updates : Permite que EXTRA! actualice su ventana de terminal cuantas veces la pantalla del mainframe se actualice:

= DDEExecute(iChannel, "[allow emulator updates]")

Block Emulator Updates : No permite que EXTRA! actualice su ventana de terminal cuando la pantalla del mainframe se actualice:

= DDEExecute(iChannel, "[block emulator updates]")

Open Layout : EXTRA ! abre las pantallas de archivos de esquema que se especifican (la pantalla de esquema predeterminada es 0, y se almacena en el archivo LAYOUT00.FWL). Se puede incluir la ruta del archivo y su nombre:

= DDEExecute(iChannel, "[open layout (c:\extrawin\user\layout09.ewl)]")

Pause : EXTRA ! coloca la conversación en espera, se mantendrá durante el tiempo especificado. Al terminar continuara con el siguiente comando que le siga:

pause(time)

Donde time se mide en incrementos de medio segundo (por ejemplo, 17 = 8.5 segundos):

= DDEExecute(iChannel, "[pause(10)]")

Send File : EXTRA ! transfiere los archivos especificados desde la estación de trabajo hacia el mainframe. Para transferir un archivo con este comando, se puede colocar una bitácora en el mainframe. Se emplean las siguientes sintaxis para el tipo de transferencia de archivo (CMS, CICS, o TSO) deseado.

Un archivo CMS o CICS emplea el siguiente comando de transferencia:

send file(PC nombre de archivo(opciones de transferencia/tiempo de espera = nn))

donde tiempo de espera = nn (1 a 1000 minutos)

Ejemplo CMS:

= DDEExecute(iChannel, "[send file(c:\temp\temp.txt a :setup text(ascii crlf)]")

Ejemplo CICS:

= DDEExecute(iChannel, "[send file(c:\temp\temp.txt a :setup text(ascii crlf)]")

Archivo TSO emplea el siguiente comando de transferencia:

archivo a transferir(PC nombre de archivo opciones de transferencia/tiempo de espera = nn)

donde tiempo de espera = nn (1 a 1000 minutos)

Ejemplo TSO:

= DDEExecute(iChannel, "[send file(c:\temp\temp.txt a :setup text ascii crlf)]")

Recive File : EXTRA ! transfiere el archivos especificados desde la estación de trabajo hacia el mainframe. Para transferir un archivo con este comando, se puede colocar una bitácora en el mainframe. Se emplean las siguientes sintaxis para el tipo de transferencia de archivo (CMS, CICS, o TSO) deseado.

Un archivo CMS o CICS emplea el siguiente comando de transferencia:

recive file(PC nombre de archivo(opciones de transferencia/tiempo de espera = nn))

donde tiempo de espera = nn (1 a 1000 minutos)

Ejemplo CMS:

= DDEExecute(iChannel, "[recive file(c:\temp\temp.txt a :setup text(ascii crlf)]")

Ejemplo CICS:

= DDEExecute(iChannel, "[recive file(c:\temp\temp.txt a :setup text(ascii crlf)]")

Archivo TSO emplea el siguiente comando de transferencia :

recive file(PC nombre de archivo opciones de transferencia/tiempo de espera = nn)

donde tiempo de espera = nn (1 a 1000 minutos)

Ejemplo TSO:

= DDEExecute(iChannel, "[recive file(c:\temp\temp.txt a :setup text ascii crlf)]")

Run Macro : EXTRA ! corre la macro que se especifique, de acuerdo a la siguiente sintaxis :

run macro(nombre de la macro/tiempo de espera = nn)

donde tiempo de espera = nn (1 a 1000 minutos)

= DDEExecute(iChannel, "[run macro(xlogon)]")

Start Emulator : EXTRA ! abre una ventana que despliega la sesión típica:

= DDEExecute(iChannel, "[start emulator]")

Stop Emulator : EXTRA ! cierra la ventana de una sesión típica:

= DDEExecute(iChannel, "[stop emulator]")

Terminal On : EXTRA ! activa la sesión típica:

= DDEExecute(iChannel, "[terminal on]")

Terminal Off : EXTRA ! desactiva la sesión típica. Sólo para las sesiones conectadas al mainframe:

= DDEExecute(iChannel, "[terminal off]")

Empleando mensajes de petición DDE

Para pedir datos, se necesita el tema de mensajes REQUEST de EXTRA!. Este puede incluir información en el espacio de presentación de una sesión de terminal. El mensaje REQUEST emplea nombres de ítems de referencia con la parte de información solicitada. Los nombres de ítems pueden estar en letras mayúsculas, letras minúsculas, o en una combinación de ambas.

"nombre de ítem"

= DDERequest(iChannel, Cursor)

A continuación, se detalla en los nombres de ítems REQUEST.

Columns: EXTRA! regresa el número de columnas del espacio de presentación de una sesión de terminal:

= DDERequest(iChannel, "Columns")

Configuration Name: EXTRA! regresa la ruta y el nombre del archivo de el archivo de configuración que se esta empleando:

= DDERequest(iChannel, "Configuration Name")

Configuration Topics: EXTRA! regresa una lista de todas las sesiones de terminal definida por la configuración activa:

= DDERequest(iChannel, "Configuration Topics")

Cursor: EXTRA! regresa la localización del cursor del mainframe, en el espacio de presentación de la sesión de terminal:

= DDERequest(iChannel, "Cursor")

Directory Name: EXTRA! regresa el directorio, incluyendo la ruta, donde los archivos de EXTRA! estan localizados:

= DDERequest(iChannel, "Directory Name")

Emulator: EXTRA! regresa la ventana indicadora del despliegado de una ventana de sesión de terminal:

= DDERequest(iChannel, "Emulator")

File Transfer: EXTRA! regresa el nombre corto de la ventana indicadora de la aplicación EXTRA! que se esta empleando en el archivo de transferencia. Si no hay transferencia el valor regresado es 0:

= DDERequest(iChannel, "File Transfer")

Keyboard: EXTRA! regresa el estado del teclado de una sesión típica (clear, locked, o Xclock). Xclock indica que el teclado esta esperando la finalización de un proceso en el mainframe del ultimo comando ejecutado:

= DDERequest(iChannel, "Keyboard")

Keystroke: EXTRA! regresa el teclazo introducido en la sesión de terminal.

Mnemotecnia de teclas en viadas por HL.LAPI:

La tabla de traducción mnemotecnia PC-3270 provee el empleo de caracteres ASCII para representar funciones especiales de un teclado 3270 cuando son enviados estos con un mensaje POKE. Notese la siguiente diferencia, por ejemplo, @A es una tecla diferente a @a.

= DDERequest(iChannel, "Keystroke")

ASCII Keys	3270 Translation	ASCII Keys	3270 Translation
@A	Alt	@4	PF4 *
@B	Backtab	@5	PF5 *
@C	Clear	@6	PF6 *
@D	Delete Char	@7	PF7 *
@E	Enter*	@8	PF8 *
@F	Erase EOP*	@9	PF9 *
@H	Help	@a	PF10 *
@I	Insert	@b	PF11 *
@J	Jump	@c	PF12 *
@K	Copy	@d	PF13 *
@L	Cursor Left	@e	PF14 *
@M	Enlarge	@f	PF15 *
@N	New Line	@g	PF16 *
@P	Print	@h	PF17 *

ASCII Keys	3270 Translation	ASCII Keys	3270 Translation
@Q	Shift	@k	PF20 *
@T	Tab	@l	PF21 *
@U	Cursor Up	@m	PF22 *
@V	Cursor Down	@n	PF23 *
@X	Reserved	@o	PF24 *
@Y	Caps Lock	@x	PA1 *
@Z	Cursor Right	@y	PA2 *
@<	Backspace	@z	PA3 *
@0	Home	@A@H	SysReq
@1	PF1 *	@A@Q	Attn
@2	PF2 *	@A@J	CurSel
@3	PF3 *	@A@F	Erlnp

* Indica teclas de ayuda.

layout Name: EXTRA ! regresa el nombre de la pantalla activa en esquema empleado por EXTRA ! :
 DDERequest(iChannel, "Layout Name")

Model: EXTRA ! regresa el número de modelo de la pantalla de emulación del mainframe empleadapor la sesión del terminal :

- = Modelo 2 (24 X 80)
- = Modelo 3 (32 X 80)
- = Modelo 4 (43 X 80)
- = Modelo 5 (27 X 132)

DDERequest(iChannel, "Model")

OIA: EXTRA! Regresa el contenido de la sesión de terminal. Area de Información(OIA). Los datos se regresan en formato HLLAPI OIA:

DDERequest(iChannel, "OIA")

Power: EXTRA! Regresa el estado de activación de la sesión de terminal. Esta petición determina si la sesión típica esta activa (ON) o inactiva (OFF):

DDERequest(iChannel, "Power")

PS: EXTRA! regresa el contenido del espacio de presentación, especificando una localización:
 sintaxis, nombre del item:

"PS"

F<numero>[U|P]"

Pnnnn[LmmmmF]"

Ejemplos de sintaxis:

"PS" : Pide todo el espacio de presentación.

"F2" : Pide el segundo campo del espacio de presentación.

"F2U" = Pide el segundo campo sin protección del espacio de presentación.

"F2P" = Pide el segundo campo protegido del espacio de presentación.

"P100" = Pide datos de la posición 100 hasta el final de un campo que este localizado en la posición 100.

"P100F" = Pide el campo número 100.

"P100L20" =Pide 20 caracteres iniciando en la posición 100.

DDERequest(iChannel, "PS")

DDERequest(iChannel, "P577L20")

Rows: EXTRA! regresa el número de filas del espacio de presentación de una sesión de terminal:

DDERequest(iChannel, "Rows")

Search: EXTRA! regresa la posición de la cadena buscada, en el caso de no encontrarse regresa cero:

DDERequest(iChannel, "Search", "Ready")

DDERequest(iChannel, "Search")

SysItems: EXTRA! regresa una lista de todos los nombres especificados a los cuales pueden pedirse datos:

DDERequest(iChannel, "SsysItems")

Timeout: EXTRA! Regresa el valor del parámetro, para el tiempo de espera:
= DDERequest(iChannel, "Timeout")

Topic: EXTRA! Regresa el nombre de la sesión de terminal:
= DDERequest(iChannel, "Topic")

Topics: EXTRA! Regresa una lista de todas las sesiones de terminal abiertas:
= DDERequest(iChannel, "Topics")

Envío de datos a la emulación

Para enviar datos a la emulación, se emplea el tema de mensajes POKE. En estos datos pueden incluirse teclas, cadenas de caracteres, o una nueva posición del cursor del mainframe. Los mensajes POKE emplean nombres de Items para referirse a cual parte de la información se quiere enviar a EXTRA!. Los nombres de Items pueden contener letras minúsculas, letras mayúsculas, o una combinación de ambas.
= DDEPoke(iChannel, Keystroke, "@E")

Los temas relacionados se detallan a continuación:

Cursor: Envía al cursor a una nueva posición de EXTRA!. El cursor del mainframe se mueve a la nueva posición del espacio de presentación de la sesión de terminal.

Sintaxis de "valor":

"PSP"

"F<nnn>[U/P]"

Ejemplos de sintaxis:

"I" = El lugar del cursor será la primera posición del espacio de presentación.

"F2" = El lugar del cursor será en el inicio del segundo campo del espacio de presentación.

"F2U" = El lugar del cursor será en el inicio del segundo campo sin protección del espacio de presentación.

"F2P" = El lugar del cursor será en el inicio del segundo campo con protección del espacio de presentación.

= DDEPoke(iChannel, "Cursor", "I25I")

Keystroke: Envía una tecla, o una cadena de teclas, hacia la sesión de terminal. La cadena puede contener un máximo de 255 caracteres. Los datos se insertan en la posición actual del cursor del mainframe en el espacio de presentación de la terminal de sesión.

= DDEPoke(iChannel, "Keystroke", "@E")

PS: Inserta una cadena de caracteres en la posición especificada. La longitud de la cadena sólo se limita por el final del espacio de presentación. Los caracteres no se insertaran si sobrepasa el límite del espacio de presentación:

Sintaxis de "valor":

"PS", "Cadena de caracteres"

"F<nnn>[U]", "Cadena de caracteres"

"Pnnnn[LmmmmF]", "Cadena de caracteres"

Ejemplos de sintaxis:

"PS" = Inserta una cadena de caracteres en el inicio del espacio de presentación (renglón 1, columna 1).

"F2" = Inserta una cadena de caracteres al inicio del segundo campo del espacio de presentación. Si el segundo campo esta protegido, se desplegara un mensaje de fallo.

"F2U" = Inserta una cadena de caracteres en el segundo campo sin protección.

"P100" = Inserta una cadena de caracteres, al inicio de la posición 100 y terminara al final del campo que esta en la posición 100.

"P100F" = Inserta una cadena de caracteres al principio del campo que guarda la posición 100.

"P100L20" = Inserta veinte caracteres, iniciando en la posición 100.

= DDEPoke(iChannel, "PS", "logon")

Search: Define una cadena de caracteres que puede buscar, para utilizar el comando de búsqueda REQUEST:

= DDEPoke(iChannel, "Search", "Ready")

= DDERequest(iChannel, "Search")

Timeout: Juego de parámetros del DDE de EXTRA! (Predeterminado = 3 minutos). Se puede cambiar este parámetro por un valor que oscile de 1 a 1000.

= DDERequest(iChannel, Chan, "Timeout", "6")

Crear vínculos entre Visual FoxPro y EXTRA!

Creará un vínculo de notificación o un vínculo automático que se utiliza en el intercambio dinámico de datos. El tipo de datos devuelto es un valor lógico. `DDEAdvise()` se utiliza para crear un vínculo de notificación o un vínculo automático con un nombre de elemento en una aplicación servidora. Cuando se crea un vínculo de notificación con `DDEAdvise()`, la aplicación servidora notifica a Visual FoxPro que el nombre del elemento se ha modificado y transfiere los nuevos datos a Visual FoxPro. Antes de que pueda crear un vínculo, debe establecer un canal con la aplicación servidora mediante `DDEInitiate()`.

También puede utilizar `DDEAdvise()` para desactivar la notificación desde el servidor. `DDEAdvise()` devuelve el valor verdadero (.T.) si se ejecuta con éxito; de lo contrario, devuelve falso (.F.).

`DDEAdvise(iChannel, 'LIC1', 'newdata', 1)`

Donde el primer parámetro especifica el número de canal; el segundo especifica el nombre del elemento. Por ejemplo, Microsoft Excel utiliza la notación de fila y columna para referirse a las celdas de una hoja de cálculo. El nombre de elemento LIC1 designa la celda de la primera fila y la primera columna de la hoja de cálculo.

El tercer parámetro, especifica la función definida por el usuario (FDU) que se ejecuta cuando se establece un vínculo de notificación o un vínculo automático y se modifica el elemento del nombre de elemento. Cuando se ejecuta la FDU, se transfieren los seis parámetros siguientes, en el orden que se indica a continuación:

Parámetro	Contenido
Número de canal	Número de canal de la aplicación servidora
Acción	ADVISE o TERMINATE
Elemento	Nombre del elemento; por ejemplo, LIC1 para una celda de una hoja de cálculo de Microsoft Excel
Datos	Nuevos datos (vínculo automático) o la cadena vacía (vínculo de notificación)
Formato	Formato de datos, por ejemplo, CF_TEXT
Estado de aviso	Tipo de vínculo (0 = manual, 1 = notificación, 2 = automático)

La FDU debería tener seis parámetros en su instrucción `LPARAMETER` o `PARAMETER` para aceptar los valores transferidos desde la aplicación servidora. Si se establece un vínculo de notificación, se ejecutará la FDU y se transferirá la cadena vacía en el parámetro datos. Puede emitir `DDERequest()` posteriormente para recuperar los datos. Si se establece un vínculo automático, la FDU se ejecutará y los datos se transferirán en el parámetro datos.

El parámetro Acción contiene ADVISE cuando el servidor está actualizando el vínculo. Se llama a la FDU, y el parámetro Acción contendrá TERMINATE cuando el cliente o el servidor cierre el vínculo.

Se pasa por alto cualquier valor devuelto por la FDU.

El cuarto parámetro, especifica el tipo de vínculo de la forma siguiente:

nTipoVínculo	Tipo de vínculo
0	Manual
1	Notificación
2	Automático

Puede desactivar la notificación desde la aplicación servidora especificando 0 en el número de tipo de vínculo. Si cambia el elemento, la FDU no se ejecutará.

`DDEAdvise(iChannel, 'LIC2', 'newdata', 2)`

Determinar errores de funciones DDE

Devuelve el número de error de la última función de intercambio dinámico de datos. Puede utilizar `DDELastError()` para ayudar a determinar la causa de un error cuando una función DDE no se ejecuta satisfactoriamente. `DDELastError()` devolverá 0 si la última función DDE se ejecuta correctamente. Devolverá un valor distinto de cero si la última función DDE fracasa. La tabla siguiente muestra los números de errores y sus descripciones:

Número de error	Descripción
1	Servicio ocupado
2	Tema ocupado
3	Canal ocupado
4	No hay tal servicio
5	No hay tal tema
6	Canal defectuoso
7	Memoria insuficiente

Número de error	Descripción
8	Tiempo de espera de acuse de recibo
9	Tiempo de espera de petición
10	No DDE Initiated
11	El cliente intentó transacción del servidor
12	Tiempo de espera de ejecución
13	Parámetro erróneo
14	Poca memoria
15	Error de memoria
16	Fallo en la conexión
17	Fallo en la petición
18	Tiempo de espera de escritura en memoria
19	No puede mostrar el mensaje
20	Transacciones síncronas múltiples
21	Servidor inactivo
22	Error interno DDE
23	Tiempo de espera de aviso
24	Identificador de transacción no válido
25	Desconocido

= DDELastError()

Terminar una conversación con EXTRA!

Para terminar una conversación con EXTRA!, se emplea el tema de mensaje TERMINATE:

= sResult = DDETerminate(iChannel)

Funciones DDE de EXTRA! para Visual FoxPro

Intercambian datos entre Visual FoxPro y otras aplicaciones basadas en Microsoft Windows. Visual FoxPro puede actuar como servidor o como cliente para enviar y recibir datos a y desde otras aplicaciones basadas en Microsoft Windows. Las aplicaciones que admiten DDE (intercambio dinámico de datos) utilizan las siguientes convenciones de nombres.

Nombre	Descripción
Nombre del servicio	Un nombre al que responde el servidor cuando un cliente intenta acceder al servidor. Un servidor puede admitir muchos nombres de servicio
Nombre del tema	Un nombre que especifica un conjunto lógico de datos. Para aplicaciones basadas en archivos, los nombres de tema son normalmente nombres de archivo. En otras aplicaciones, los nombres de tema son específicos de la aplicación. Para acceder al servidor, el cliente debe especificar un nombre de tema además del nombre de servicio del servidor
Nombre del elemento	Un nombre que especifica una unidad de datos que el servidor puede transferir al cliente que pide los datos

Para pedir datos de otra aplicación, cree un programa de Visual FoxPro que establezca Visual FoxPro como cliente. A continuación se ofrece una breve orientación sobre la creación de un programa de Visual FoxPro sencillo que pide datos de otra aplicación:

- Establezca un vínculo con la aplicación servidora mediante DDEInitiate().
- Si el vínculo se establece con éxito, utilice DDERequest() para pedir datos de la aplicación servidora. DDERequest() puede emitirse de forma repetitiva para pedir datos adicionales.
- Una vez recibidos los datos, emita DDETerminate() para terminar el vínculo con la aplicación servidora y liberar recursos del sistema.

Las funciones anteriores establecen un vínculo frío. Un vínculo frío se produce cuando el cliente inicia todas las comunicaciones entre las aplicaciones. Para ver una descripción de otros tipos de vínculos, vea DDEAdvise().

A continuación se ofrece una breve orientación para un programa que establece Visual FoxPro como servidor:

- Utilice DDESetService() para crear un servicio y especificar el tipo de servicio.
- Utilice DDESetTopic() para crear un tema de servicio y especificar el procedimiento que se ejecutará cuando se especifique el tema en una petición de cliente.

- Cree el procedimiento especificado en DDESetTopic() para aceptar los parámetros transferidos al procedimiento.
- Dentro del procedimiento, procese la petición y, si corresponde, devuelva los datos pedidos al cliente.

Observe que estas funciones DDE se diferencian de las convenciones de las funciones previas de Visual FoxPro en los aspectos siguientes:

- Los cuatro primeros caracteres de estas funciones no son únicos.
- Los nombres de función exceden de diez caracteres y no pueden abreviarse.

Función DDE	Descripción
DDEAbortTrans()	Termina una transacción DDE asíncrona
DDEAdvise()	Crea un vínculo de notificación o un vínculo automático utilizado en DDE
DDEEnabled()	Activa o desactiva el proceso DDE o devuelve el estado del proceso DDE
DDEExecute()	Utilizando DDE, envía un comando a otra aplicación
DDEInitiate()	Establece un canal DDE entre Visual FoxPro y otras aplicaciones basadas en Microsoft Windows
DDELastError()	Devuelve un número de error para la última función DDE
DDEPost()	Envía datos entre las aplicaciones cliente y servidora en una conversación DDE
DDERequest()	Pide datos a la aplicación servidora en una conversación dinámica DDE
DDESetOption()	Cambia o devuelve la configuración DDE
DDESetService()	Crea, libera o modifica los nombres de servicio y la configuración de Visual FoxPro
DDESetTopic()	En una conversación DDE, crea o libera un nombre de tema de un nombre de servicio
DDETerminate()	Cierra un canal DDE establecido mediante DDEInitiate()

DISEÑO DE LA APLICACIÓN CLIENTE/SERVIDOR

Los hay de código pequeño pero muy poderosos; crípticos y casi impenetrables; los hay con interfaces gráficas rudimentarias o con interfaces que son una auténtica obra de arte pero de bajo rendimiento y poco productivos.

Los paquetes más populares son los que tienen una mezcla perfecta de código, interfase apropiada y precio competitivo. Con esto en mente, los desarrolladores privados deben competir en términos de calidad cada vez que lanzan un nuevo producto en el mercado, compitiendo quizás desventajosamente con las campañas agresivas de publicidad de los productos ya establecidos. Esto no es motivo de cejar en los empeños y hacer de la calidad su mejor estrategia de venta.

Como uno de los parámetros para evaluar un software está su utilización sencilla, intuitiva, una apariencia agradable a la vista y, con el ambiente Windows, la presentación de iconos, cuadros y ventanas fáciles de operar.

La definición de un producto como utilizable, aún en la etapa de diseño y prueba, es que el producto permita al usuario concentrarse en la tarea que está haciendo o en la obtención de un objetivo, y no en forma de utilización del paquete de computación para llevar a cabo las tareas requeridas.

Las sesiones de prueba de un producto son más estresantes para el programador que para el usuario. En ocasiones cuando una persona está luchando con un programa para hacer algo, el programador debe reconocer que las cosas no son tan fáciles como él cree y tiene que hacer modificaciones y mejoras a su programa.

El usuario piensa que va a lograr su meta haciendo esto y lo otro con el programa pero, cuando el programa no responde como él quiere, entonces es donde comienza la etapa de intuición; se puede escuchar lo que piensa la persona y ver como interactúa con la interfase hasta resolver el problema.

Una interfase no va a trabajar perfectamente para todos porque cada uno de nosotros va a armar una hoja de cálculo de distinta manera. Cuando se crea una interfase cercana a lo que todas esperamos, que sirva a todos en términos generales, que nos guíe para realizar nuestra tarea y nos permita trabajar en la manera más eficiente, entonces es una interfase exitosa.

Si el sujeto se tarda en hacer algo y lo intenta varias veces con pocos resultados significa que debemos cambiar algo allí. Cuando se presenta una interfase y nadie hace algo con ella después de cinco minutos, tenemos un gran problema.

Nosotros no sólo le decimos a los desarrollados qué funcionó y qué no funcionó, lo decimos porqué no funcionó.

La presencia del equipo desarrollador en las sesiones de prueba de usuario es importante porque aprenden cómo la gente está trabajando con sus proyectos y los puntos de la interfase en los que deben mejorarla. Con esta observación directa tratamos de apegarnos a la filosofía de que el cliente tiene la razón y hacer las cosas que le satisfagan más.

DISEÑO DE LA INTERFASE GRAFICA

Con la introducción del ambiente Windows la utilización de las interfaces gráficas ha recibido un gran impulso (a pesar de que este concepto no es nuevo, pues Apple lo ha explotado por años), presentándose fenómeno: Los programadores conocen de lenguaje de computación, de utilerías y diseño de sistemas, pero no saben nada de percepción visual, percepción auditiva, puntos de atención, teoría del color, comunicación visual, ergonomía y diseño gráfico, por lo

que al crear un nuevo producto cometen errores, grandes o pequeños, al establecer las pantallas de interacción entre el usuario y el paquete.

En esta era la interfase amigable debe tener un diseñador industrial o gráfico que nos dé su valoración del producto en la etapa de prototipo. La utilización de diferentes tonos de gris en la interfase de Windows no se hizo por azar, se quiere dar la impresión de tres dimensiones y de movimiento en los botones. Esto obedece a la idea de hacer que los elementos de la pantalla, aún en la primera ocasión en que un usuario trabaja con ella, no la sean del todo extraños o quizás que sean cercanos a su realidad.

Por ejemplo, el caso de una pantalla con una línea negra alrededor de la palabra Ok al ir hacia ella, dar un clic y no tener una respuesta, la gente no puede afirmar que su elección se ejecutará; pero cuando se presenta en la pantalla una imagen tridimensional de un botón (utilizando diferentes tonos de gris con lo que el botón en la pantalla realmente da la impresión de sobresalir), al dar un clic el botón se sume y vuelve a salir, esto da una impresión muy cercana al mundo real y la persona está segura de que su elección sí será ejecutada.

En las pantallas actuales solamente tenemos 256 colores y el único que podemos utilizar en el efecto de tres dimensiones es el gris, en realidad se utiliza el blanco, gris claro, gris oscuro y negro. Las personas escogen sus propios colores para la interfase conforme lo permite el equipo.

Al elegir los colores de las interfases debemos ser meticulosos, pues las personas somos emocionales ante los colores, casi todo el mundo acepta el azul y el rojo, pero fuera de estos colores, se tienen diferentes opiniones incluso a causa de diferencias culturales.

Debemos buscar colores para fondos de las pantallas que permitan ver perfectamente los textos, así las personas verán con mayor exactitud su trabajo. El blanco de fondo de la idea de papel y la letra negra le da realismo a un escrito, pero es una pantalla muy brillante para trabajar. Debemos pensar en colores que no fatiguen al usuario, es mejor lo sencillo y directo que de el área de trabajo comodidad de movimiento en el default, las operaciones de cambio de color de la pantalla o los textos para elección del usuario puede ser usados como un plus al producto.

Para los iconos se busca acercarse a la realidad y los objetos representados tienen colores naturales. Los colores son muy importantes, pues marca la respuesta del programa a una acción realizada por el usuario, y los colores resaltados marcan los puntos que el desarrollador quiere enfatizar en una pantalla. Si se quiere llamar la atención de un usuario sobre algo, se usa un color más fuerte o dominante; si se desea dar la sensación de movimiento también se deben utilizar colores dentro de esta gama. Si una pantalla tiene muchos colores la vista estará en movimiento todo el tiempo y hará que el usuario le sea más difícil concentrarse y reaccionar. Se prefiere ser conservador, así el usuario podrá permanecer 6 u 8 horas frente a la misma interfase sin que se fatigue.

La tarea de crear interfases gráficas y software en general, es difícil cuando se trabaja para un país con la misma forma de pensar y hacer las cosas, pero cuando se trabaja para el mercado internacional, con una gama amplia de modos de pensar, esto es aún más complicado. El equipo de psicólogos de laboratorio de Microsoft tiene que viajar a los distintos mercados. El estudio de las diferencias culturales es definitivo cuando para cuando se trabaja para un mercado mundial.

ANÁLISIS DE INFORMACION

Actualmente se necesitan personas que puedan desarrollar especificaciones correctas y completas ya que los administradores están preocupados por la elevación de costos y por la baja productividad. El secreto de un buen diseño se encuentra en la información, por lo tanto, debemos recurrir a los expertos de la información y ellos son los usuarios.

Para realizar un buen diseño, primeramente debemos entender el *problema* antes de diseñar una solución. Ya que de otra forma se gastara una mayor cantidad de dinero en algo de poco valor y se gastara una cantidad adicional de dinero para salvar algo de valor.

Método tradicional

- Se asume un particular medio ambiente técnico.
 - Dificultando las conversaciones.
- El problema se define en términos de su implantación técnica.
 - Confusión del usuario, problemas para la aceptación.
- El usuario desalentado a involucrarse.
 - Mantenimiento para corregir errores, suposiciones equivocadas.

Los usuarios insatisfechos por algunos resultados:

- Los analistas de computación entienden mal y descuidan algunos aspectos del problema.

- La terminología computacional dificulta el entendimiento y la evaluación.
- No se aprueban los conceptos hasta que el presupuesto se agota.
- Adiciones al sistema básico son frustrantemente costosas, revelan mayores problemas, son realizadas con parches.

Los analistas también frustrados:

- Los usuarios tienen supuestos no comunicados.
- Múltiples usuarios tienen necesidades conflictivas, aparentemente no se entienden uno al otro.
- La terminología técnica por parte del usuario.
- Gran imaginación influenciada por experiencias pasadas difíciles.
- Restricciones pre-impuestos, algunas veces impiden la solución del problema.
- El enfoque de añadidos no trabaja para problemas grandes.
- Como hacer que el usuario distinga entre sus deseos y sus necesidades.
- Es imposible ser experto en múltiples áreas.
- Apoyo de los directivos.

Por lo tanto, la administración es afectada:

- Los sistemas actuales y sus autores están atados a la primera generación de tecnología.
- El soporte de aplicaciones limita las oportunidades de negocios y mejoras.
- Baja productividad en el desarrollo de sistemas.
- Diferencia entre lo que el director piensa que se le habla prometido y lo que se le entregó inicialmente y el costo de corregir el sistema.
- Se dan cambios imprevistos.
- Control de cambios.

Análisis de información

Se desarrollo por el reconocido científico en información, Dr. C: M. Nijssen. Como un resultado de diez años de investigación, desarrollo y pruebas en la materia. El uso es de alcance mundial.

Metodología de definición de requerimientos:

- Medios para incrementar la efectividad de comunicación entre el analista de información y el usuario, basados en lenguaje natural.
- Un conjunto de pasos precisos para el análisis y especificación de requerimientos para solucionar un problema-producto principal: contrato de información.
- Validación de los requerimientos mediante el prototipo del modelo de información.

Estrategia de implantación:

- Arquitectura de coexistencia.
- Interfase normalizada.

El enfoque de análisis de información:

- Se deja a un lado el ambiente técnico para concentrarse en el ambiente del usuario.
 - Libertad para comunicarse.
- Se define el problema en términos de su estructura fundamental.
 - Facilidad de comprensión.
- Se motiva al usuario para que se comprometa.
 - Para resolver correctamente el problema completo.

Axioma ENALIM

De lo anterior, toda *comunicación* entre un usuario y un sistema de información consiste de un conjunto de ocurrencias del *lenguaje natural*. ENALIM (Envolving Natural Language Information Model).

Esquema de la base de oraciones:

- Los datos por si mismos no tienen significado.
- En el lenguaje natural (español), los verbos y la estructura de la oración dan significado a los nombre y sustantivos en las oraciones.

- La base de oraciones contienen datos.
- El esquema de la base de oraciones contiene la información estructural que da significado a los datos.

La base de oraciones consiste en un conjunto de ocurrencias de oraciones, en donde el contenido de una base de oraciones se actualiza únicamente:

- Agregando una oración específica.
- Borrando una oración específica.

Una gramática de una base de oraciones consiste de:

- Un conjunto de tipos de oraciones.
- Un conjunto de tipos de nombres de objetos.
- Un conjunto de restricciones a la base de oraciones.

Axioma INSYGRAM

Toda comunicación posible entre un usuario y un sistema de información puede ser escrita por una gramática única, la gramática de la base de oraciones. INSYGRAM (Information System Grammar).

Un validador de la gramática de una base de oraciones, es la unidad la cual es (entre otras), responsable de garantizar que todos los estados y transiciones, satisfagan la gramática de la base de oraciones. Por lo tanto, un programa de aplicación es la interfase entre el usuario y el validador de la gramática de la base de oraciones.

La gramática de la base de oraciones (nivel signficacional), se puede transformar un esquema relacional normalizado (quinta forma normal), por medio del siguiente algoritmo:

- Combinar todas aquellas oraciones elementales, las cuales tengan una restricción de singularidad en un rol refiriéndose a un mismo tipo de objeto y eliminar de esta combinación, todos los roles que son objeto de la restricción de singularidad, excepto uno.
- Agregar al esquema, todas aquellas oraciones que no se pudieron combinar.

METODOLOGIA PARA EL DESARROLLO DE SISTEMAS

Un buen diseño de base de datos proporciona a la aplicación gran flexibilidad, simplificando así el mantenimiento del sistema. Sin esta infraestructura para el desarrollo el programador no podrá superar un diseño deficiente.

A continuación se presenta la propuesta de metodología de análisis de información para realizar el diseño de la base de datos, así como el diseño del sistema; todo esto en 17 pasos:

1. Toda aplicación se genera para dar una solución a la necesidad o por iniciativa del área de computo.
2. Involucrarse en la información de la aplicación.
3. Funcionamiento actual.
4. Si es necesario el diseñador tendrá que obtener una muestra.
5. Solicitar formatos de salida, con sus respectivas restricciones.
6. Análisis de cada formato:
 - Oraciones compuestas.
 - Oraciones elementales o simples.
7. Tablas de población.
8. Diagrama ENALIM.
9. Diagrama ENALIM final.
 - Eliminar objetos similares.
 - Identificar campos similares.
10. Diseño de tablas, bases de datos y/o archivos.
 - Eliminar campos redundantes.
 - Estandarizar tipos y nombres de campos, así como, la longitud.
 - Diccionario de datos.
11. Modulación.
12. Diseño de pantallas.
13. Descripción de programas.
14. Asignación de programas para captura y aplicación.
15. Pruebas parciales y en paralelo.
16. Liberación.
17. Mantenimiento.

1. Toda aplicación se genera para dar una solución a la necesidad o por iniciativa del área de computo.

Remitirse a la parte de Introducción de esta tesis.

2. Involucrarse en la información de la aplicación.

Actualmente la consulta de saldos y adeudos en periodos establecidos para los pagos y/o en fechas de vencimiento se realizan en forma automatizada; en terminales que emulan una sesión VTAM 3270. En este caso la reposición predial se presenta una pantalla emulando el host central directamente. La pantalla de emulación a través de un programación en el host central, muestra un menú principal (PA00); albergando en ella una serie de 13 aplicaciones:

PA01	Menú de mantenimiento
AG00	Menú de agua
AG15	Alta de cuenta a emitir fuera de rango
AG20	Actualización de registro de control
AG21	Actualizaciones leyendas de emisión
AG22	Menú de reposición de agua
PR00	Menú predial
PR20	Actualización porcentajes y fechas de veredicto
PR21	Actualización de leyendas de emisión de predial
PR22	Menú de reposición de predial
PR29	Traslado de pagos del impuesto predial
PR31	Consulta traslado de pagos del impuesto predial
PR33	Consulta por alta y/o fecha de grabación

El sistema de predial utiliza el menú de predial, que es la aplicación PR00; que a su vez contiene 10 aplicaciones que se detallan a continuación:

PR01	Consulta padrón (cuenta)
PR03	Consulta padrón (nombre)
PR04	Consulta padrón (ubicación)
PR05	Consulta pagos (cuenta)
PR09	Consulta créditos (No. de crédito o cuenta)
PR11	Estado de cuenta y recibo normal
PR14	Recibo anual o anticipado
PR23	Consulta catastral
PR25	Corrección de pagos
PR26	Consulta de autodeterminados (cuenta)

El sistema de información y recaudación predial sólo empleara la aplicación Estado de Cuenta y Recibo Normal (PR11).

3. Funcionamiento actual.

El proceso manual para consultar adeudos y saldos de predial se muestra a continuación:

1. Actualmente, se requiere que el contribuyente lleve su recibo anterior a la ventanilla correspondiente.
2. La persona a cargo teclea el número de cuenta impreso en el recibo anterior y de esta forma puede consultar los adeudos del contribuyente.
3. Se imprime el recibo de adeudos con la especificación de reposición de recibo, comprobante de no adeudos o impresión de adeudo según el caso.
4. Si el recibo impreso es un adeudo, la persona puede pasar a cajas y saldar su cuenta.
5. En todas las administraciones tributarias se pueden realizar consultas de saldos a través de búsquedas por cuenta, nombre del contribuyente, y domicilio de notificaciones.
6. Cada operación lleva un número de folio o número de operación la cual se graba en la base de datos central en el caso de la impresión del predial.

Para la impresión de los recibos del predial se emplea una impresora de matriz, en un formato impreso que cuenta con original y dos copias; además de un código de barras.

La información que rodea la infraestructura de kioscos multimedia que posee la Secretaría de Finanzas en las administraciones tributarias, se cita a continuación:

1. Existen 30 unidades ya instaladas, posteriormente se instalarán 10 más en oficinas auxiliares.
2. El kiosco multimedia con el que se cuenta tiene un sistema cargado en disco duro que sólo muestra información de registros para el pago de predial y de otros tipos de trámite (agua, tenencia, contribución de mejoras, espectáculos públicos, etc.), así como las localizaciones de las diferentes oficinas de recaudación.
3. Los kioscos no se encuentran conectados a la red local, o al host central.
4. Este kiosco multimedia incluye un cálculo de predial, el cual se realiza en base a los datos proporcionados por el contribuyente.
5. Los kioscos instalados cuenta con una estructura de madera, conectándose con la siguiente configuración :
 - Microprocesador Pentium a 133 Mhz
 - Disco duro de 1.2 GB
 - MB en RAM
 - Bocinas de 60 Watts
 - Pantalla SVGA con resolución de 600 x 800
 - Touchscreen
 - Windows 3.11

4. Si es necesario solicitar una muestra.

Remitirse al apéndice B, donde se muestra un recibo emitido de adeudo predial.

5. Solicitar formatos de salida, con sus respectivas restricciones.

TESORERIA DEL DEPARTAMENTO DEL DISTRITO FEDERAL SECRETARIA DE FINANZAS

SISTEMA DE RECAUDACION E INFORMACION PREDIAL

CUENTA PREDIAL
369 733 03 151 1

NÚMERO DE OPERACION
716607 7

PERIODO				IMPUESTO	ACTUALIZADO	RECARGOS	TOTAL
DEL		AL					
MES	AÑO	MES	AÑO				
1	98	1	98	21.00		4.00	25.00
2	98	2	98	21.00		3.00	24.00
3	98	3	98	21.00		2.00	23.00
4	98	4	98	21.00		1.00	22.00
TOTALES				84.00		10.00	94.00

	 013697330315100000947166070	FECHA DE EMISION	
		06 10 98	10 19 31
NOMBRE Y FIRMA DEL CONTRIBUYENTE	EMITIDO EN CENTRAL		

Restricciones :

1. Cuenta predial única; 12 caracteres: (1-11, cuenta; 12-1, dígito verificador >= 1 & <= 9)
Cuenta : (1-3, región >= 001 & <= 999); (4-6, manzana >= 001 & <= 999);
(7-8, lote >= 01 & <= 99) ; (9-11, condominio >= 001 & <= 999) ; (12-1, DV >= 1 & <= 9)
2. Período inicial; 4 caracteres: (1_2, mes; 3_2 año)
mes >= 01 ; <= 12
año >= 01 ; <= 99
3. Período final; 4 caracteres: (1_2 mes; 3_2 año)
Mes >= 01 ; < 12
Año >= 01 ; <= 99
4. Dígito rectificador ; 7 caracteres. (1_6, número consecutivo) (7_1, dígito verificador) número consecutivo >= 000001 ; <= 999999 Dígito verificador >= 1 ; <= 9
5. Impuesto ; 11 Número ; 2 decimales, impuesto >= 0.00 ; <= 99999999.99
6. Actualizado ; 11 numérico ; 2 decimales, actualizado >= 0.00 ; <= 99999999.99
7. Recargos ; 11 numérico ; 2 decimales, recargos >= 0.00 ; <= 99999999.99
8. Total ; 13 numérico ; 2 decimales, totales >= 0.00 ; <= 999999999.99

6. Análisis de cada formato.

• Oraciones compuestas:

1. El número de cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0198 al 0198, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 4.00 y un total de \$ 25.00. A la cual se le asigna el número de operación 7166077.
2. El número de cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0298 al 0298, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 3.00 y un total de \$ 24.00. A la cual se le asigna el número de operación 7166077.
3. El número de cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0398 al 0398, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 2.00 y un total de \$ 23.00. A la cual se le asigna el número de operación 7166077.
4. El número de cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0498 al 0498, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 1.00 y un total de \$ 22.00. A la cual se le asigna el número de operación 7166077.

• Oraciones elementales:

1. La cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0198 al 0198, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 4.00 y un total de \$ 25.00
2. La cuenta predial 369 733 03 151 l paga su cuota predial a la cual se le asigna el número de operación 7166077.
3. La cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0198 al 0198, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 3.00 y un total de \$ 24.00.
4. La cuenta predial 369 733 03 151 l paga su cuota predial a la cual se le asigna el número de operación 7166077.
5. La cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0198 al 0198, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 2.00 y un total de \$ 23.00.
6. La cuenta predial 369 733 03 151 l paga su cuota predial a la cual se le asigna el número de operación 7166077.
7. La cuenta predial 369 733 03 151 l paga la cuota predial el periodo de 0198 al 0198, el impuesto de \$ 21.00, actualizado de \$ 0.00, recargos de \$ 1.00 y un total de \$ 22.00.
8. La cuenta predial 369 733 03 151 l paga su cuota predial a la cual se le asigna el número de operación 7166077.

Las oraciones elementales deben escribirse sin ambigüedades, esto es, identificando de la oración a:

- Tipos de objeto
- Tipos de referencia
- Referencias
- Predicado

1	La cuenta predial	(Tipo de objeto)
	Con número de cuenta	(Tipo de referencia)
	369 733 03 151 1	(Referencia)
	Paga la cuota predial el	(Predicado)
	Periodo	(Tipo de objeto)
	De	(Tipo de referencia)
	0198	(Referencia)
	Al	(Tipo de referencia)
	0198	(Referencia)
	El	(Predicado)
	Impuesto	(Tipo de referencia)
	\$ 2.00	(Referencia)
	Actualizado	(Tipo de referencia)
	\$ 0.00	(Referencia)
	Recargos	(Tipo de referencia)
	\$ 4.00	(Referencia)
	Con un	
	Total	(Tipo de referencia)
	\$ 25.00	(Referencia)

2	La cuenta predial	(Tipo de objeto)
	Con número de cuenta	(Tipo de referencia)
	369 733 03 151 1	(Referencia)
	Paga su cuota predial a la cual se le asigna el	(Predicado)
	Número de operación	(Tipo de referencia)
	7166077	(Referencia)

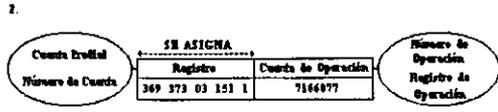
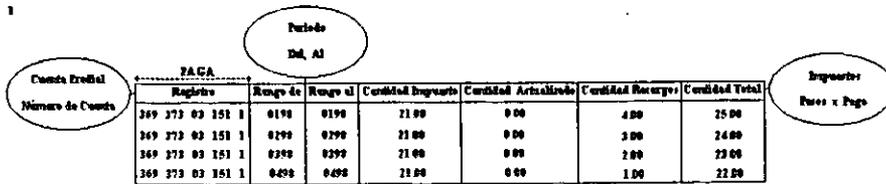
7. Tablas de población.

PAGA							
(-----)							
Cuenta Predial		Periodo	Periodo	Impuesto	Actualizado	Recargos	Total
Número de Cuenta		Del	Al	Pesos e Imp.	Pesos e Actualiza.	Pesos e Recargos	Pesos e Total
Registro		Rango de	Rango al	Cantidad Impuesto	Cantidad Actualizado	Cantidad Recargos	Cantidad Total
369	373 03 151 1	0198	0198	21 00	0 00	4 00	25 00
369	373 03 151 1	0298	0298	21 00	0 00	3 00	24 00
369	373 03 151 1	0398	0398	21 00	0 00	2 00	23 00
369	373 03 151 1	0498	0498	21 00	0 00	1 00	22 00

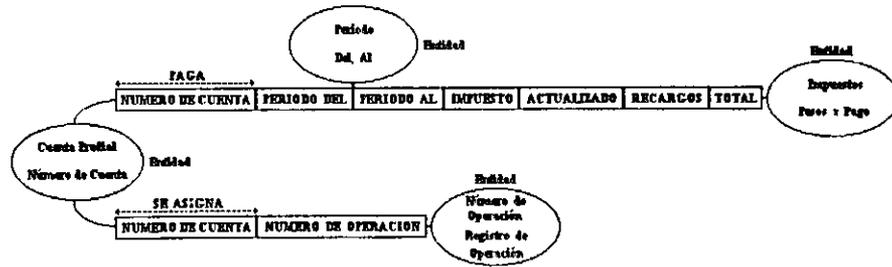
Marca	No. Localización
Bandera de Pago	Posición a Marcar
A Pagar	A Marcar
X	9
X	10
X	11
X	12

SE ASIGNA	
(-----)	
Cuenta Predial	Número de Operación
Número de Cuenta	Registro de Operación
Registro	Cuenta de Operación
369 373 03 151 1	7166077

8. Diagrama ENALIM.

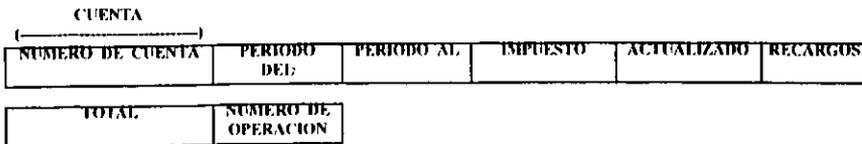


9. Diagrama ENALIM final



10. Diseño de tablas, bases de datos y/o archivos.

- Eliminar campos redundantes.



CUENTA CUENTA PREDIAL TABLA CUENTA PREDIAL

NUMERO DE CUENTA	PERIODO DEL	PERIODO AF	IMPUESTO	ACTUALIZADO	RECARGOS
TOTAL	NUMERO DE OPERACION				

• Estandarizar tipos y nombres de campos así como su longitud.

NOMBRE : (PREDIAL) NOMBRE TECNICO: (T_CUENTA_PREDIAL) - TABLA CUENTA PREDIAL

PK CUENTA	PERINICIAL	PERFINAL	IMPUESTO	ACTUALIZADO	RECARGOS
12 C	4 C	4 C	11,2 N	11,2 N	11,2
TOTAL	NUMOPERA				
13,2 N	6 C				

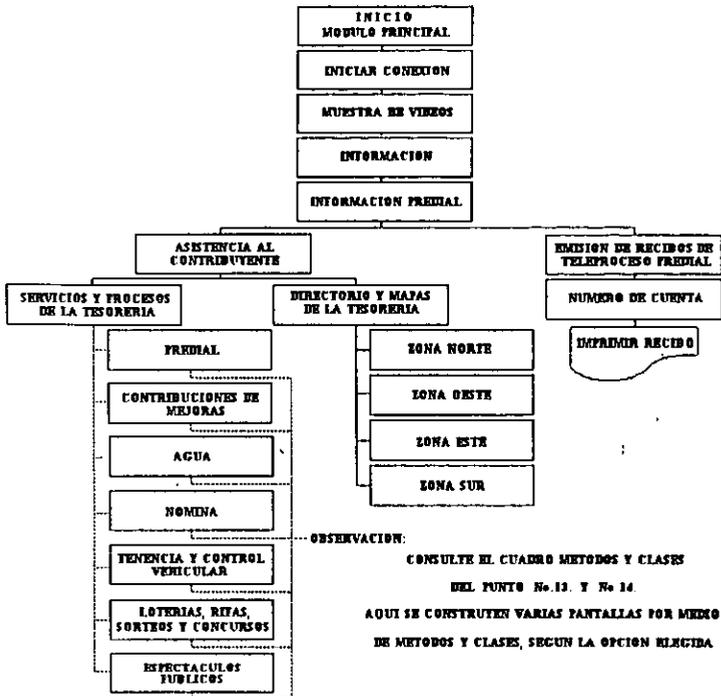
• Diccionario de datos.

SEC	NOMBRE	TIPO Y LONGITUD	DESCRIPCION	OBSERVACIONES
1	CUENTA	X(12)	NUMERO DE CUENTA	CHAR , = 12(9) (1-3, REGIÓN >= 001 & <= 999) (4-6, MANZANA >= 001 & <= 999) (7-8, LOTE >= 01 & <= 99) (9-11, CONDOMINIO >= 001 & <= 999) (12-1, DIGITO VERIFICADOR >= 1 & <= 9)
2	PERINICIAL	X(4)	PERIODO INICIAL	CHAR , = 4(9) (1-2, PERIODO >= 01 & <= 12) (3-2, AÑO >= 01 & <= 99)
3	PERFINAL	X(4)	PERIODO FINAL	CHAR , = 4(9) (1-2, PERIODO >= 01 & <= 12) (3-2, AÑO >= 01 & <= 99)
4	IMPUESTO	9(11,2)	CANTIDAD DE IMPUESTO	> 0,00 Y <= 99999999 99
5	ACTUALIZADO	9(11,2)	CANTIDAD DE ACTUALIZADO	>= 0,00 Y <= 99999999 99
6	RECARGOS	9(11,2)	CANTIDAD DE RECARGOS	> 0,00 Y <= 99999999 99
7	TOTAL	9(13,2)	CANTIDAD TOTAL	> 0,00 Y <= 9999999999 99
8	NUMOPERA	X(6)	NUMERO DE OPERACION	CHAR , = 6(9) , >= 1 & <= 999999

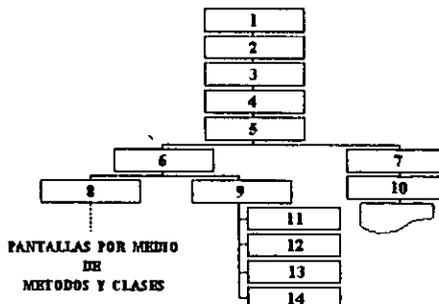
ESQUEMA	TABLA	ALIAS
PAGOS	PREDIAL	SIN ALIAS

NOMECLATURA
2 CAMPO LLAVE
N NUMERICO
A ALFABETICO
X ALFANUMERICO
9 NUMERICO

11. Modulación.

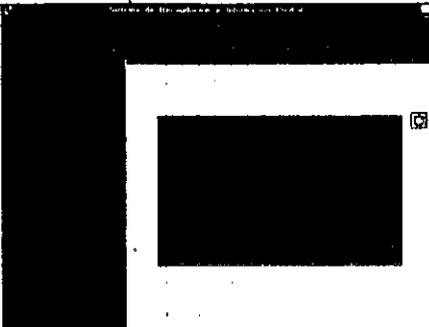


12. Diseño de pantallas.



Se numeran los módulos, para detallar la descripción de cómo fueron diseñadas las pantallas una por una; además se indica a continuación las figuras que componen a las pantallas así como sus botones y algunos detalles de las figuras.

Pantalla No. 1. Inicio. Modulo Principal.	Nombre de Formulario: xPred001.
	Descripción: Pantalla de inicio para la carga de variables globales, asignación de rutas de trabajo, etc. Observaciones: a. Se crea en tiempo de ejecución. b. Se oculta al usuario.
Vista Previa	

Pantalla No. 2. Iniciar Conexión.	Nombre de Formulario: InicPred.
	Descripción: Por medio de instrucciones DDE, se inicia la petición de conexión al host central. Observaciones: a. Se crea en tiempo de diseño.
Vista Previa	

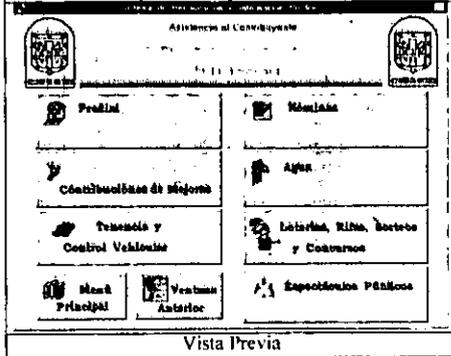
Pantalla No. 3. Muestra de Videos.	Nombre de Formulario: Scren01.
	Descripción: Se muestran dos cuadros con videos de las principales inversiones en la ciudad de México. Observaciones: a. Se crea en tiempo de diseño. b. Tiene un temporizador público: que se activa a los 60 segundos de no usar la pantalla, regresando a esta.
Vista Previa	

<p>Pantalla No. 4. Información.</p>  <p>DEPARTAMENTO DEL DISTRITO FEDERAL SECRETARÍA DE FINANZAS Tesorería</p> <p>SISTEMA INTERACTIVO DE INFORMACIÓN 1998</p> <p>En el menú:</p> <ul style="list-style-type: none"> Consultar información sobre los tributos y servicios que le ofrece la Tesorería Consultar la localización de las oficinas de la Tesorería Consultar cuáles son las deudas de los contribuyentes Consultar el directorio de representantes delegacionales Calcular los impuestos (PROM) <p><i>Para consultar sobre el sistema</i></p> <p>Vista Previa</p>	<p>Nombre de Formulario: Sren02</p> <p>Descripción: Pantalla sólo de índole informativo.</p> <p>Observaciones:</p> <p>a. Se crea en tiempo de diseño.</p>
--	---

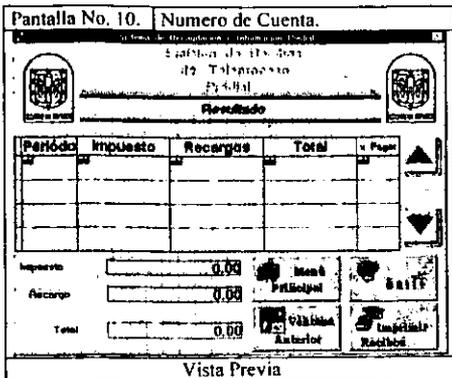
<p>Pantalla No. 5. Información Predial.</p>  <p>SECRETARÍA DE FINANZAS TESORERÍA</p>  <p>ASISTENCIA AL CONTRIBUYENTE</p> <p>EMISIÓN DE RECIBOS DE TELEPROCESO, PREDIAL</p> <p>Basis Ventana Anterior</p> <p>Vista Previa</p>	<p>Nombre de Formulario: Predial.</p> <p>Descripción: Pantalla principal, de aquí parte la navegación a través de las diferentes opciones.</p> <p>Observaciones:</p> <p>a. Se crea en tiempo de diseño.</p> <p>b. El botón de salir solo se empleara en el diseño del sistema, pruebas parciales y pruebas en paralelo.</p>
---	---

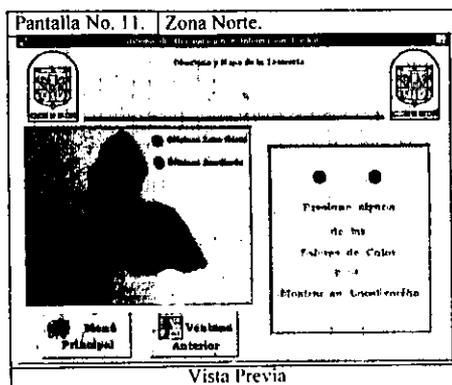
<p>Pantalla No. 6. Asistencia al Contribuyente</p>  <p>ASISTENCIA AL CONTRIBUYENTE</p>  <p>Servicios y Procedos de la Treasoría</p> <p>Directorio y Mapas de la Tesorería</p> <p>Ventana Anterior</p> <p>Vista Previa</p>	<p>Nombre de Formulario: Asiste01</p> <p>Descripción: Módulo de asistencia al contribuyente.</p> <p>Observaciones:</p> <p>a. Se crea en tiempo de diseño.</p>
---	---

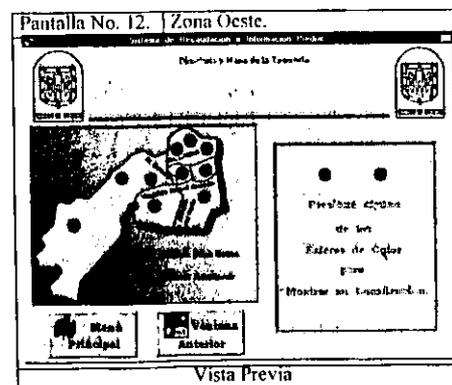
Pantalla No. 7. Emisión de Recibos de Teleproceso Predial.	Nombre de Formulario: Emision01.
	<p>Descripción: Aquí, el contribuyente captura su número de cuenta predial. Si es una cuenta inválida se avisa y rechaza en espera de una correcta. Lo mismo sucede en el caso de no existir tal cuenta.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño. El botón de salir solo se empleara en el diseño del sistema, pruebas parciales y pruebas en paralelo.

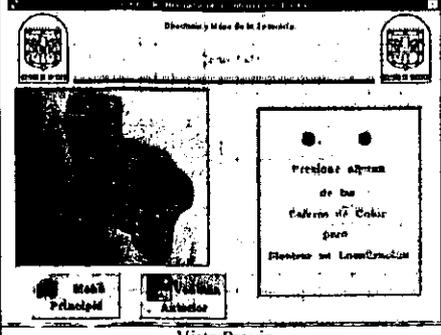
Pantalla No. 8. Servicios y Procesos de la Tesorería.	Nombre de Formulario: Asiste02.
	<p>Descripción: Módulo de información de los diferentes procesos de la Tesorería.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño.

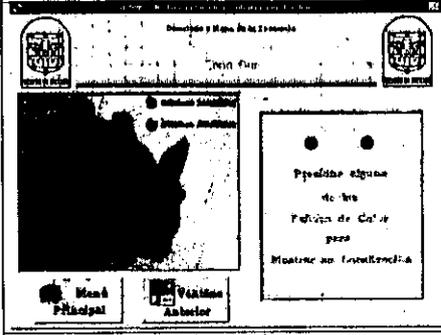
Pantalla No. 9. Directorio y Mapas de la Tesorería.	Nombre de Formulario: Mapas.
	<p>Descripción: Módulo de elección de la zona deseada a buscar alguna ubicación.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño.

<p>Pantalla No. 10. Numero de Cuenta.</p>  <p>Vista Previa</p>	<p>Nombre de Formulario: Emision02.</p> <p>Descripción: Se muestra el total de adeudos que tiene la cuenta solicitada, en donde el contribuyente puede elegir entre los que quiere pagar. Mostrando el total a pagar e imprimir su recibo para pasar al área de cajas y pagar.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño. El botón de salir solo se empleara en el diseño del sistema, pruebas parciales y pruebas en paralelo.
---	---

<p>Pantalla No. 11. Zona Norte.</p>  <p>Vista Previa</p>	<p>Nombre de Formulario: Mapas01.</p> <p>Descripción: Módulo zona norte, en donde puede seleccionar un botón de las diferentes localizaciones y se mostrara un mapa de la ubicación y dirección del punto elegido.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño.
--	--

<p>Pantalla No. 12. Zona Oeste.</p>  <p>Vista Previa</p>	<p>Nombre de Formulario: Mapas02.</p> <p>Descripción: Módulo zona oeste, en donde puede seleccionar un botón de las diferentes localizaciones y se mostrara un mapa de la ubicación y dirección del punto elegido.</p> <p>Observaciones:</p> <ol style="list-style-type: none"> Se crea en tiempo de diseño.
--	--

Pantalla No. 13.	Zona Este.	Nombre de Formulario: Mapas03.
		<p>Descripción: Módulo zona este, en donde puede seleccionar un botón de las diferentes localizaciones y se mostrara un mapa de la ubicación y dirección del punto elegido.</p> <p>Observaciones:</p> <p>a. Se crea en tiempo de diseño.</p>
Vista Previa		

Pantalla No. 14.	Zona Sur.	Nombre de Formulario: Mapas04.
		<p>Descripción: Módulo zona sur, en donde puede seleccionar un botón de las diferentes localizaciones y se mostrara un mapa de la ubicación y dirección del punto elegido.</p> <p>Observaciones:</p> <p>a. Se crea en tiempo de diseño.</p>
Vista Previa		

13. Descripción de programas.

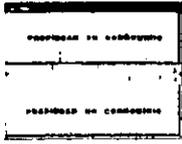
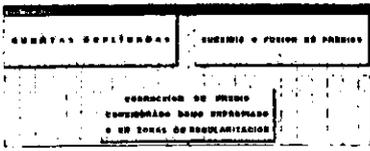
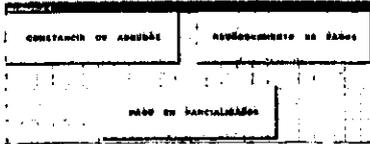
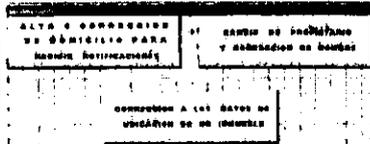
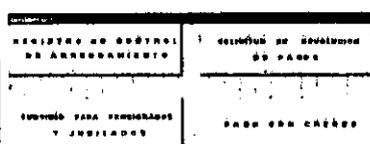
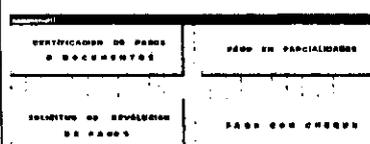
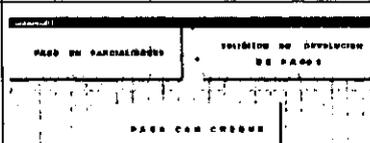
MÉTODOS QUE LLAMAN A LAS CLASES (PANTALLAS) DEL MÓDULO SERVICIOS Y PROCESOS. FORMULARIO ASISTE02	
Método	Código
ActiPred01	THISFORM.AddObject('xNomCLAS','PredMenu01') THISFORM.xNomCLAS.Left = 30 THISFORM.xNomCLAS.Top = 95 THISFORM.xNomCLAS.VISIBLE = .T.
ActiPred02	THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','PredMenu02') THISFORM.xNomCLAS.Left = 177 THISFORM.xNomCLAS.Top = 113 THISFORM.xNomCLAS.VISIBLE = .T.
ActiPred03	THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','PredMenu03') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE = .T.

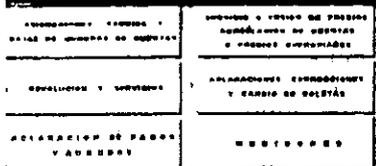
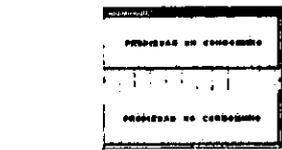
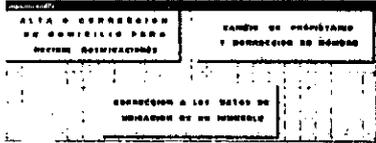
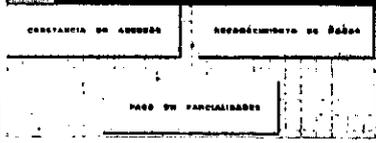
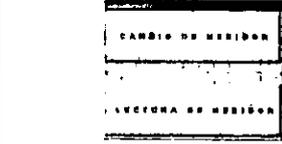
ActiPred04	<pre> THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','PredMenu04') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. </pre>
ActiPred05	<pre> THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','PredMenu05') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. </pre>
ActiPred06	<pre> THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','PredMenu06') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. </pre>
ActiNomi	<pre> xNomCLAS = 'xCLAS'+xNPan THISFORM.AddObject('xNomCLAS','NomiMenu01') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. </pre>
ActiCont	<pre> xNomCLAS = 'xCLAS'+xNPan THISFORM.AddObject('xNomCLAS','ContMenu01') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. </pre>
ActiAgua	<pre> xNomCLAS = 'xCLAS'+xNPan DO CASE CASE xNPan = '01' THISFORM.AddObject('xNomCLAS','AguaMenu01') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 92 THISFORM.xNomCLAS.VISIBLE =.T. CASE xNPan = '02' THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','AguaMenu02') THISFORM.xNomCLAS.Left = 181 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. CASE xNPan = '03' THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','AguaMenu03') THISFORM.xNomCLAS.Left = 181 THISFORM.xNomCLAS.Top = 176 THISFORM.xNomCLAS.VISIBLE =.T. CASE xNPan = '04' THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','AguaMenu04') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. CASE xNPan = '05' THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','AguaMenu05') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE =.T. CASE xNPan = '06' THISFORM.RemoveObject('xNomCLAS') </pre>

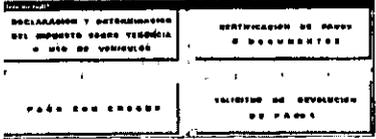
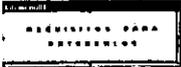
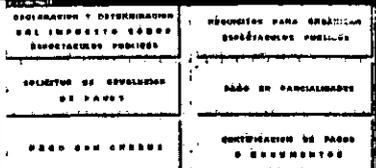
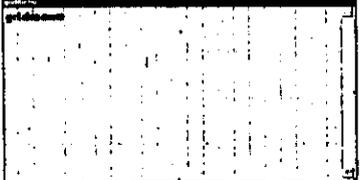
	THISFORM.AddObject('xNomCLAS','AguaMenu06') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE = .T. CASE xNPan = '07' THISFORM.RemoveObject('xNomCLAS') THISFORM.AddObject('xNomCLAS','AguaMenu07') THISFORM.xNomCLAS.Left = 177 THISFORM.xNomCLAS.Top = 113 THISFORM.xNomCLAS.VISIBLE = .T. ENDCASE
ActiTene	xNomCLAS = 'xCLAS'+xNPan THISFORM.AddObject('xNomCLAS','TeneMenu01') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 122 THISFORM.xNomCLAS.VISIBLE = .T.
ActiLote	xNomCLAS = 'xCLAS'+xNPan THISFORM.AddObject('xNomCLAS','LoteMenu01') THISFORM.xNomCLAS.Left = 181 THISFORM.xNomCLAS.Top = 176 THISFORM.xNomCLAS.VISIBLE = .T.
ActiEspe	xNomCLAS = 'xCLAS'+xNPan THISFORM.AddObject('xNomCLAS','EspeMenu01') THISFORM.xNomCLAS.Left = 32 THISFORM.xNomCLAS.Top = 92 THISFORM.xNomCLAS.VISIBLE = .T.
ActiGrid	THISFORM.RemoveObject('xNomCLAS') THISFORM.MasNum(1) THISFORM.AddObject('xNomCLAS','GridMenu') THISFORM.xNomCLAS.Left = 43 THISFORM.xNomCLAS.Top = 81 THISFORM.xNomCLAS.VISIBLE = .T.
HabiBoto	THISFORM.IMAGE4.VISIBLE = .T. THISFORM.IMAGE5.VISIBLE = .T. THISFORM.IMAGE6.VISIBLE = .T. THISFORM.IMAGE7.VISIBLE = .T. THISFORM.IMAGE8.VISIBLE = .T. THISFORM.IMAGE9.VISIBLE = .T. THISFORM.IMAGE10.VISIBLE = .T.
OculBoto	THISFORM.IMAGE4.VISIBLE = .F. THISFORM.IMAGE5.VISIBLE = .F. THISFORM.IMAGE6.VISIBLE = .F. THISFORM.IMAGE7.VISIBLE = .F. THISFORM.IMAGE8.VISIBLE = .F. THISFORM.IMAGE9.VISIBLE = .F. THISFORM.IMAGE10.VISIBLE = .F.

CLASES (PANTALLAS) DEL MÓDULO SERVICIOS Y PROCESOS. FORMULARIO ASISTE02

Clase	Código	Vista Previa
PredMenu01 (1)	THIS.PICTURE = RutaBoto+'PredA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredA01.bmp' THISFORM.LABEL2.CAPTION = 'Asignación de Cuentas' THISFORM.MasNum(1) THISFORM.ActiPred02	

PredMenu02 (2)	THIS.PICTURE RutaBoto+'PredB02.bmp' DO ClicBoto THIS.PICTURE RutaBoto+'PredB01.bmp' THISFORM.LABEL2.CAPTION 'Propiedad en Condominio' GO 1 WAIT WIND 'xNPan: '+xNPan TIME 2 THISFORM.ActiGrid	
PredMenu03 (3)	THIS.PICTURE = RutaBoto+'PredB06.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB05.bmp' THISFORM.LABEL2.CAPTION = 'Cuentas Duplicadas' GO 3 THISFORM.ActiGrid	
PredMenu04 (4)	THIS.PICTURE = RutaBoto+'PredB12.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB11.bmp' THISFORM.LABEL2.CAPTION = 'Constancia de Adeudos' GO 6 THISFORM.ActiGrid	
PredMenu05 (5)	THIS.PICTURE = RutaBoto+'PredB18.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB17.bmp' THISFORM.LABEL2.CAPTION = 'Alta o Corrección de Dominio' GO 9 THISFORM.ActiGrid	
PredMenu06 (6)	THIS.PICTURE = RutaBoto+'PredB24.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB23.bmp' THISFORM.LABEL2.CAPTION = 'Registro de Control de Arrendamiento' GO 12 THISFORM.ActiGrid	
NomiMenu01 (7)	THIS.PICTURE = RutaBoto+'NomiA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'NomiA01.bmp' THISFORM.LABEL2.CAPTION = 'Certificación de Pagos o Documentos' GO 16 THISFORM.ActiGrid	
ContMenu01 (8)	THIS.PICTURE = RutaBoto+'PredB16.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB15.bmp' THISFORM.LABEL2.CAPTION = 'Pago en Parcialidades' GO 20 THISFORM.ActiGrid	

<p>AguaMenu01 (9)</p>	<p>THIS.PICTURE = RutaBoto+'AguaA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'AguaA01.bmp' THISFORM.LABEL2.CAPTION = 'Asignaciones, Cambios y Bajas' XNPar='02' THISFORM.ActiAgua</p>	
<p>AguaMenu02 (10)</p>	<p>THIS.PICTURE = RutaBoto+'PredB02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB01.bmp' THISFORM.LABEL2.CAPTION = 'Propiedad en Condominio' GO 23 THISFORM.ActiGrid</p>	
<p>AguaMenu03 (11)</p>	<p>THIS.PICTURE = RutaBoto+'PredB06.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB05.bmp' THISFORM.LABEL2.CAPTION = 'Cuentas Duplicadas' GO 25 THISFORM.ActiGrid</p>	
<p>AguaMenu04 (12)</p>	<p>THIS.PICTURE = RutaBoto+'PredB26.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB25.bmp' THISFORM.LABEL2.CAPTION = 'Solicitud de Devolución de Pagos' GO 26 THISFORM.ActiGrid</p>	
<p>AguaMenu05 (13)</p>	<p>THIS.PICTURE = RutaBoto+'PredB18.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'PredB17.bmp' THISFORM.LABEL2.CAPTION = 'Alta o Corrección de Domicilio' GO 30 THISFORM.ActiGrid</p>	
<p>AguaMenu06 (14)</p>	<p>THISFORM.IMAGE4.PICTURE = RutaBoto+'PredB12.bmp' DO ClicBoto THISFORM.IMAGE4.PICTURE = RutaBoto+'PredB11.bmp' THISFORM.LABEL2.CAPTION = 'Constancia de Adeudos' GO 33 THISFORM.ActiGrid</p>	
<p>AguaMenu07 (15)</p>	<p>THISFORM.IMAGE4.PICTURE = RutaBoto+'AguaB28.bmp' DO ClicBoto THISFORM.IMAGE4.PICTURE = RutaBoto+'AguaB27.bmp' THISFORM.LABEL2.CAPTION = 'Cambio de Medidor' GO 36 THISFORM.ActiGrid</p>	

TeneMenu01 (16)	THIS.PICTURE = RutaBoto+'TeneA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'TeneA01.bmp' THISFORM.LABEL2.CAPTION = 'Declaración y Determinación del Impuesto' GO 38 THISFORM.ActiGrid	
LoteMenu01 (17)	THIS.PICTURE = RutaBoto+'LoteA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'LoteA01.bmp' THISFORM.LABEL2.CAPTION = 'Requisitos para Obtenerlos' GO 42 THISFORM.ActiGrid	
EspeMenu01 (18)	THIS.PICTURE = RutaBoto+'EspeA02.bmp' DO ClicBoto THIS.PICTURE = RutaBoto+'EspeA01.bmp' THISFORM.LABEL2.CAPTION = 'Declaración y Determinación del Impuesto' GO 43 THISFORM.ActiGrid	
GridMenu (19)	Sin código, ya que tiene de referencia el número de registro desde el método. Tabla de referencia: Datos.	

14. Asignación de programas para captura y aplicación.

Nombre	Módulo	Descripción	Restricciones	Código
(1) Predial Programa	Inicio	Carga el procedimiento de errores generados, variables globales y locales, y rutas de trabajo	Sin restricciones	<pre> SET CURSOR OFF ON ERROR DO ErrorGenerado WITH , !ERROR!, MESSAGE(), MESSAGE(1), PROGRAM(), LINENO() DO Vard'ublAsig DO Inc'ValoSFTs DO FORM Inc'Pred IF nSesion = -1 READ EVENTS ENDIF DO Borra'Var'ub </pre>

Nombre	Módulo	Descripción	Restricciones	Código
2) VariPublAsig Procedimiento	Inicio	Cargar variables locales y globales	Sin restricciones	<pre> PROC VariPublAsig PUBL nSesion, SeCargo3270 PUBL RutaF-xtr3270, RutaBase, RutaBoto, RutaFigu, RutaWavs STOR 0 TO nSesion, SeCargo3270 RutaF-xtr3270 =%ATTACH\FIRMINI\FXE RutaBase = 'C:\Predial6v0\JbcPred\ RutaBoto = 'C:\Predial6v0\JbcPred\ RutaFigu = 'C:\Predial6v0\FigPred\ RutaWavs = 'C:\Predial6v0\WavPred\ *---- Variable de Cuenta Predial PUBL Cad01, Cad02, Cad03, Cad04, CTA1 PUBL xDigVer, DIGVer, AnexarDig, NumReg, xTIME *---- Variables de Suma para Emisión del Recibo PUBL xAdeudoImpuesto, xAdeudoRecargo, xAdeudoTotal </pre>
3) Local:Ma Programa	Mapas	Mensaje en un archivo wav (Toca las esferas dentro del mapa)	Sin restricciones	<pre> SET BELL TO RutaWavs+'Local:Ma wav', 0 ?? CHR(7) </pre>
4) JbcBoto Programa	N formularios	Sonido corto en un archivo wav, indicador de que se pulso un botón.	Sin restricciones	<pre> SET BELL TO RutaWavs+'Ding wav', 0 ?? CHR(7) </pre>
5) IndCade Método	InicPred	Busca una cadena en el número de sesión solicitada, en la pantalla de la emulacion	XContador < 60 segundos	<pre> PARA xSesion, xCndBuscar ValorPosCadena = 0 SeContinuaProg = 1 xContador = SECO() DO WHILE VAL(ValorPosCadena) < 1 =DI:Poket(Sesion,"Search",xCndBuscar) ValorPosCadena =DI:Request(Sesion,"Search") =DI:Execute(Sesion,"[pause(1)]") IF xContador+60 < SECO() SeContinuaProg = 0 EXIT ENDIF ENDIF RETU SeContinuaProg </pre>

Nombre	Módulo	Descripción	Restricciones	Código
(6) FinPrematuro Método	InicPred	Se encarga de cerrar la conexión nSesion, el parámetro xTexto contiene el mensaje del error generado durante la sesión al host	Sin restricciones	<pre> PARA xTEXTO = MESSAGEBOX(xTEXTO,0+16+0,'Favor de Notificar lo Siguiente') SET MESSAGE TO Terminando con la Emulación EXTRA! y la Aplicación Servidor CENTRO' = DDEPoke(nSesion, "Keystroke", "@C") = DDEExecute(nSesion, "[pnuse(6)]") = DDEPoke(nSesion, "Keystroke", "CSSF LOGOFF@E") = FindCade(nSesion, "DIGITE EL SERVICIO DESEADO") = DDEExecute(nSesion, [stop emulator]) = DDETerminate(nSesion) nSesion = -1 RELEASE THISFORM QUIT </pre>
(7) InicSesi3270 Método	InicPred	Manda ejecutar la aplicación de emulación 3270, carga el número de emulación, además contiene los diferentes errores de inicio de conexión y durante la ejecución del verifica que la conexión se sostenga Nota Por motivos de seguridad sólo se presenta una pequeña parte del código total	Sin restricciones	<pre> RUN /N & Ruta\Extra3270 THISFORM.xTimer = DDESetOption('SAFETY', F.) nSesion = THISFORM.Run3270(-1, SECO) IF nSesion < 0 = MESSAGEBOX('Error al Cargar EXTRA! for Windows',0+16+0,'Favor de Notificar lo Siguiente') ELSE SET MESSAGE TO 'Iniciando la Emulación EXTRA!' IF FindCade(nSesion, "DIGITE EL SERVICIO DESEADO") 0 THISFORM.FinPrematuro('No se Cargo la Pantalla DGI') ENDIF THISFORM.RunAntena = DDEPoke(nSesion, "Keystroke", "LOGON PPLID(CICS210)") </pre>
(8) Run3270 Método	InicPred	Verifica en la pantalla de la emulación si se logro la conexión Si es así, se le asigna de forma constante un número de conexión Nota Por seguridad no se indica como se obtiene de forma constante un número de conexión	xRUNCon < 30 segundos	<pre> PARA xRUN3270, xRUNCon (DO WHILE xRUN3270 = -1 IF xRUNCon+30 < SECO) EXIT SetCargo3270 = -1 ENDIF xRUN3270 = DDEInitiate('Extra!', 'B') SetCargo3270 = xRUN3270 ENDDO RETURN SetCargo3270 </pre>

Nombre	Módulo	Descripción	Restricciones	Código
9) digitoVerificador Método	Emision1	Algoritmo para verificar que la cuenta otorgada se correcta	Cuenta: 11 posiciones	<pre> DIME CTA[11] DIME DIG[11] NDigVer= SUBSCTHIFORM CasillaPredial CAPTION.12.1) CTA1 = SUBSCTHIFORM CasillaPredial CAPTION.1.11) FOR i=1 TO 11 CTA[i] = "" CTA[i] = SUBS(CTA1,i,1) ENDFOR FOR N=1 TO 11 STEP 2 DIG[n] = "" DIG[n] = LTRIM(STR(VAL(CTA[n])*2)) IF LEN(DIG[n]) = 2 DIGVer = DIGVer + VAL(LEFT(DIG[n], 1)) + VAL(RIGHT(DIG[n], 1)) ELSE DIGVer = DIGVer + VAL(DIG[n]) ENDF ENDFOR DIGVer=DIGVer+VAL(CTA[2])+VAL(CTA[4])+VAL(CTA[6])+VAL(CTA[8])+VAL(CTA[10]) FOR i=1 TO 10 m = (i * 10) IF (DIGVer > j AND DIGVer <= m) Anexar(Dig = (m - DIGVer) EXIT ENDIF j = (i * 10) ENDFOR </pre>

Nombre	Módulo	Descripción	Restricciones	Código
(10) Remplaza- tos Método	Emision	Localiza los items en la pantalla de presentacion del host. Buscando en una o en N pantallas los items constantes. Como los es numero de cuenta, periodo inicial, periodo final, impuesto, recargos, total	Hay Adeudos = 1 NoPantallas = 1	<pre> [[Línea de Código] HayAdeudos = DDERequest(nSesion, "P9771.2") If V=1 (HayAdeudos) = 0 WAIT WIND/NO EXISTEN ADEUDOS']]SE [[Línea de Código] xCUENTA = " NcAR = LEN(xCUENTA) XCUFNFA = SUBS(xCUENTA,1,NcAR-2) XCUFNFA = STR(xCUENTA,":") =DDEPoke(nSesion, "Cursor", "11") =DDEPoke(nSesion, "Keystroke", "D@E") =DDEExecute (nSesion, "[pause4]") NoPantallas = DDERequest(nSesion, "P558L2") Control = 1 DO WHILE Control <= VAL(NoPantallas) PosA = 722 PosB = 725 PosC = 734 PosI = 741 PosE = 808 PosF = 834 PosG = 847 NcAR = 0 STOR " TO DataA,DataB,DataC,DataD,DataE,DataF, DataG STOR" TO PosAc,PosBc,PosCc,PosDc,PosEc,PosFc, osGc IF Control <= 2 =DDEPoke(nSesion, "Cursor", "11") =DDEPoke(nSesion, "Keystroke", "A@E") =DDEExecute(nSesion, "[pause4]") ENDIF FOR i=1 TO 7 IF i >= 2 PosA = PosA + 160 PosB = PosB + 160 [[Línea de Código] STOR " TO DataA,DataB,DataC,DataD,DataE,DataF,DataG STOR " TO PosAc, PosBc, PosCc, PosDc, PosEc, PosFc, PosGc ENDIF PosAc = 'P' + ALLT(STR(PosA)) + 'L1' DataA = DDERequest(nSesion, PosAc) NcAR = LEN(DataA) DataA = SUBS(DataA,1,NcAR-2) PosBc = 'P' + ALLT(STR(PosB)) + 'L2' DataB = DDERequest(nSesion, PosBc) NcAR = LEN(DataB) DataB = SUBS(DataB,1,NcAR-2) DataC = 'P' + ALLT(STR(PosC)) + 'L4' DataC = DDERequest(nSesion, PosC) NcAR = LEN(DataC) DataC = SUBS(DataC,1,NcAR-2) PosDc = 'P' + ALLT(STR(PosD)) + 'L4' DataD = DDERequest(nSesion, PosDc) NcAR = LEN(DataD) DataD = SUBS(DataD,1,NcAR-2) PosEc = 'P' + ALLT(STR(PosE)) + 'L10' DataE = DDERequest(nSesion, PosEc) PosFc = 'P' + ALLT(STR(PosF)) + 'L12' DataF = DDERequest(nSesion, PosFc) PosGc = 'P' + ALLT(STR(PosG)) + 'L15' DataG = DDERequest(nSesion, PosGc) [[Línea de Código] REPL CUENTA WITH xCUENTA REPL TOTAL WITH VAL(STR(DataG,":")) [[Línea de Código] </pre>

15. Pruebas parciales y en paralelo.

16. Liberación.

17. Mantenimiento.

CONCLUSIONES

Para lograr el inicio, desarrollo y término de este proyecto conté con la ventaja de estar desarrollando sistemas para la Secretaría de Finanzas desde hace algunos años. Conociendo así, el software instalado (emulación, conectividad y lenguaje de programación) y hardware (red, comunicaciones y equipo instalado). Así, se pudo explotar toda esta infraestructura para el diseño y desarrollo del sistema de recaudación e información predial.

En base ha esto, uno de los primeros objetivos se logra, ya que se utilizaron los productos de software y hardware que tiene instalado la Secretaría de Finanzas en su módulo central y sus Administraciones Tributarias Locales, dispersas en cuarenta puntos en el Distrito Federal. Proporcionando beneficios como los siguientes:

- No adquirir, software para desarrollo de sistemas multimedia. Ahorro de dos a cuatro meses por no licitar (competencia entre empresas para vender y/o mostrar su producto). Claro, sin tener en cuenta el tiempo para capacitación para el nuevo producto.
- No contratar o licitar, de forma externa a un despacho para el análisis y diseño del sistema. Un ahorro de dos a cuatro meses.
- En cuestión monetaria, no se gastó para el desarrollo de la interfase alrededor de 1 a 2 millones y medio de pesos de gastos de análisis, desarrollo, e instalación del sistema.
- No tener contratada gente externa (despacho); después de finalizado el proyecto se necesita para el mantenimiento del sistema. Ya que esto causa muchos problemas con el personal de base sindicalizada.

Continuando con la funcionalidad del sistema, se logró un buen diseño de las pantallas, en donde se contienen botones gráficos, letras con un buen tamaño y color, además de un gran colorido en las imágenes gráficas; permitiendo buena funcionalidad y navegación a través de todo el flujo de información. Permitiendo de una manera sencilla trasladarse a consultar los servicios y procesos, localización de las AT's y la emisión de adeudos de predial. La emisión de recibos de predial, se elaboró en tres pasos:

1. Se incorpora en una pantalla con gráficos sencillos con tonos de color café claros, teclas amplias tipo calculadora para incorporar el número de cuenta a consultar, la casilla donde se muestra el número de cuenta es de buen tamaño, teclas que incorporan funciones de borrado parcial o total del número de cuenta y por último botones que permiten salir de la operación actual o de continuar con la siguiente pantalla.
2. La pantalla contiene diferentes ítems, una de ellas muestra gráficamente todos los adeudos que tiene la cuenta solicitada, permitiendo al contribuyente marcar entre los que desea pagar; botón de regresar a pantalla anterior; botón para dirigirse al menú principal; campos para observación de impuestos, recargos y total; botones para navegar a través de los diferentes adeudos.
3. Se permite la emisión del recibo de adeudo predial por el contribuyente.

De lo anterior, permite al contribuyente tener los elementos necesarios de información sobre su estado de cuenta, además de, simplificar los trámites y de una interfase simple y fácil de utilizar.

Hay otro punto que tocar, detrás de la interfase gráfica, se construyeron métodos que contribuyen a la funcionalidad del sistema; esto se debió a proponer el uso del intercambio dinámico de datos (DDE), protocolo de comunicación entre aplicaciones Windows. Donde la programación de funciones DDE y el lenguaje de programación Visual Fox Pro 3.0 fue sencilla de aprender y utilizar. A pesar de esto, el empleo del intercambio dinámico de datos genera los siguientes problemas:

- Deja gran parte de la tarea de compartir la información a cada programa en particular (aplicación cliente).

- Se debe utilizar con cada programa un método diferente para vincular información general mediante DDE y es preciso consultar el manual de programa para conocer las funciones.

También se observó con buenos resultados el utilizar Visual Fox Pro 3.0, como lenguaje de programación ya que contiene instrucciones, propiedades y métodos que permitieron la simulación de una aplicación multimedia, como por ejemplo: poder manipular los sonidos (grabaciones de guiones) de formato wav; incorporar objetos (iconos, figuras, y fondos de pantalla en formato bmp, gif y jpg); propiedades en los objetos de color, color de fondo, tamaño de fuente; y sin dejar a un lado de las bondades de la programación de objetos (clon). A esto, le agregamos un factor externo: no tener licencia renovada de Visual Basic 3.0 (que permitiría el uso de funciones API de Windows, incorporadas en controles VBX y/o OCX pero que de todas maneras tenían que comprarse).

En lo que se refiere a comunicaciones, la respuesta de consulta de adeudos oscila de entre 3 a 10 segundos, esto varía de acuerdo al número de pantallas a consultar ya que se tiene que emplear gran cantidad de líneas de código de las funciones DDE, que aunque no se emplean más de cinco o seis de las doce funciones existentes se tiene que codificar bastantes líneas. Además, influye la respuesta de conexión entre las redes LAN enlazadas por módem local y central, tal respuesta es de 2 a 5 segundos; por último, la respuesta del gateway según carga de trabajo entre milésimas de segundo a 2 segundos. Donde, en el peor de los casos (consulta exhaustiva) se tiene tiempo de respuesta, desde introducir el número de cuanta solicitada y la presentación de la información en pantalla, de 20 segundos.

La respuesta se reducirá en futuro al incorporarse la arquitectura cliente/servidor además de la instalación de fibra óptica en las redes locales y central. El descenso en la tasa de consulta será de 50 %.

Las limitaciones y los alcances del presente estudio las juzgaré como diseñador del sistema; a futuro, si existieran modificaciones en las pantallas de presentación del host en cualquiera de ellas o en su totalidad, se tendrían que cambiar de forma parcial o total las posiciones de búsqueda de los campos solicitados, ya que las funciones DDE hacen referencia a localizaciones de items en las pantallas de presentación. Por lo tanto, el cambio de código en los métodos de las pantallas sería mínimo, y en el peor de los casos, total.

ESTA TESIS NO SALE DE LA BIBLIOTECA

APENDICE A

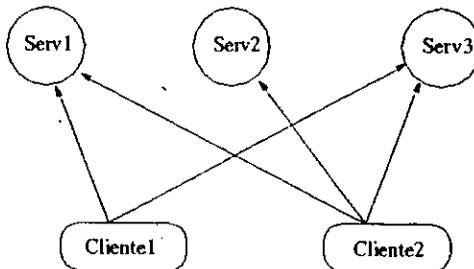
TECNOLOGIA CLIENTE SERVIDOR: RPC

En este tema se exponen las ideas básicas subyacentes al modelo cliente/servidor de construcción de aplicaciones distribuidas. En particular, se expone el modelo de implementación y uso de una tecnología particular ampliamente usada para el diseño de aplicaciones cliente/servidor: La Llamada a Procedimiento Remoto, o RPC (Remote Procedure Call).

El modelo Cliente/Servidor

- División del software de una aplicación en dos conjuntos,
- Servicios.
 - Proveen funciones a programas externos.
 - Pueden ser utilizados por aplicaciones diversas
- Clientes:
 - Actúan de usuarios de los servicios definidos.
 - Implementan realmente las aplicaciones.
 - Actúan como usuarios de los servicios definidos.
- NOTAR: los elementos de una aplicación no colaboran directamente entre ellos, sino a través de los servicios.

Estructura típica



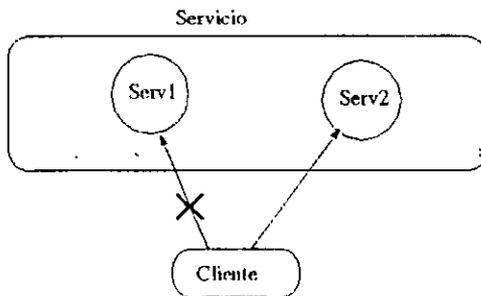
Ejemplos de servicios

- Servidor de ficheros (i.e. NFS, SMB)
Procesos que gestionan unidades de almacenamiento físico.
Los clientes son los sistemas operativos de los nodos en red.
- Servidores de bases de datos.
- Servidores de nombres (servicios de directorio).
- Servidores de tiempo.
- Servidores de seguridad.
- Correo electrónico y servicios de grupos de noticias.
- Servidores de HTTP (WWW).
Servicio = conjunto de servidores.
Servidor = proceso.

Entace de cliente a Servidor (binding)

- ¿Cómo enlazan clientes con servidores?
- Un enlace sucede cuando un cliente que necesita un servicio queda asociado con servidor. particular para ese servicio, comunicándose con él.
- Generalmente incluye varios pasos:
 - Encontrar un nodo donde haya un servidor adecuado.
 - Problema genérico de localización de recursos.
 - En este caso pueden haber más de un recurso (servidor) que puede ser utilizado.
 - Encontrar la forma de comunicarse con el servidor del nodo elegido.
 - Que canal de comunicaciones emplear.
 - Opcionalmente, reenlazar, en caso necesario, para tolerar fallos, o degradación del servicio.

Reenlace



RPC

- El protocolo más común para comunicar clientes y servidores.
- Surgido en 1980 (Xerox PARC).
- Unificación de los procedimientos utilizados en muchas aplicaciones de red.
- Petición -> Espera -> Ejecución -> Respuesta.
- Aparentemente, el procedimiento es el mismo que el utilizado en una llamada a una función o procedimiento.
- Idea atractiva: servicios remotos accesibles con los mismos medios que servicios locales (bibliotecas de funciones).
- Desafortunadamente existen diferencias fundamentales en la semántica data la implementación de la tecnología.
- Es posible conseguir semánticas idénticas, a expensas del rendimiento.
- En términos prácticos, RPC tiene un comportamiento diferente.

Tecnología RPC hoy en día

- Varios Estándares;
- SUN RPC, es el más extendido.
- DCE RPC, es una propuesta alternativa.
- ANSA RPC, el más reciente.
- Todos los estándares se basan en conceptos similares.
- Existen divergencias en aspectos cosméticos (lenguajes de definición de interfaces).
- Los estándares en uso hasta la fecha poseen semántica similar, derivada de su implementación.

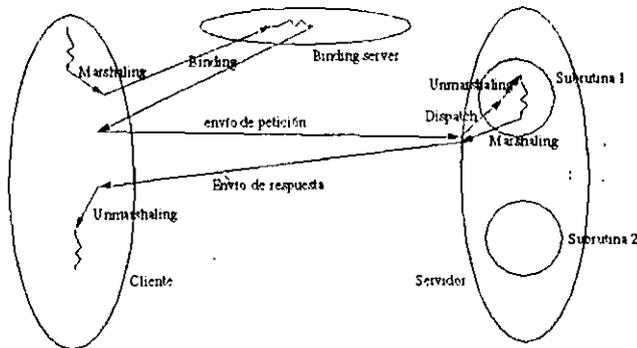
Aplicaciones basadas en RPC

- Como en cualquier lenguaje procedural, hay que definir las subrutinas de un servicio: número y tipo de parámetros que usan.
- El cliente debe utilizar algo para pasar los parámetros necesarios en una llamada, y encapsularlos en un mensaje.
- Utilización de funciones de biblioteca para ello.
- El servidor debe exportar sus subrutinas.
- Utilización de algún tipo de servicio de enlace.
- Especificación del nombre y de la versión del programa o subrutina.
- Codificación de programas muy similar a programas no distribuidos.
- Aspecto diferencial en la separación entre la definición de las interfaces y el código utilizado para su implementación.
- Utilización de una biblioteca para llevar a cabo las funciones básicas.

Funciones básicas de un sistema RPC

- Encapsular los parámetros de una llamada dentro de un mensaje, MARSHALING.
- Enlazar al cliente con el servidor que va a manejar la llamada, BINDING.
- Enviar el mensaje de llamada.
- Encaminar el mensaje de llamada a la función que debe manejarlo, DISPATCHING.
- Marshaling de los parámetros de retorno.
- Envío del mensaje de respuesta.
- Desencapsular los parámetros de una llamada desde el mensaje de invocación, UNMARSHALING

Fases de una RPC



Diferencias con llamadas locales

- La mayoría de las funciones descritas anteriormente están ocultas al programador.
- Realizadas automáticamente por las bibliotecas y sistema de soporte.
- Limitaciones:
- Conjunto limitado de tipos de argumentos que pueden ser pasados.
- Punteros no pueden ser compartidos.
- Limitaciones de tamaños.
- Estructuras de datos basadas en punteros deben ser transformadas (copiadas).
- Modificación de los argumentos pasados está severamente restringida, dependiendo del paquete de RPC.
- Diferencias en rendimiento.
- Fuerzan al programador a considerar diseños extraños.

Introducción de nuevos tipos de errores

- Un sistema de RPC típico introduce nuevos errores que no aparecen en llamadas a subrutinas (excepciones).
- Posibilidad de no encontrar al servidor.
- Posibilidad de que la versión del servidor sea incompatible.
- Posibilidad de que el estado de la red (congestión) produzca el TIMEOUT de la petición.
- Fallos de nodos.
- Los casos anteriores jamás se dan en llamadas a subrutinas locales.
- En general, la forma razonable de tratar lo anterior es como un fallo/parada del servicio en el sistema.
- En algunos casos, la semántica del servicio permite recuperarse del fallo
- Recuperación compleja.
- Posible introducción de inconsistencias.

Lenguaje de Definición de Interfaces (IDL)

- Todo servicio precisa definir qué subrutinas forman parte de él.
- Al conjunto de firmas de las subrutinas que forman parte de un servicio se le llama la interfaz de un servicio.
- Un lenguaje de IDL especifica la sintaxis a utilizar para definir dicha lista de firmas
- Al mismo tiempo el lenguaje indica el tipo de estructuras de datos que pueden ser utilizados como parámetros.
- La estructura del lenguaje indica las capacidades de manejo automático del paso de parámetros.
- Con el fin de controlar la versión del servicio, se suele incluir la especificación del número de versión del servicio.
- Finalmente, la definición incluye el nombre dado al servicio.
- NOTA: no es obligatorio el uso de IDL. Un servidor/cliente RPC puede acceder directamente a la biblioteca de soporte.

Uso del IDL: generación automática de stubs

- El IDL es procesado por un compilador de IDL.
- El compilador retorna stubs y definiciones de estructuras de datos para (C).
- También retorna cabeceras para las subrutinas a definir en el servidor.
- Existe un stub para el cliente y uno para el servidor.
- Un stub de cliente es una subrutina que toma los argumentos formales de un procedimiento remoto y se encarga del:
 - Marshaling.
 - Efectuar el binding.
 - Ejecutar el envío del mensaje.
 - Esperar la respuesta.
 - Efectuar el unmarshaling de los parámetros de retorno.
 - Devolver control al código de cliente.

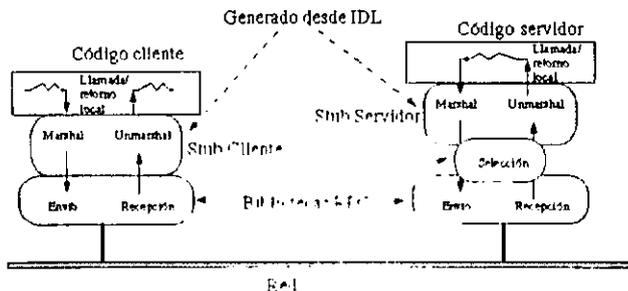
Stubs de Servidor

- El stub de servidor en un sistema RPC típico es más simple que el de cliente.
- Puede ser visto como la imagen especular del stub del cliente.
- Recoge el mensaje de petición de servicio.
- Comprueba la compatibilidad de la versión.
- Efectúa el unmarshal de los parámetros.
- Efectúa la invocación local de la subrutina del servidor.
- Efectúa el marshaling de los parámetros de retorno.
- Envía el mensaje de retorno a través del canal de comunicación establecido al recibir el mensaje de invocación.

Definición de estructuras de datos

- En IDL, las estructuras de datos se definen con un lenguaje especial que refleja las restricciones de paso de parámetros.
- El compilador traduce dichas definiciones a estructuras de datos del lenguaje de programación (típicamente C, o C++).
- Dichas traducciones quedan reflejadas en ficheros de cabecera que deben ser utilizados por los programas cliente y servidor.
- Al mismo tiempo el compilador genera subrutinas que sirven para realizar el marshaling y unmarshaling de las estructuras de datos definidas.
- Dichas subrutinas son adoptadas por los stubs para realizar el marshaling/unmarshaling.
- Las subrutinas emplean funciones de biblioteca especiales para llevar a cabo su función.

Diagrama de componentes



Binding en RPC

- Problema general, relacionado con los servicios de directorio.
- En RPC, es preciso hacer llegar una llamada de un cliente a una función a un servidor.
- Aspectos que debe cubrir un servicio de binding:
- Encontrar el nodo más apropiado para el servicio.
- Establecer la compatibilidad del servicio pedido con el ofertado.
- Establecer un camino de comunicación entre el cliente y el servicio.
- Establecer un camino de acceso desde el cliente hasta la función del servidor.
- En ocasiones, el servicio de binding puede comenzar la ejecución del proceso servidor.
- Operación cara en términos de tiempo empleado.
- Aunque tan sólo ocurre una vez por par cliente/servidor

Binding para el Servidor

- La descripción anterior desde el punto de vista del cliente.
- Desde el punto de vista del servidor, el binder debe:
- Registrar el ejecutable del servidor, asociándolo con el nombre del servicio.
- Dependiendo del sistema concreto, registrar el número de puerto a utilizar por el servidor.
- En algunos casos establecer un enlace de comunicaciones entre el cliente y el servidor.
- La última parte del binder corresponde al dispatcher. El dispatcher suele estar integrado en el servidor, a través de las funciones de biblioteca.

Binding: implementación

- El binder es el servicio del cual todos los demás dependen.
- Suele tener incorporado algún esquema de soporte de fallos.
- I.e. replicación, almacenamiento de datos en memoria estable.
- Generalmente uno por nodo que accede a servicios RPC.
- Encontrar el binder debe ser independiente del servicio de binder mismo.
- Generalmente disponible en una dirección bien conocida
- Nodo local.
- Puerto estándar.

Consideraciones acerca del Marshaling

- La representación de números enteros depende de la CPU utilizada.
- Alineamiento de datos: en algunas arquitecturas, los datos deben alinearse en 32 (64) bits.
- La representación de números en coma flotante puede diferir.
- Los punteros, por sí mismos, no son transmisibles. En algunos casos, cuando el tipo apuntado esté claro, será posible efectuar una copia por recorrido.
- Notar que esto dificulta la semántica de paso por referencia, tan común en llamadas a subrutinas en lenguajes procedurales.
- La aproximación tomada es el copy-in, copy-out, con características totalmente diferentes en sistemas concurrentes.
- Incompatibilidades en los tipos básicos de datos.
- I.e. pasar enteros desde representaciones de 64 a 32 bits.

Estrategias de implementación de RPC

- Debido a la naturaleza de las comunicaciones (pérdida de mensajes) varias estrategias son posibles:
- Acción frente a falta de respuesta.
- Reenviar la petición.
- Acción frente a la recepción de peticiones duplicadas.
- Filtrar las peticiones.
- Acción frente a la necesidad de reenviar una respuesta.
- Guardar una historia de las peticiones servidas y su respuesta para evitar reejecutar la petición.

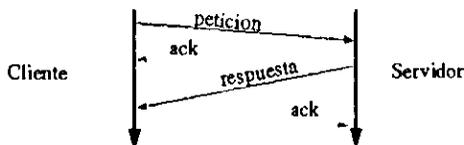
Semántica de RPC

- De acuerdo con las estrategias vistas, aparecen las siguientes posibles semánticas de RPC:
- Semántica quizás.
- Sin garantías. Si no se ha recibido una respuesta tras un TIMEOUT, no se sabe nada de lo sucedido con la petición.
- Semántica al menos una vez:
- Se retransmiten las peticiones hasta que se recibe una respuesta, pero el servidor no filtra las peticiones.
- Si el servidor falla, el cliente es advertido mediante una excepción.
- Semántica máximo una vez:
- Se retransmite hasta recibir una respuesta.

- Se filtran las peticiones descartando las duplicadas.
- Si el servidor falla, el cliente es advertido con una excepción.

El protocolo RPC

- Objetivo: imitar el funcionamiento de una llamada a procedimiento local.
- Caso 1: Sistema totalmente fiable (comunicaciones y nodos).
- Implementación trivial.
- Los únicos problemas son los derivados del paso de parámetros y de la velocidad de invocación.
- Caso 2: Nodos totalmente fiables.
- Relativamente simple.



El protocolo RPC: Cont

- Caso 3: Nodos también pueden fallar.
- Cuando la comunicación entre dos procesos no ocurre puede ser debido a:
- Fallo del receptor.
- Fallo de comunicaciones.
- Ambos.
- Difícil asegurar la semántica de exactamente una vez.
- Necesario técnicas de alta disponibilidad y replicación.
- Lo mejor que se puede hacer es máximo una vez..

Estructuración del servidor

- Una cuestión importante es cómo organizar un servidor.
- Cuando se recibe una petición externa existen básicamente dos alternativas:
- Event-driven.
- Sólo una tarea.
- Cuando es necesario esperar a alguna condición es imprescindible crear estructuras de datos para almacenar el estado intermedio.
- Para llamadas simples puede ser lo más adecuado.
- Para llamadas complejas, se hace difícil llevar la cuenta de todo el contexto asociado a una petición.
- Puede dejar el servidor fuera de servicio ante peticiones costosas.
- Multi-thread.

Multi-thread

- Cada petición es servida por una tarea diferente.
- No es necesario crear estructuras de datos para almacenar el contexto de una petición cuando se hace necesario esperar.
- El contexto lo almacena la propia tarea.
- Cuando hay que esperar, la tarea simplemente se bloquea sin indisponer al servidor.
- Inconvenientes.
- El mantenimiento de las tareas (scheduling) tiene un overhead apreciable.
- Es necesario añadir código para control de concurrencia en acceso a datos compartidos del servidor.
- CONCLUSION: uso de multitarea, con uso ocasional de modelo de eventos.

Ejemplo: La RPC de SUN

- La más ampliamente utilizada.
- Todo sistema UNIX la implementa para NFS.
- Semántica de llamada: al menos una vez.
- IDL:
- Se utiliza una extensión del XDR (eXtensal Data Representation).
- Rpgen es el compilador.
- Sólo un parámetro en cada llamada, pero puede ser una estructura arbitraria.
- Permite la definición de funciones de marshaling por parte del usuario (definidor de un servicio).
- Permite el marshaling de estructuras arbitrarias.
- Existen funciones básicas de biblioteca para los tipos primitivos.

Sun RPC: Binding

- Utilización de un proceso simple en cada nodo con un servidor: el portmapper.
- Cada servicio es identificado por un número, una versión.
- El cliente debe conocer el nodo donde reside el servidor.
- El portmapper provee el puerto donde el servidor escucha la llegada de peticiones.
- La petición de bind es `clnt_create`, que crea el camino de comunicación (bien UDP, bien TCP).

RPC's asíncronas

- Semántica parecida a la RPC normal, pero no se espera respuesta.
- No existen parámetros de salida (o de vuelta).
- El cliente no queda bloqueado esperando una respuesta.
- Variación:
- Se esperan parámetros de salida.
- El cliente se sincroniza más tarde para recoger las respuestas de los servidores.

RPC's ligeras

- Frecuentemente, cliente y servidor se encuentran en el mismo nodo.
- Es posible reducir el número de copias de los parámetros.
- En una RPC normal, se efectúan al menos 2 llamadas al sistema (envío de mensaje y recepción).
- Al menos 4 copias de los parámetros:
- Marshaling.
- Paso al kernel para construir mensaje.
- Recepción de los parámetros en el espacio del servidor.
- Unmarshal.
- LRPC:
- Uso de memoria compartida entre cliente/servidor.
- Uso de la misma tarea para evitar `context-switch` y llamadas al sistema.

Contexto en Interacciones C/S

- En muchas ocasiones es necesario realizar varias operaciones para conseguir un único fin.
- Ejemplos: Reserva de viajes, Pago electrónico, etc.
Aplicaciones de tipo transaccional.
- Sistemas de registro y notificación (login), sistemas de comercio electrónico.
- En tales casos es necesario establecer un contexto que relacione las diversas operaciones entre sí, permitiendo al servidor considerarlas en su conjunto.
- Necesidades:
- Identificar el contexto a través de diversas invocaciones.
- Asociar estado en el servidor para cada invocación.
- Recuperarse del fallo del cliente y reconocer contextos caducados.

Statefull: Características

- Contexto compartido entre Cliente y Servidor.
- Cliente y/o servidor pueden realizar acciones bajo el supuesto que dicho estado es correcto.
- El servidor debe llevar cuenta de los clientes existentes y de la información que del servidor tienen.
- Cuando cambia la información, debe ejecutarse un protocolo de invalidación.
- El cliente puede ser informado tan pronto como se produce un cambio en el servidor.
- Los clientes deben comunicar explícitamente con el servidor cuando abandonan una conversación.
- Puede utilizarse para transferir información de forma coherente a otros servidores, y disminuir la carga existente sobre cada uno de ellos.

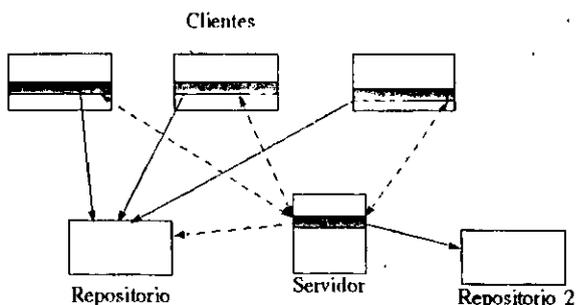
Stateless: Características I

- Un Servidor no controla cuantos ni qué clientes tiene.
- Un servidor no da ninguna garantía más allá de la ejecución correcta de una petición de servicio.
- Un servidor puede cambiar su información sin necesidad de comunicarse con los clientes.
- Cada invocación del cliente hace referencia tan sólo a sus parámetros explícitos.
- Cliente y Servidor pueden compartir de forma implícita información: el cliente puede almacenar resultados previos del servidor.
- Frecuente uso de caching por parte del cliente, con el fin de eliminar la necesidad de acceder frecuentemente al servidor.
- Para poder usar caching, es necesario tener una forma de detectar cuando la información copiada es inválida.

Stateless: Características II

- La aproximación stateless es la mejor entendida, más simple y más extendida.
- Muchos sistemas de ficheros están basados en esta aproximación.
- El sistema WWW está basado en esta aproximación.
- Sin embargo, existe una forma de conseguir identificar una interacción de un cliente por parte del servidor: cookies.
- Es posible usar la aproximación anterior en un contexto más general.
- El sistema RPC no ofrece en sí mismo, ningún mecanismo para intercambiar estado entre cliente y servidor.
- No es posible identificar dentro del esquema RPC a ninguna entidad del servidor y/o cliente.
- Fallos de los clientes pueden ser ignorados por el servidor.

Ejemplo



Ejemplo (Stateless)

- Cuando el servidor cambia su estado.
- El servidor no comunica con los clientes, pero marca la información anterior como inválida.
- Cuando un cliente trata de utilizar su información
- Detecta que la información es inválida al tratar de usarla.

- Contacta al servidor de nuevo, para refrescar el valor de su información.
- Reinicia la operación basada en la información anteriormente inválida.
- Si el coste de acceso del cliente a la información es alto, o existe alguna garantía en tiempo real que el cliente deba satisfacer, la aproximación no es buena, ya que añade el tiempo de refresco a la latencia de la invocación original.
- Pistas (Hints).

Ejemplo (statefull)

- Cuando el servidor cambia de estado.
- Antes de ello, bloquea el acceso de nuevos clientes al estado.
- Recorre la lista de clientes, potencialmente informando del nuevo estado (al menos invalidando).
- Desbloquea el acceso al nuevo estado.
- Cuando el cliente usa su información, ésta es válida.
- No necesita detectar mala información, pues tiene asegurada la validez de la información anterior.
- En el peor de los casos, sabe que la información es inválida antes de usarla, y puede refrescarla en ese momento.
- Puede haber un proceso asíncrono que vaya refrescando la información inválida para que el cliente la tenga válida cuando acceda.

Ejemplo (NFS)

- Ejemplo de sistema de fichero distribuido stateless:



- El servidor usa vnodes para los ficheros (con ID unico)
- Un cliente obtiene el ID de un Vnode cuando abre un fichero
- El cliente almacena una cache del fichero
- Cuando lee el cliente usa la cache local
- Cuando escribe, se escribe la cache al servidor

NFS (Cont)

- En el nodo cliente, no hay inconsistencias.
- Cuando hay un close, se sincroniza la cache con el servidor.
- Pueden haber inconsistencias entre procesos de diferentes clientes.
- Escenario
- El nodo 1 abre el fichero A y accede varios bloques.
- El nodo 2 abre el fichero A, y modifica algunos de los bloques accedidos por Nodo 1.
- El nodo 1 no se dará cuenta de que el fichero ha cambiado hasta que intente acceder a bloques no accedidos hasta el momento.

APENDICE B

RECIBO EMITIDO DE ADEUDO PREDIAL



ESTADO DE MÉXICO

IMPUESTO PREDIAL

RECIBO 1 DE 1

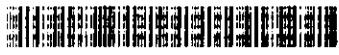
CUENTA : 369 733 03 151
 FRANCISCO CESAR MORALES Y CARR.
 COLONIA : JAL. EL SALADO
 CDD. POST : 39140
 MUNICIPIO : BENITO JUAREZ

USO RAO TIPO 05 CLASE 4
 M2. SUELO 37 M2S 54.80
 VALOR TOTAL SUELO 2,707.50
 M2. CONSTRUCCION 68 M2S 340.50
 VALOR DE LA CONST. 25,043.63
 VALOR CATASTRAL 25,071.25
 TOTAL RENTAS MENS. 21.00
 VAL CAT BASE STA.
 IMP70. BIMESTRAL 22.00

PROPUESTA DE DECLARACION DE ACUERDO A LOS ARTS. 30, 149, 151, 152 Y 153 DEL C.O.F.I.P. FINANCIERO DEL D.F.

PERIODO			VENCIMIENTO			IMPUESTO	ACTUALIZADO	REDARGOS	TOTAL
DEL	AL	AÑO	DIA	MES	AÑO				
16	1	98				21.00	4.00	25.00	
16	2	98				21.00	3.00	24.00	
16	3	98				21.00	3.00	24.00	
16	4	98				21.00	1.00	22.00	
TOTALES						84.00	11.00	95.00	

LA DECLARACION SE PRESENTA CONFORME A LOS ARTICULOS 17 Y 18 DEL CODIGO FINANCIERO DEL D.F. DE FUNDAMENTO DE EL ARTICULO 43 FRACCION I, ARTS. 17 Y 18 DEL PROPIO CODIGO SE DECLARA QUE NO EXISTE ADEUDO PREDIAL QUE LOS DATOS ENTABADOS SON CIERTOS



01369733031500000947166070

EMITIDO EN: CENTRAL

FECHA DE EMISION
06 10 98

NUMERO DE OPERACION
716607-7

NOMBRE Y FIRMA DEL CONTRIBUYENTE
PARA EL CONTRIBUYENTE

GLOSOARIO

Acceso remoto	Pertenece a la comunicación con una computadora a través de una terminal distante de la computadora.
ACF	Función Avanzada de Comunicaciones (Advanced Communications Function). Nombre oficial de la línea de productos para programas SNA de IBM, tales como VTAM (ACF/VTAM), NCP (ACF/NCP), etc.
ADE	Ambiente de desarrollo de aplicaciones (Application Development Environmet). Diseño de IBM para el desarrollo de aplicaciones que puedan ejecutarse en todos los ambientes SAA. El software de desarrollo se basa en clientes y servidores; las funciones principales residen en la computadora central.
Almacenamiento en disco	Grabación de información en placas magnéticas en rotación continua. El almacenamiento puede ser secuencial o de acceso aleatorio
Análisis y diseño de sistemas	Examen de un problema y creación de una solución. El análisis de sistemas es efectivo cuando se pueden examinar todos los lados del problema. El diseño de sistemas es lo más efectivo cuando puede proponerse más de una solución, por ejemplo, una alternativa manual, o una segunda alternativa automatizada.
Aplicador	Es un sistema de banda ancha, es un dispositivo para elevar las señales de radio a un nivel necesitado por otros dispositivos del sistema.
ASCII	American National Standar Code For Information Interchange. Es un código de paridad de 7 bits establecido por American National Standar Institute para lograr compatibilidad entre servicios de datos y consta de 96 caracteres mayúsculos y minúsculos visibles en la pantalla y 32 caracteres de control no visibles.
AUI	Interfase de la unidad de enlace. El cable, conectores y circuitos de transmisión que utilizan para interconectar el sustrato de señalización física (PLS) y la MAU.
Base de datos	Conjunto no redundante de elementos de datos interrelacionados procesables por una o más aplicaciones.
Batch processing	Procesamiento por lote. El procesamiento de un grupo de transacciones. Las transacciones se reúnen y se procesan frente a los archivos maestros (con actualización de los archivos maestros) al final del día o en algún otro periodo de tiempo.
BMP	Mapa de Bits (Bit Map). Formato de gráficos Windows que se puede desarrollar de forma independiente o dependiente.
BPI	Bits por segundo.
BPS	Bits por segundo.
BSC	Binary Synchronous Communication.
Buffer	Pequeña área de almacenamiento.
Bus	Línea de comunicación de datos entre los componentes de una computadora.
Byte	Equivale a 8 bits y un bit sólo puede tener dos valores '0' ó '1'.
Cabezas	Pequeños electrones capaces de leer o escribir ceros o unos, ya sea en la superficie de un disco o de una cinta magnética.
Cable coaxial	Conductores electromagnéticos separados por un material dieléctrico. Dicho cable, tiene como función la transformación de señales en forma de pulsos eléctricos.

Cable duplex trenzado	Dos alambres de un circuito de señalización trenzados entre sí para minimizar los efectos de la inductancia.
Cache	Memoria que trabaja a la velocidad del CPU.
Canal	Procesador de E/S. Se encarga de interactuar con los dispositivos periféricos con el fin de liberar al procesador central de dichas operaciones.
CCITT	Consulative Committee Internacional Telegraph and Telephone. Organización establecida por E.U. para desarrollar estándares mundiales de tecnología de comunicaciones; por ejemplo, protocolos que utilizarán dispositivos que intercambian datos.
CCU	Central Control Unit..
Circuiteria	Conjunto de componentes electrónicos conectados para ejecutar una función especial..
Cliente/Servidor	El cliente es la máquina solicitante y el servidor es la máquina proveedora. Esto implica que existe un software.
Colisión	Múltiples transmisiones concurrentes en el cable, generando datos desorganizados.
Conectividad	Es una LAN, posibilidad de cualquier dispositivo conectado al sistema de distribución de establecer una sesión con cualquier otro dispositivo.
Conector BNC	Conector de cable coaxial de la serie BNC de 50 ohms del tipo que se encuentran comunmente en equipo de televisión y video para el consumidor.
Conexión	Punto de acceso, con un conector adecuado, a un medio de comunicación.
CPS	Columnas por segundo.
CPU	Unidad central de proceso.
CSMA/CD	Método de acceso a redes para manejar colisiones de paquetes de datos.
CTRL	Control.
CUI	Interfaces de usuarios que se basan en caracteres o en texto, como el DOS, que presenta datos en el modo estándar de texto de 25 líneas y 80 columnas.
Chip	Componente electrónico de tamaño muy reducido.
DASD	Direct Access Storgere Device.
DAT	Dynamic Address Translation.
Data streaming	Facilidad de hardware que permite transferir datos a una velocidad de 3 M B/S.
DDE	Intercambio dinámico de datos (Dynamic Data Exchange). Protocolo de mensajes de Microsoft Windows que permite que los programas de aplicación pidan e intercambien los datos automáticamente. Una programa en una ventana puede interrogar a un programa en otra ventana utilizando el protocolo DDE.
Disco de Winchester	Medios de almacenamiento en discos duros magnéticos alojados en contenedores sellados. No todos los discos sellados son Winchester.
Duplex	Indica una transmisión independiente en dos direcciones, que tiene lugar en ambos sentidos simultáneamente.
E/S	Entrada/Salida.
EBCDIC	Extended Binary Coded Decimal Interchange Code.
Emulador	Sistema de software/hardware para microcoputadoras que permiten a la microcomputadora comportarse como alguna terminal específica, como una DEC VT100 o una IBM 3270/79
Estación	Dispositivo físico que se puede enlazar a una LAN con medios comparativos con el fin de transmitir y recibir información en ese medio compartido
Ethernet	LAN y su protocolo asociado producido por (pero no limitado) Xerox. Ethernet es un sistema de banda base.
FCB	Format Control Block.
FEP	Dispositivo de comunicaciones que se utiliza para entrada de datos en un sistema de computación. EL FEP ofrece en general puertos asincrónicos o sincrónicos, o ambos, para el sistema.
Floppy disk	Disco flexible ó diskette.
Gateway	Pasarela, puerta de acceso. Una computadora que conecta dos tipos diferentes de redes de comunicaciones. Realiza la conversión de protocolos de una red a otra. Por ejemplo, una puerta de acceso podría conectar una red LAN de computadoras.
GCR	Group Coded Recording.
GUI	Graphical User Interface. Interfaz gráfica de usuario basada en gráficos que incorpora iconos, menús enrollables y un ratón. Windows.
Half duplex	Término que denota una transmisión independiente alterna, que se produce en un sólo sentido cada vez.
handle	Nombre o número temporal asignado a un archivo, tipo de letra u otro objeto. Por ejemplo, un sistema operativo puede asignar un número temporal a cada archivo que él abra como forma de

	identificación y de rastreo del mismo.
Hardware	Se refiere a los componentes físicos de la computadora y accesorios periféricos.
Host	Anfitrión. La computadora central o la computadora controladora en un entorno de procesamiento en tiempo compartido o distribuido.
Hotkey	Tecla caliente. Una tecla seleccionada o una combinación de teclas que producen que alguna función se ejecute en la computadora, sin importar qué otra cosa se está ejecutando en ese momento, las "teclas calientes" se usan comúnmente para activar un programa residente en memoria (TSR).
IBG	Inter Block Gap. Separación entre los datos, sobre la superficie del carrete.
Icono	Una diminuta representación pictórica de un objeto, tal como una aplicación, archivo o unidad de disco, que se utiliza en interfaces gráficas de usuario (GUI).
IEEE	The Institute of Electrical and Electronic Engineers, Inc.
IN/S	Pulgadas/Segundo
Interface	Frontera compartida entre elementos del sistema; definida por interconexiones físicas comunes, señales y significados de señales intercambiadas.
ISO/OSI	Internacional Standards Organization Open Systems Interface. Modelo de red de siete estratos.
K	Kilo, equivalente a mil unidades (1000).
K B/S	Kilo bytes/Segundo.
L/E	Lectura/escritura.
LAN	Red de Area Local (Local Area Network). Red de comunicaciones que sirve a usuarios dentro de un área geográficamente limitada.
Logout	Salir del sistema, desconexión. Igual que logoff.
LPI	Líneas por pulgada
LPM	Líneas por minuto.
LPS	Líneas por segundo.
LU	Unidad Lógica (Logical Unit). En SNA, un final de una sesión de comunicaciones. La sesión completa LU a LU se define como tipo de sesión.
LU 6.2	También llamado APPC (Advanced Program-to-Program Communications) Comunicaciones Avanzadas de Programa a Programa. Sesión SNA que proporciona comunicaciones entre dos programas de aplicaciones.
M	Mega, equivalente a un millón de unidades (1000000)
Macro	Una pequeña rutina, o escritura, que automatiza operaciones normalmente activadas seleccionando menus o introduciendo una orden cada vez.
Mainframe	Macrocomputadora. Una computadora grande. A mediados de los años 60, las épocas antiguas de las computadoras, todas las computadoras eran mainframes (literalmente "bastidor principal"), ya que el término se refería al gabinete que contenía la CPU. Aunque mainframe aún significa gabinete principal, usualmente se refiere a un gran sistema de computación.
MAN	Red de Area Metropolitana (Metropolitan Area Network). Red de comunicaciones que abarca un área geográfica como una ciudad o un suburbio.
MAU	Unidad de conexión al medio. Porción del estrato físico entre la MDI y AUI que interconecta el cable troncal al cable ramal y contienen los componentes electrónicas que envían, reciben y manejan las señales codificadas impresas y recuperadas del cable troncal.
MAU	Unidad de acceso a múltiples estaciones (Multi-station Access Unit). Núcleo central en una red de área local de tipo anillo de señales.
MDI	Interfase dependiente del medio o distante. Interfase mecánica y eléctrica entre el medio de cable troncal y la MAU.
Mensaje	Un conjunto de datos que se transmite en una línea de comunicaciones.
Mhz	Megahertz. Un millón de ciclos por segundo. A menudo es usado con referencia a la velocidad del reloj de una computadora, lo cual es una medida bruta de su velocidad de procesamiento interno.
Micro to mainframe	Micro a computadora. Interconexión de PC a macrocomputadoras. Puede hacerse que una computadora personal emule a una terminal de macrocomputadora con una tarjeta de expansión, como una IRMAboard, y tener una sesión interactiva con la macrocomputadora. Los datos también pueden ser transferidos desde la macrocomputadora a la computadora personal para su análisis.
Módem	Modulador-Demodulador. Un dispositivo que adapta una terminal o computadora a una línea telefónica. Convierte los pulsos digitales de la computadora a frecuencias dentro del rango de audio del teléfono y los vuelve a convertir en pulsos en el lado receptor.
Módulo	Los módulos de programas se diseñan para manejar una tarea específica dentro de un programa

	mayor.
Multimedia	La comunicación de información en más de una forma; incluye el uso de texto, audio, gráficos, animación de gráficos y vídeo de pleno movimiento.
Nodo	Cualquier estación, terminal, computadora u otro dispositivo en una red de computadoras.
Nodo	En comunicaciones, un punto de empalme o conexión en una red (una terminal o una computadora).
NS	Nano Segundo, equivalente a un mili segundo.
PDU	Unidad de datos de protocolo. Secuencia de octetos contiguos que se entregan como una unidad. Una PDU de LLC válida mide cuando menos 3 catetos de largo y contiene 2 campos de dirección y un campo de control. Una PDU puede o no incluir además un campo de información. Phase Encoded.
PC	Equipo de cómputo externo a la CPU que realiza diversas funciones de entrada y salida.
Periférico	Técnica mediante la cual se interroga periódicamente a cada uno de los terminales que comparten una línea de comunicaciones, con el fin de averiguar si necesita utilizar la línea.
Poleo	Conjunto de instrucciones escritas en un lenguaje de programación que utilizan para definir una operación o conjunto de operaciones para una computadora.
Programa	Conjunto de instrucciones que son recibidas/enviadas por el canal del CPU, un ejemplo de ellos son: lectura, escritura, iniciar, impresión, parar, etc.
Programa de canal	Conjunto formal de convenciones que rigen el formato y la sincronización relativa de intercambio de mensajes en una red de comunicaciones.
Protocolo	Secuencia de intercambio de mensajes entre dos entidades del mismo estrato; utilizan los servicios de los estratos contiguos para efectuar la óptima transferencia de datos y/o información de control de un sitio a otro.
Protocolo principal	Unidad física (Physical Unit). En SNA, el software responsable de la administración de los recursos de un nodo, tales como los enlaces de datos. Una PU soporta una conexión con el anfitrión (SSCP) para recolectar estadísticas de administración de la red.
PU	Hardware y software necesario para dos redes que emplean una misma tecnología, para comunicarse; en forma más específica. Hardware y software necesarios para enlazar segmentos de las mismas redes o similares en el estrato de enlace de datos del modelo de referencia OSI; puente o nivel de MAC.
Puente de enlace	Random Access Memory. Tipo de memoria electrónica, donde el acceso es al azar.
RAM	Red de computadoras.
Red	Red de computadoras con un nodo de procesamiento central a través del cual circulan todos los datos y comunicaciones.
Red centralizada	Red de computadoras y comunicaciones que cubren una área geográfica limitada, que permite que todos los nodos se comuniquen con todos los otros nodos, y no requieren un nodo o procesador central.
Red de área local	Una o más computadoras enlazadas con usuarios o entre sí vía una red de comunicaciones.
Red de computadora	Red total de dispositivos y medios de transmisión (radio, cables, etc.) necesarios para transmitir y recibir inteligencia.
Red de comunicaciones	Configuración de una red en que todas las parejas de nodos están conectadas directamente a través de vías de acceso redundantes, a través de nodos intermedios.
Red distribuida	Unidad básica de datos que son leídos o grabados por un comando de entrada/salida en la computadora.
Registro físico	Conjunto de elementos independientes de su entorno físico. Porciones del mismo registro lógico pueden estar ubicados en diferentes registros físicos.
Registro lógico	Dispositivo que se utiliza para extender la longitud, topología o interconectividad del medio físico más allá de las especificaciones impuestas por un solo segmento, hasta la máxima longitud aceptable de la línea de transmisión troncal en una sesión de comunicación.
Repetidor	Introducción de trabajo remoto o distante. Entrada de un trabajo de lote desde un sitio remoto y recepción de la salida vía una impresora en línea u otro dispositivo situado en lugar diferente.
RJE	Synchronous Data Link Control.
SDLS	En una red, computadora que es compartida por múltiples usuarios.
Servidor	Conexión activa entre un usuario y una computadora o entre dos computadoras.
Sesión	Página de memoria (virtual o real) equivalente a 4 bytes.
SLOT	System Network Architecture. Arquitectura de redes de sistemas. Principal estrategia de IBM para el uso de redes, introducida en 1974. La SNA está compuesta por una variedad de productos de hardware y software que interactúan todos entre sí.
SNA	Punto de control de servicios de sistema (System Services Control Point). Programa de control
SSCP	

Swap buffer	de una red SNA. Reside en el anfitrión, y es un componente del software VTAM.
Thread	Area de almacenamiento que auxilia a la memoria universal char set.
Token Ring Network	Hilo, camino. Una transacción o mensaje en un sistema multihilo.
Topologia	Red de área local creada por IBM que utiliza un cable especial de alambres trenzados y el método de acceso por paso de señales, transmitiendo a 4 ó 16 Mbits por segundo.
Touch screen	En una red de comunicaciones, el patrón de interconexión entre nodos.
Vías de acceso	Pantalla táctil. Pantalla de visualización sensible al tacto que utiliza un panel claro sobre la superficie de la pantalla. El panel es una matriz de celdas que transmite información de presión al software.
VTAM	Hardware y software necesario para hacer que dos redes tecnológicamente diferentes se comuniquen entre sí; una vía de acceso ofrece conversión de protocolo de una arquitectura de red a otra y puede, por lo tanto, utilizar los siete estratos del modelo de referencia OSI.
WAN	Método virtual de acceso a telecomunicaciones (Virtual Telecommunications Access Method). También llamado ACF/VTAM (Advanced Communications Function/VTAM). Software que controla comunicaciones en un ambiente SNA de IBM.
	Red de Area Ancha (Wide Area Network). Red de comunicaciones que abarca áreas geográficas amplias, como pueden ser estados y países.

BIBLIOGRAFIA

1. **ADMINISTERING NETWARE 5.**
Dorothy L. Cady.
McGraw-Hill Professional Publishing; (September 29, 1999); 600 pp.
2. **CONCEPCIÓN Y DISEÑO DE BASES DE DATOS DEL MODELO E/R AL MODELO RELACIONAL.**
Mario G. Piattini.
Addison Wesley-Rama. 1998. 1040 pp.
3. **DATABASE DESIGN FOR MERE MORTALS: A HANDS-ON GUIDE TO RELATIONAL DATABASE DESIGN.**
Michael J. Hernandez.
Addison-Wesley Pub Co; (January 1997); 480 pp.
4. **DISEÑO DE BASES DE DATOS.**
Wiederhold, Gio.
McGraw-Hill; 2a edición (3a edición en español), México; (Septiembre, 1996).
5. **DISEÑO Y USO DE BASES DE DATOS RELACIONALES.**
Irene Luque, Miguel A. Gómez-Nieto.
Rama. 1998. 480 pp.
6. **EL GUION MULTIMEDIA.**
Guillem Bou Bouzá.
Anaya Multimedia, México (Enero, 1999). 416 pp.
7. **ELEMENTOS Y HERRAMIENTAS EN EL DESARROLLO DE SISTEMAS DE INFORMACION.**
Mario G. Piattini, Sunil N. Daryanani.
Addison Wesley-Rama. 1999. 528 pp.
8. **ESTRATEGIAS CLENTE/SERVIDOR PARA EL SIGLO XXI.**
David Shimer, Lisa-Ann Barnes.
Prentice Hall Hispanoamericana S.A.; México (1998); 222 pp.

9. **GETTING STARTED IN MULTIMEDIA DESIGN.**
Gary Olsen.
North Light Books; First edition (March 1997); 144 pp.
10. **GUIA COMPLETA DE MULTIMEDIA.**
Kindersley D.
Ediciones B Argentina S.A. 1999. 405 pp.
11. **INGENIERIA DE SISTEMAS.**
Eduardo Arbones Malisani.
Marcombo. 1998. 162 pp.
12. **INTRODUCCIÓN A LOS SERVICIOS DEL DIRECTORIO NETWARE.**
Novell, Inc.
USA; (March, 1997).
13. **LAN TIMES. GUIA DE REDES DE ÁREA EXTENSA (INTERMEDIO).**
Parnell Tere.
Mc Graw-Hill. 1999. 504 pp.
14. **MCSE: IMPLEMENTING AND SUPPORTING MICROSOFT SNA SERVER 4.0.**
Javier L. Mariscal.
Prentice Hall; First edition (August 27, 1999); 642 pp.
15. **MICROSOFT SNA SERVER 4.0 RESOURCE GUIDE (IT-RESOURCE KIT).**
Microsoft Corporation (Editor).
Microsoft Press; Bk&Cd Rom edition (September 1999); 752 pp.
16. **MODERN SYSTEMS ANALYSIS AND DESIGN.**
Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich.
Addison-Wesley Pub Co (Net); 2nd edition (November 1998); 912 pp.
17. **MULTIMEDIA. EDICIÓN ESPECIAL.**
Cañizares Funcia, Carmen.
Anaya Multimedia, México (Junio, 1999). 192 pp.
18. **MULTIMEDIA: MAKING IT WORK.**
Tay Vaughan.
Osborne McGraw-Hill; 4th Bk&cdr edition (July 8, 1998); 581 pp.
19. **MULTIMEDIA-BASED INSTRUCTIONAL DESIGN: COMPUTER-BASED TRAINING, WEB-BASED TRAINING, AND DISTANCE LEARNING.**
William W. Lee, Diana L. Owens.
Jossey-Bass; Bk/CD-ROM edition (April 2000); 304 pp.
20. **SOFTWARE DEVELOPMENTS.**
Net/Master Systems. Basic Operations V2.2.

Developments International Pty; Sydney, Australia; (November, 1997).

1. **TECNOLOGIAS DE INTERCONECTIVIDAD DE REDES.**
Merilee Ford, H. Kim Lew, Steve Spanier, y Tim Stevenson.
Prentice Hall Hispanoamericana S.A.; México (1998); 489 pp.
2. **USER'S MANUAL. MICROSOFT VISUAL FOX PRO RELEASE 3.0.**
Microsoft Corporation.
USA; (Marzo 1997); 823 pp.