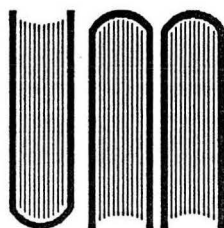


UNIVERSIDAD DE MONTERREY

DIVISIÓN DE PROGRAMAS DE TECNOLOGÍA Y ADMINISTRACIÓN



UNIVERSIDAD
DE MONTERREY

040.0016
G2455i
2000

Sistema de Tarificación de la Red de Voz de H.E.B. México

PROGRAMA DE EVALUACIÓN FINAL QUE PRESENTA:

Héctor Manuel Garza Garza

en opción al título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

San Pedro Garza García, N.L.

Mayo, 2000

403590

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

DEDICATORIAS

A Dios, por ser mi guía y compañero durante el transcurso de mi vida y por haberme ayudado a cumplir una meta más.

A mis padres y hermanos, mi más sincero agradecimiento por su apoyo, confianza y cariño que me han entregado durante toda mi vida.

A Erika, por ser parte de la inspiración para cumplir con esta meta.

AGRADECIMIENTOS

A mi Asesor, Ing. Ma. del Consuelo Jiménez de González, por apoyo, confianza y dedicación brindadas para el desarrollo de este proyecto.

A la empresa HEB, al Ing. Macedonio Garza y al Ing. William Mejía por todas las facilidades prestadas y por haberme brindado la oportunidad de realizar este proyecto. En especial agradezco al Ing. Salvador Hernández por su interés y apoyo prestado para la realización de este proyecto.

A mis Amigos, por su apoyo y buenos deseos, en especial al grupo de los 6+1+otro por su especial amistad. Así mismo, agradezco a todas aquellas personas que han confiado en mí.

PRÓLOGO

El uso de la tecnología en las organizaciones ya es muy común, no sólo si se trata de automatizar los procesos de la cadena de valor de la organización sino de lograr la ventaja competitiva por todos buscada.

El hecho es que la tecnología de información que ayuda a solucionar problemas u aprovechar áreas de oportunidad conlleva a ser más eficientes los procesos de la empresa obteniendo grandes beneficios, gracias a que la información puede ser administrada por sus usuarios de la forma como cada quien lo requiere, para una oportuna toma de decisiones.

Y en este proceso de decisiones en donde el ahorro de ciertos gastos que la empresa tiene que realizar como serían las llamadas que los empleados de una organización realizan para mantenerse en contacto con proveedores, o entre los mismos empleados desempeñando su labor día con día, también es posible que la tecnología pueda ayudar a supervisar e informar el mal uso de las llamadas realizadas en la empresa.

Tal es el caso de HEB que a través de su infraestructura de comunicaciones y apoyada por los conmutadores y el software que este utiliza y a través del uso de plataformas en visual basic y vbscript para ambiente Intranet es posible apoyar procesos de administración del conmutador, recolección de llamadas, administración de solicitudes de nuevas extensiones y explotación de información entre otros.

Este proyecto que aquí se presenta será de gran ayuda para la empresa permitiendo en primera instancia, conocer el estado de sus gastos por concepto de llamadas de manera detallada por departamento, información que ayudará no sólo para determinar si lo indicado por Teléfonos de México es lo correcto sino tomar acciones en un futuro con cada empleado por el mal uso de su extensión de telefónica.

Por otra parte si la tecnología ayuda a reportar esta información, también podrá ayudar a satisfacer de manera más rápida las necesidades de los empleados o departamentos de la empresa en relación a poseer una extensión.

Y es en esta parte donde el proyecto representa un mayor reto debido al conocimiento que se requiere para aprender a programar el conmutador y poder actualizar y cambiar información de las extensiones o activar y desactivarlas.

Asimismo debido a la flexibilidad que tiene el sistema en un futuro se podrá agregar la funcionalidad que permita tener mejor control de presupuestos a asignarse por departamento por concepto de uso de teléfono, debido al conocimiento exacto al día de los gastos por este concepto.

Por consiguiente puedo concluir que el uso de las tecnologías de red y en ambiente Intranet, además del uso de una metodología de desarrollo de aplicaciones en estos ambientes, han permitido que su desarrollador Héctor Garza haya terminado en el tiempo previsto o planeado con excelentes resultados, tanto en el sistema como en su documentación. Mis más sinceras felicitaciones a Héctor por su dedicación, organización y gran creatividad para el desarrollo de este proyecto.

Ing. Ma. Del Consuelo Jiménez de Gzz.
Asesor del proyecto
Mayo 2000

ÍNDICE

INTRODUCCIÓN	1
1. LA EMPRESA	3
1.1. <i>Historia</i>	3
1.1.1. El principio	3
1.1.2. H.E.B. en la actualidad	4
1.1.3. H.E.B. en México	5
1.1.4. H.E.B. y la comunidad	6
1.2. <i>Misión, visión y valores de H.E.B. México</i>	7
1.2.1. Misión	7
1.2.2. Visión	7
1.2.3. Valores	7
1.3. <i>Productos y servicios</i>	8
1.4. <i>Estructura organizacional</i>	9
1.4.1. Organigrama de la dirección de administración y finanzas	10
1.4.2. Organigrama de la subdirección de contraloría	11
1.4.3. Organigrama de la subdirección de sistemas	12
1.5. <i>Infraestructura computacional</i>	14
2. JUSTIFICACIÓN DEL PROYECTO	16
2.1. <i>Situación actual</i>	16
2.1.1. Antecedentes	16
2.1.2. Descripción del proceso actual	18
2.1.3. Áreas de Oportunidad	20
2.2. <i>Solución propuesta</i>	20
2.2.1. Objetivo general del proyecto	23
2.2.2. Objetivos particulares	23
2.2.3. Alcances	24
2.2.4. Análisis costo-beneficio	25
2.3. <i>Requerimientos de hardware y software</i>	26
2.3.1. Software	26
2.3.2. Hardware	26
3. MARCO TEÓRICO	28
3.1. <i>PBX Ericsson Consono MD-110</i>	28
3.1.1. Descripción general	28
3.1.2. Lenguaje	30
3.1.3. Formato del registro de la llamada terminada	37
3.2. <i>Desarrollo Rápido de Aplicaciones</i>	38
3.2.1. Definición	38
3.2.2. Fases del RAD	38
3.2.3. Beneficios del RAD	42
3.2.4. La técnica de prototipo	43
3.3. <i>Active Server Pages</i>	44

3.3.1.	Las Páginas Activas de Servidor (ASP)	44
3.3.2.	Sintaxis de las ASP	45
3.4.	<i>NetCharts</i>	46
3.4.1.	Características	46
3.4.2.	Beneficios	46
3.4.3.	Programación	47
3.4.4.	Integración	47
3.4.5.	Especificaciones	48
3.5.	<i>Visual Basic vs. Java</i>	49
3.5.1.	Java	49
3.5.2.	Visual Basic	50
3.5.3.	Comparación de Java y Visual Basic	51
4.	ANÁLISIS	55
4.1.	<i>Usuarios clave</i>	55
4.2.	<i>Expectativas de los usuarios</i>	55
4.3.	<i>Objetivos y Alcances</i>	57
4.4.	<i>Funciones del sistema</i>	57
4.5.	<i>Procesos del Sistema</i>	62
4.5.1.	Descomposición de Procesos	62
4.5.2.	Dependencia de procesos	72
5.	DISEÑO	82
5.1.	<i>Arquitectura</i>	82
5.2.	<i>Dependencia entre procesos y entidades</i>	85
5.3.	<i>Modelo entidad-relación</i>	96
5.4.	<i>Estándares</i>	97
5.4.1.	Estándares de diseño en Intranet	97
5.4.2.	Estándares de diseño en Visual Basic	100
5.5.	<i>Jerarquía de formas</i>	102
5.6.	<i>Descripción de tablas</i>	105
5.7.	<i>Diagramas de Acción</i>	110
5.8.	<i>Prototipos</i>	128
5.8.1.	Recopilación de Llamadas	128
5.8.2.	Perfiles y Usuarios	128
5.8.3.	Utilerías	129
5.8.4.	Depuración de Base de Datos	130
5.8.5.	Generación de Solicitudes de Servicio	131
5.8.6.	Reportes	131
5.8.7.	Conciliación	132
5.8.8.	Administración del Conmutador	133
6.	CONSTRUCCIÓN	135
6.1.	<i>Estándares</i>	135
6.1.1.	Estándares de Programación en Intranet	135

6.1.2.	Estándares de Programación en Visual Basic	139
6.1.3.	Estándares de Programación en SQL Server	142
6.2.	<i>Construcción del backend</i>	146
6.3.	<i>Procesos de mayor complejidad</i>	148
6.3.1.	Actualización de Información en el Conmutador	148
6.3.2.	Recopilación de Llamadas	160
6.4.	<i>Problemas y soluciones</i>	167
6.5.	<i>Resumen</i>	168
6.5.1.	Elementos utilizados en Visual Basic	168
6.5.2.	Elementos utilizados en ASP's	169
7.	LIBERACIÓN	173
7.1.	<i>Pruebas en ambiente Intranet</i>	173
7.2.	<i>Pruebas en ambiente Visual Basic</i>	191
7.3.	<i>Sinopsis de pruebas</i>	194
	CONCLUSIÓN	195
	TRABAJOS FUTUROS	198
	BIBLIOGRAFÍA DE REFERENCIA	200
	BIBLIOGRAFÍA GENERAL	202
	ANEXO A – INTERFACES	205
	ANEXO B – CÓDIGO DEL SISTEMA	222
	ANEXO C – REFERENCIAS DE INTERNET	311

ÍNDICE DE TABLAS

Tabla 1.1. Departamentos de las tiendas HEB México	9
Tabla 1.2. Infraestructura computacional de HEB México	15
Tabla 2.1. Costos de Desarrollo	25
Tabla 2.2. Costos de llamadas locales	25
Tabla 2.3. Costos de llamadas a celulares	25
Tabla 3.1. Formato del registro de la llamada terminada	37
Tabla 3.2. Comparación de características de Java y VB	54
Tabla 4.1. Usuarios clave involucrados en el sistema	55
Tabla 4.2. Funciones del sistema	58
Tabla 4.3. Prioridad de las funciones del sistema	59
Tabla 4.4. Beneficios de las funciones del sistema	61
Tabla 4.5. Relación de Entidades y Procesos	81
Tabla 5.1. Estándares de interfases para Intranet	100
Tabla 5.2. Elementos y características para un workspace	101
Tabla 5.3. Niveles de Interfaces de Visual Basic	102
Tabla 5.4. Niveles de Interfaces de Intranet	102
Tabla 6.1. Prefijos para variables en VB	140
Tabla 6.2. Prefijos especiales para variables en VB	140
Tabla 6.3. Calificadores para variables en VB	140
Tabla 6.4. Prefijos para controles de VB	142
Tabla 6.5. Resumen de construcción en VB	168
Tabla 6.6. Resumen de construcción en ASP's	172
Tabla 7.1. Sinopsis de pruebas	194

ÍNDICE DE FIGURAS

Figura 1.1. Organigrama de la empresa	9
Figura 1.2. Organigrama de la dirección de administración y finanzas	10
Figura 1.3. Organigrama de la subdirección de contraloría	11
Figura 1.4. Organigrama del departamento de cuentas por pagar	12
Figura 1.5. Organigrama de la subdirección de sistemas	12
Figura 1.6. Organigrama de la gerencia de servicios de cómputo	13
Figura 2.1. Red de voz de H.E.B. México	17
Figura 2.2. Procesos actuales involucrados con el proyecto	18
Figura 2.3. Procesos de la solución propuesta	22
Tabla 2.4. Requerimientos de Software	26
Figura 3.1. Módulos del PBX Ericsson Consono MD110	29
Figura 3.2. Fases del Desarrollo Rápido de Aplicaciones (RAD)	39
Figura 3.3. Pasos del JRP	40
Figura 3.4. Pasos del JAD	41
Figura 3.5. Etapas de la técnica de prototipo	43
Figura 3.6. Modelo de las Active Server Pages	45
Figura 3.7. Ejemplo de gráfica con NetCharts	48
Figura 4.1. Descomposición de procesos (Sistema)	62
Figura 4.2. Descomposición de procesos (Administración Conmutador)	62
Figura 4.3. Descomposición de procesos (Act. Inf. del Conmutador)	63
Figura 4.4. Descomposición de procesos (Administración de Solicitudes)	64
Figura 4.5. Descomposición de procesos (Conciliación de Llamadas)	64
Figura 4.6. Descomposición de procesos (Generación Solicitud)	65
Figura 4.7. Descomposición de procesos (Recopilación de Llamadas)	66
Figura 4.8. Descomposición de procesos (Reportes)	66
Figura 4.9. Descomposición de procesos (Configuración del Sistema)	67
Figura 4.10. Descomposición de procesos (Depuración BD)	67
Figura 4.11. Descomposición de procesos (Perfiles y Usuarios)	68
Figura 4.12. Descomposición de procesos (Act. de Teléfonos Permitidos)	69
Figura 4.13. Descomposición de procesos (Act. Cat. Conmutadores)	69
Figura 4.14. Descomposición de procesos (Act. Cat. Tipo Extensión)	69
Figura 4.15. Descomposición de procesos (Act. Cat. Tipo Llamadas)	70
Figura 4.16. Descomposición de procesos (Act. Cat. Tipo Llamadas)	70
Figura 4.17. Descomposición de procesos (Act. Cat. Tipo Solicitud)	70
Figura 4.18. Descomposición de procesos (Act. Cat. Áreas Genéricas)	71
Figura 4.19. Dependencia de procesos (Acceso)	72

Figura 4.20. Dependencia de procesos (Recopilación de Llamadas)	72
Figura 4.21. Dependencia de procesos (Generación Solicitud)	73
Figura 4.22. Dependencia de procesos (Act. Inf. del Conmutador)	73
Figura 4.23. Dependencia de procesos (Administración de Solicitudes)	74
Figura 4.24. Dependencia de procesos (Depuración BD)	75
Figura 4.25. Dependencia de procesos (Conciliación de Llamadas)	75
Figura 4.26. Dependencia de procesos (Reportes)	76
Figura 4.27. Dependencia de Procesos (Perfiles y Usuarios)	76
Figura 4.28. Dependencia de Procesos (Act. de Teléfonos Permitidos)	77
Figura 4.29. Dependencia de Procesos (Act. Cat. Conmutadores)	77
Figura 4.30. Dependencia de Procesos (Act. Cat. Tipo Extensión)	78
Figura 4.31. Dependencia de Procesos (Act. Cat. Tipo Llamadas)	78
Figura 4.32. Dependencia de Procesos (Act. Cat. Claves de LD)	78
Figura 4.33. Dependencia de Procesos (Act. Cat. Tipo Solicitud)	79
Figura 4.34. Dependencia de Procesos (Act. Cat. Áreas Genéricas)	79
Figura 5.1. Arquitectura del Sistema	84
Figura 5.2. Relación Procesos y Entidades (Acceso)	85
Figura 5.3. Relación Procesos y Entidades (Recopilación de Llamadas)	85
Figura 5.4. Relación Procesos y Entidades (Alm. datos de Archivo Local)	86
Figura 5.5. Relación Procesos y Entidades (Generación Solicitud)	86
Figura 5.6. Relación Procesos y Entidades (Act. Inf. Conmutador)	87
Figura 5.7. Relación Procesos y Entidades (Act. Rel. Socio-Extensión)	87
Figura 5.8. Relación Procesos y Entidades (Admon. de Solicitudes)	88
Figura 5.9. Relación Procesos y Entidades (Act. Inf. en Conmutador)	88
Figura 5.10. Relación Procesos y Entidades (Depuración BD)	88
Figura 5.11. Relación Procesos y Entidades (Conciliación de Llamadas)	89
Figura 5.12. Relación Procesos y Entidades (Conciliación Manual)	89
Figura 5.13. Relación Procesos y Entidades (Reportes)	90
Figura 5.14. Relación Procesos y Entidades (Actualización de Usuarios)	91
Figura 5.15. Relación Procesos y Entidades (Actualización de Perfiles)	91
Figura 5.16. Relación Procesos y Entidades (Asoc. Perfiles y Usuarios)	92
Figura 5.17. Relación Procesos y Entidades (Act. Teléfonos Permitidos)	92
Figura 5.18. Relación Procesos y Entidades (Act. Cat. Conmutadores)	93
Figura 5.19. Relación Procesos y Entidades (Act. Cat. Tipo Extensión)	93
Figura 5.20. Relación Procesos y Entidades (Act. Cat. Tipo Llamadas)	94
Figura 5.21. Relación Procesos y Entidades (Act. Cat. Claves de LD)	94
Figura 5.22. Relación Procesos y Entidades (Act. Cat. Tipo de Solicitud)	95
Figura 5.23. Relación Procesos y Entidades (Act. Cat. Áreas Genéricas)	95

Figura 5.24. Diagrama de entidad-relación del sistema.	96
Figura 5.25. Diagrama de entidad-relación para la seguridad del sistema.	96
Figura 5.26. Ejemplo de encabezado de reporte.	97
Figura 5.27. Ejemplo de cuerpo del reporte.	97
Figura 5.28. Ejemplo de reporte.	98
Figura 5.29. Ejemplo de Actualizador (pantalla inicial).	99
Figura 5.30. Ejemplo de Actualizador (pantalla consulta).	99
Figura 5.31. Ejemplo clásico de workspace.	101
Figura 5.32. Jerarquía de formas en Visual Basic.	103
Figura 5.33. Jerarquía de Formas de Intranet.	104

RESUMEN

El cálculo de gastos y la elaboración de presupuestos es un proceso contable que se realiza día a día en las empresas actualmente ayudando así a elaborar la planeación estratégica para el cumplimiento de las metas y visiones.

El presente proyecto apoya al cálculo de gastos de servicios telefónicos y a mantener actualizada y centralizada la información de la red de voz de la organización. Asimismo, apoya a la accesibilidad de los empleados a este tipo de servicios permitiendo administrar el conmutador mediante procesos e interfaces sencillas de operar.

El proyecto consiste en el desarrollo de un sistema computacional utilizando tecnología cliente/servidor a través de herramientas como ASP's, Visual Basic y SQL Server.

INTRODUCCIÓN

En los últimos años, el desarrollo de sistemas en ambientes Web, como Internet e Intranet, han tenido mucha popularidad. La mayoría de las grandes empresas cuentan o están desarrollando un site en Internet y/o una Intranet. Las facilidades que otorga este ambiente de desarrollo son amplias y altamente prometedoras, como el fácil acceso a los sistemas sin tener que realizar una instalación previa, el uso de tecnología cliente/servidor, el constante crecimiento de esta tecnología y el acceso a nuevos componentes y nuevas herramientas que van haciendo más sencillo el desarrollo de sistemas en este ambiente.

Considerando este tipo de tecnologías y tomando en cuenta los beneficios que representa es posible aprovechar el área de oportunidad en relación al manejo de los servicios de telefonía en la empresa HEB. Estos servicios de telefonía administrados por el conmutador representan un costo para la empresa y por esto se vuelve una necesidad obtener a detalle la información de cómo se generan estos costos para poder tomar decisiones adecuadas.

Tomando en cuenta lo anterior y considerando la necesidad del departamento de cuentas por pagar y el de comunicaciones para centralizar la información de la red de voz de la empresa y poder recolectar y explotar la información de gastos de las extensiones de la red, se ha desarrollado el presente proyecto para la empresa Supermercados Internacionales HEB. El desarrollo de este proyecto se basa en la creación de un sistema para explotar la información de las llamadas externas de la empresa y la administración de los conmutadores de esta.

El proyecto esta relacionado con el desarrollo de un proceso de recolección de llamadas, que recibe información del conmutador de la empresa y la almacena en una base de datos, además de un conjunto de reportes gráficos y de texto que permiten analizar la información recopilada y por último un módulo que permite administrar el conmutador y mantener la información de las extensiones centralizada. Esto permitirá distribuir mejor los costos entre los departamentos, obtener presupuestos más exactos y confiables y tener una visión real de los gastos involucrados con la empresa y los gastos personales por socio. Además ayudará a obtener ahorro en los servicios de telefonía debido a la conciencia que producirá sobre el uso de las extensiones. En relación a esto estamos hablando de 5.05 pesos diarios por socio en llamadas locales. Anualmente la empresa ahorrará 1845 pesos por socio logrando el retorno de inversión para este proyecto en 13.3 días aproximadamente.

Para este proyecto se utilizó el desarrollo rápido de aplicaciones (RAD) como metodología de desarrollo de sistemas. Además, se utilizaron como herramientas para su construcción Visual Basic, Active Server Pages y NetCharts.

El documento en cuestión inicia presentando la información general de la empresa para la que se desarrolló el proyecto. Continúa en el siguiente capítulo explicando el problema a solucionar, la solución propuesta, los objetivos y los beneficios que traerá el proyecto. En el tercer capítulo se expone el funcionamiento del conmutador Ericsson MD-110 y el lenguaje de comandos que este utiliza, también se explica la metodología y las tecnologías utilizadas y por último las bases para la selección del lenguaje de programación del agente recolector de llamadas. En los siguientes capítulos se presentan los resultados obtenidos en las distintas etapas del proyecto, análisis (JRP), diseño (JAD), construcción y liberación. Por último, se incluyen las conclusiones y una descripción de las mejoras a futuro que podrían incluirse en el sistema basado en el desarrollo actual y en la flexibilidad de su diseño.

1. LA EMPRESA

A continuación se describe la información de Supermercados Internacionales H.E.B., la empresa para la cual se desarrolla el presente proyecto. Se muestra su historia, su misión, los productos y servicios que maneja y su estructura organizacional, entre otros datos relevantes.

1.1. Historia.

1.1.1. El principio.

En el año de 1905, la temperatura en las tiendas de abarrotes se mantenía confortable por medio de abanicos de techo en lugar de acondicionadores de aire; la mercancía se transportaba en canastas de mimbre y carretas de madera en lugar de carros de compras. Ese fue el año en el que Florence Butt abrió las puertas de una pequeña tienda de abarrotes familiar a sus amigos y vecinos en Kerville. Esta pequeña población está situada en las ondulantes colinas del centro de Texas. La tienda prosperó, pero se mantuvo pequeña hasta que uno de los hijos de la señora Butt, Howard Edward Butt, se hizo cargo de los negocios después de prestar servicio en la Primera Guerra Mundial. Él tuvo la visión de una compañía de abarrotes que pudiese llegar más allá de las conocidas calles de Kerville.

Durante este tiempo, las tiendas de abarrotes comenzaron a cambiar. Los días del barril abierto de galletas estaban desvaneciéndose. Los negocios cambiaron

del método de pedido a crédito y por teléfono al método de autoservicio, con pago al contado y sin entrega a domicilio.

H.E.Butt se mudó al Valle de Río Grande en 1928 e inició la primera expansión importante de la compañía. La compañía no sólo añadió nuevas tiendas en nuevos lugares, sino que también comenzó a manufacturar productos. La Harlingen Canning Company inició la producción de "productos cultivados y empacados en Texas" en los años treinta. Después se establecieron otras plantas manufactureras. Otras actividades relacionadas con la compañía incluían la agricultura, los huertos y bienes raíces.

En 1938, la compañía mudó sus oficinas centrales de Harlingen a Corpus Christi. Nuevas tiendas se abrieron a lo largo de la costa, Austin, Waco, San Antonio, donde actualmente se albergan en el Complejo de H.E.B. en el Arsenal.

A través de los años, la compañía y sus tiendas cambiaron de diversas maneras. Los estacionamientos reemplazaron los postes para enganchar los caballos. Nuevos productos manufacturados por H.E.B. se colocaron al lado de otros productos en los anaqueles de las tiendas.

Se desarrolló el concepto de hacer todas las compras en un solo viaje. Esto consolidó la pescadería, la carnicería, la farmacia, la florería, y la tienda de departamentos bajo un solo techo. Las tiendas de abarrotes se fueron convirtiendo en supermercados. [HEB, 1999]

1.1.2. H.E.B. en la actualidad.

H.E.B. ha logrado establecerse como uno de los mayores vendedores regionales de alimentos al menudeo en los Estados Unidos. Es propiedad de la familia Butt y dirigida por ellos mismos. Desde 1971, cuando Charles Butt se convirtió en el presidente, el número de socios de H. E. Butt Grocery Co. H.E.B. ha crecido de 4,500 a más de 45,000.

El manejo de alimentos para venta al menudeo de H.E.B. consta de cinco divisiones geográficas, cada una encabezada por un vicepresidente de división y un gerente general. Las divisiones del norte, sur y centro de Texas, costera y San Antonio tienen aproximadamente 180 tiendas de venta al menudeo.

H.E.B. también opera tiendas de renta de videos bajo el nombre de H.E.B. Video Central. Con más de 30 tiendas de videos a través del Estado, H.E.B. Video Central es parte de un excitante y diversificado plan para ampliar los horizontes de la compañía.

El formato más nuevo de venta al menudeo de H.E.B. son las tiendas llamadas "Pantry Foods" aunque son más pequeñas y compactas que los supermercados

H.E.B., las tiendas "Pantry Foods" cuentan con cuatro departamentos esenciales: abarrotes, salud y belleza, carnicería y frutería. Estas tiendas han abierto de manera firme nuevos mercados para H.E.B., con más de 40 tiendas en Houston y sus alrededores, el noreste de Texas y con planes a largo plazo para futura expansión.

La operación de las tiendas cuenta con el apoyo de instalaciones para la fabricación, distribución y transportación de productos. Un centro de distribución de gran capacidad en San Antonio se complementa con una bodega en Corpus Christi y otra en San Marcos. Una flotilla de más de 900 tractores y contenedores que operan fuera de las terminales en San Antonio, Vvrharton, Waco, Corpus Christi y Harlingen entregan diariamente mercancía a las tiendas.

La compañía también opera una planta lechera, una planta de carne, una planta de helados, una planta procesadora de fotografía, una pastelería y una cocina en San Antonio, además de una panadería y pastelería en Corpus Christi.

El personal administrativo de apoyo en las oficinas centrales de la compañía aporta su experiencia en las áreas de contabilidad, publicidad, arquitectura, trámites, construcción, procesamiento de datos, recursos humanos y desarrollo de bienes raíces. [HEB, 1999]

1.1.3. H.E.B. en México.

Al iniciar la década de los noventa, la afluencia por más de 5 décadas de clientes mexicanos a las tiendas H.E.B. en Texas, despierta la inquietud en Charles Butt, Presidente del Consejo de H.E.B., de abrir tiendas H.E.B. en México, por lo que el 1 de diciembre de 1995 se inicia la construcción de la primer tienda en México. Esta se encuentra ubicada en la Ave. Gómez Morín #300 Col. Valle del Campestre en San Pedro Garza García, N.L. El total de la inversión fue alrededor de ocho millones de dólares, y abrió sus puertas al público regiomontano el 6 de febrero de 1997. Esta tienda cuenta con siete mil metros cuadrados de piso de venta y quince mil metros cuadrados de estacionamiento.

Actualmente H.E.B. de México cuenta con 5 tiendas en el estado de Nuevo León y una más en el estado de Coahuila.

A medida que H.E.B. avanza hacia el siglo XXI, mantiene el compromiso de lograr tres importantes metas a través de sus tiendas: la máxima consistencia en servicios de utilidad para el cliente, los precios más bajos todos los días y frescura en todos sus productos. El alcanzar estas metas diariamente hará que H.E.B. sea la mejor tienda posible en cada comunidad. [HEB, 1999]

1.1.4. H.E.B. y la comunidad.

H.E.B. se compromete a continuar mejorando la calidad de vida de los clientes y sus comunidades. Para lograr este compromiso, H.E.B. estimula a sus socios a participar activamente en eventos cívicos. La participación activa es la piedra angular de la buena voluntad demostrada por H.E.B. hacia el cliente y la comunidad. H.E.B. contribuye con fondos de la compañía para aquellas organizaciones educativas y de caridad que son de importancia general para los socios y las diversas comunidades en las que presta servicio. La empresa dona cada año el 5% del total de sus utilidades antes de impuestos a obras de beneficencia. La conciencia social de H.E.B. se remonta al 3 de enero de 1934 cuando inicia la fundación H.E.B..

En 1984 realizó uno de sus más grandes compromisos comunitarios al dar de comer al hambriento dentro de sus comunidades, estableciendo el "Return Goods Center", el cual permite el recolectar los productos que ya no se venden en la tienda, pero que todavía son útiles, de estas forma, en lugar de tirar los productos éstos son utilizados a beneficio de personas necesitadas.

Para esto, H.E.B. estableció un sistema donde los artículos de la tienda son almacenados y entregados por camión a Bancos de Alimentos localizados en toda su área de mercado. Como resultado de este programa, H.E.B. dona alrededor de 5,400 toneladas de comida cada año a 13 Bancos de Alimentos en Texas, sirviendo a 2,000 comunidades en el Estado. Esto es transportado en 500 trailers llenos de comida.

El compromiso para luchar contra el hambre, va más allá de la frontera del estado de Texas. H.E.B., debido a su experiencia en almacenamiento y distribución, apoya en el diseño y la creación del primer Banco de Alimentos en tierras mexicanos, llamado "Banco Internacional de Alimentos (BIA), el cual fue inaugurado en el mes de agosto de 1993 en Cd. Acuña, Coahuila. El BIA es administrado en su totalidad por H.E.B. y con esto se da inicio a las actividades filantrópicas de H.E.B. en México.

H.E.B. proporcionará un trailer por mes de 10 toneladas de comida, proveniente del Banco de Alimentos de Texas. Esta comida será distribuida dentro de la población de Nuevo León por medio del Banco de Alimentos de Cáritas Monterrey, apoyándose en la experiencia que posee H.E.B. en lo referente a la distribución de alimentos. El apoyo de H.E.B. en la comunidad mexicana se ve reflejado en varias áreas como son alimentación, educación y ecología. Destinando el 90% de sus esfuerzos a la alimentación. Existen grandes expectativas por crear nuevos Bancos de Alimentos similares en los Estados donde H.E.B. instale sus tiendas.

Además, el Banco de Alimentos H.E.B. apoya por medio de donativos en efectivo y en especie, los diferentes proyectos comunitarios que desean llevar a

cabo las instituciones de beneficencia de las comunidades donde se encuentren instaladas sus tiendas. El 22 de diciembre de 1996 se realizó por primera vez en Monterrey el "Feast of Sharing". Esta posada, proporcionó una tradicional comida de Navidad a más de 12,000 personas. El evento fue patrocinado totalmente por H.E.B.. Este evento marca la institucionalización de la posada que ofrecerá H.E.B. a la comunidad regiomontana cada año. [HEB, 1999]

1.2. Misión, visión y valores de H.E.B. México.

1.2.1. Misión.

"Mejorar la calidad de vida de nuestros clientes y socios".

1.2.2. Visión.

- Tener de 40 a 50 tiendas con un amplio enfoque regional.
- Alcanzar ventas cercanas a los \$10,000 millones de pesos.
- Obtener rendimientos financieros competitivos con los autoservicios mexicanos de renombre
- Continuar siendo una compañía "única" donde ¡cada socio cuenta!.

1.2.3. Valores.

1.2.3.1. Valores de negocio.

- Frescura
- Calidad
- Servicio
- Precio Bajo
- Variedad

1.2.3.2. Valores Individuales.

- Liderazgo.
H.E.B. confía plenamente en la capacidad de sus socios para dirigir e influir positivamente en las actividades clave de nuestro negocio y en sus fortalezas para comprometerse, responsabilizarse y tomar las decisiones acertadas en el momento que así se requiera.
- Trabajo en Equipo.
Trabajar en equipos multidisciplinarios es la base del éxito al desarrollar y capitalizar nuestra capacidad para lograr la unión de fuerzas, habilidades, conocimientos, destrezas y competencias de los socios.

- **Innovación.**
Se busca constantemente la mejora continua para alcanzar los estándares que identifican a una organización de alto desempeño, por lo que se alienta la creatividad y participación a través de la comunicación como el motor para las ideas e iniciativas en beneficio de la empresa, su gente, clientes, servicios y productos.
- **Honestidad.**
Se debe ser integro, pensando y actuando siempre con base en la educación que se recibe desde pequeño, y así distinguir el comportamiento, reforzar el ambiente de trabajo, para ganar la confianza de los clientes y alcanzar el respeto de los compañeros.
- **Orientación al servicio.**
La principal ventaja competitiva de H.E.B. esta sustentada en su cultura de servicio que la identifica además de que promueve el ambiente idóneo para la productividad, comunicación abierta, lealtad y satisfacción personal.
- **Enfoque a resultados.**
H.E.B. cree que para asegurar la rentabilidad y el desarrollo sostenible del negocio es necesario alinear sus procesos, estructuras y actividades a la misión y estrategias de la empresa, viviendo una cultura que los distinga por ofrecer siempre una valor agregado a sus clientes.
- **Entusiasmo.**
H.E.B. fomenta el “entusiasmo organizacional” a través del balance adecuado entre familia, trabajo y empresa, siendo conscientes en todo momento de su compromiso para garantizar a sus clientes la calidad y calidez del servicio que se merecen.

1.3. Productos y servicios.

HEB México es una cadena de Supermercados enfocada a satisfacer las necesidades de productos básicos de sus clientes, basando su estrategia de ventas en mantener siempre precios bajos, frescura y calidad para el bienestar de sus consumidores. Las tiendas en México cuentan con los siguientes departamentos:

Carnes	Novedades
Pescados	Música
Panadería	Farmacia
Mercancías Generales	Salchichonería
Higiene personal	Café
Perfumería	Comida Caliente
Abarrotes No Comestibles	Restaurante
Laboratorio de Fotografía	Abarrotes
Bebes	Vinos
Mercancía Mundo Infantil	Lácteos
Congelados	Mundo Infantil
Servicio	Frutas y verduras
Florería	Empaques

Tabla 1.1. Departamentos de las tiendas HEB México.

1.4. Estructura organizacional.



Figura 1.1. Organigrama de la empresa.

La estructura organizacional de la empresa está encabezada por las siguientes seis direcciones: administración y finanzas, comercial, operaciones, construcción y bienes raíces, recursos humanos y logística. Todas estas comandadas por una dirección general presidida por el Sr. Howard E. Butt III, dueño de gran parte de la compañía. Lo anterior se puede observar en la figura 1.1. Cabe mencionar que la dirección de administración y finanzas es quien hará uso y dará el soporte a este proyecto.

Las funciones principales de la dirección de administración y finanzas son las siguientes: desarrollo de la planeación estratégica de la empresa, llevar a cabo el plan de negocios de cada año, controlar y asignar los presupuestos del resto de las áreas de la empresa, desarrollar la estrategia tecnológica que debe seguir la compañía, además de realizar pagos a los proveedores, procesos contables y asuntos fiscales y legales de H.E.B.

El área encargada de los departamentos de cada una de las tiendas es la dirección comercial, esta realiza el análisis de mercado y de la competencia,

realiza las negociaciones con los proveedores, cuida la imagen de los precios, difunde y controla los valores del negocio.

La dirección de operaciones lleva a cabo el control interno de la operación de las tiendas, mantiene bajos costos de control interno, fomenta e incrementa la calidad en el servicio a los clientes, proveedores, y la calidad de vida de sus empleados.

Las principales funciones de la dirección de construcción y bienes raíces son localizar y negociar terrenos de compra o venta para la futura construcción de una tienda, controlar las obras que se lleven a cabo para la apertura de nuevas tiendas y edificios corporativos dentro y fuera de la ciudad.

La dirección de recursos humanos realiza el reclutamiento y la selección del personal, además de encargarse del desarrollo humano de los socios y mantener el buen ambiente organizacional, administra los préstamos y compensaciones de los socios.

El área encargada de coordinar la cadena de resurtido de mercancía desde los proveedores hasta la tienda, es la dirección de logística, esta controla y coordina los siguientes puntos: la distribución de la mercancía entre tiendas y bodegas, inventarios a nivel compañía, y la transportación y surtido de los productos a las tiendas.

1.4.1. Organigrama de la dirección de administración y finanzas.

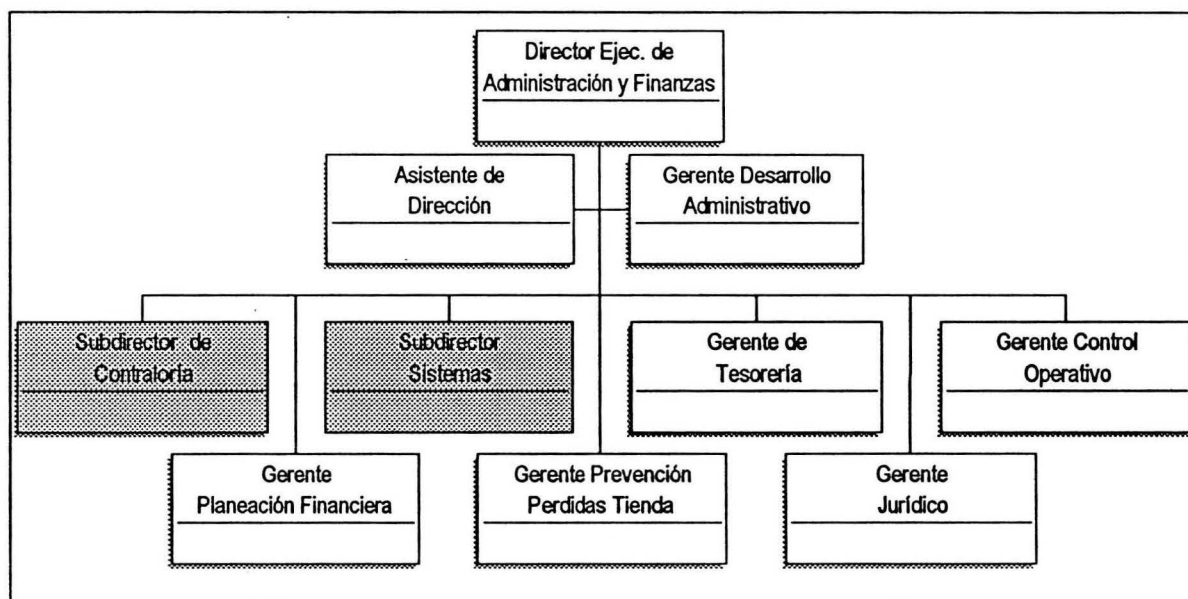


Figura 1.2. Organigrama de la dirección de administración y finanzas.

La dirección de administración y finanzas esta integrada por dos subdirecciones, de contraloría y de sistemas, y seis gerencias, desarrollo administrativo, planeación financiera, prevención de perdidas en tienda, tesorería, jurídica, y la gerencia operativa.

La subdirección de contraloría se encarga de todos los procesos contables de HEB México, como la nómina, impuestos, cuentas por pagar, cuentas por cobrar, costos, inversiones, etc.

La subdirección de sistemas se encarga de todo el desarrollo tecnológico de la empresa, como son los sistemas de información, la red de comunicaciones, el punto de venta, así como generar los procedimientos operativos y planes de contingencia para la operación de las tiendas y las oficinas corporativas.

1.4.2. Organigrama de la subdirección de contraloría.

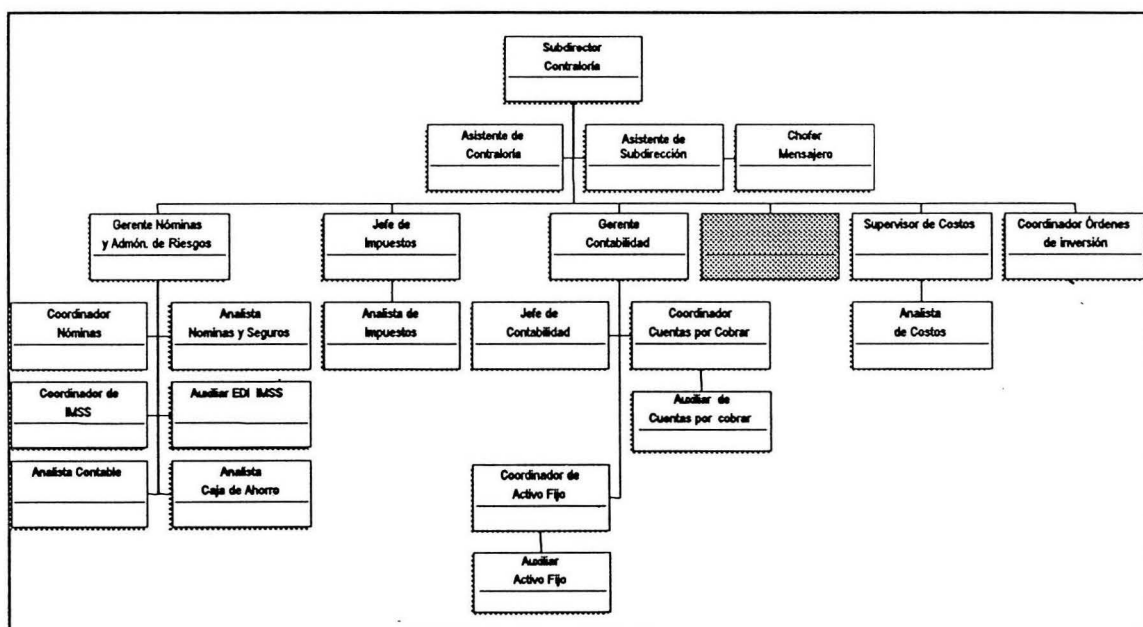


Figura 1.3. Organigrama de la subdirección de contraloría.

La subdirección de contraloría está integrada por la gerencia de nóminas y administración de riesgos, la gerencia de contabilidad, la jefatura de impuestos, la supervisión de costos, la supervisión de cuentas por pagar y la coordinación de órdenes de inversión.

La supervisión de cuentas por pagar se encarga de administrar y controlar los pagos a todos los proveedores, tanto de productos como de servicios.

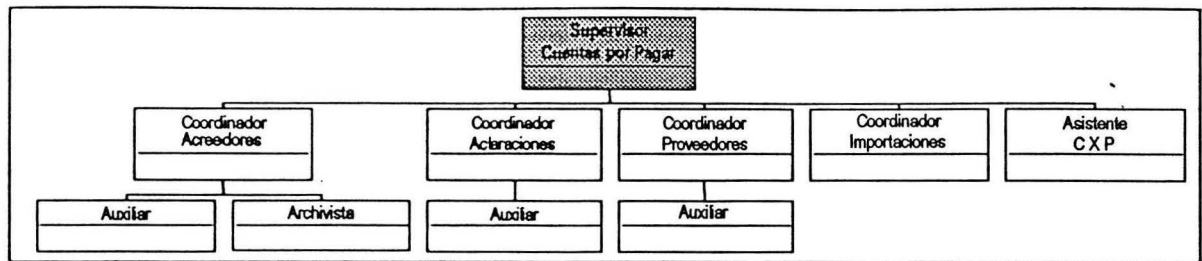


Figura 1.4. Organigrama del departamento de cuentas por pagar.

La supervisión de cuentas por pagar se integra por varias coordinaciones que se encargan de los acreedores, aclaraciones, proveedores, y de las importaciones. El supervisor de cuentas por pagar será uno de los usuarios del sistema.

1.4.3. Organigrama de la subdirección de sistemas.

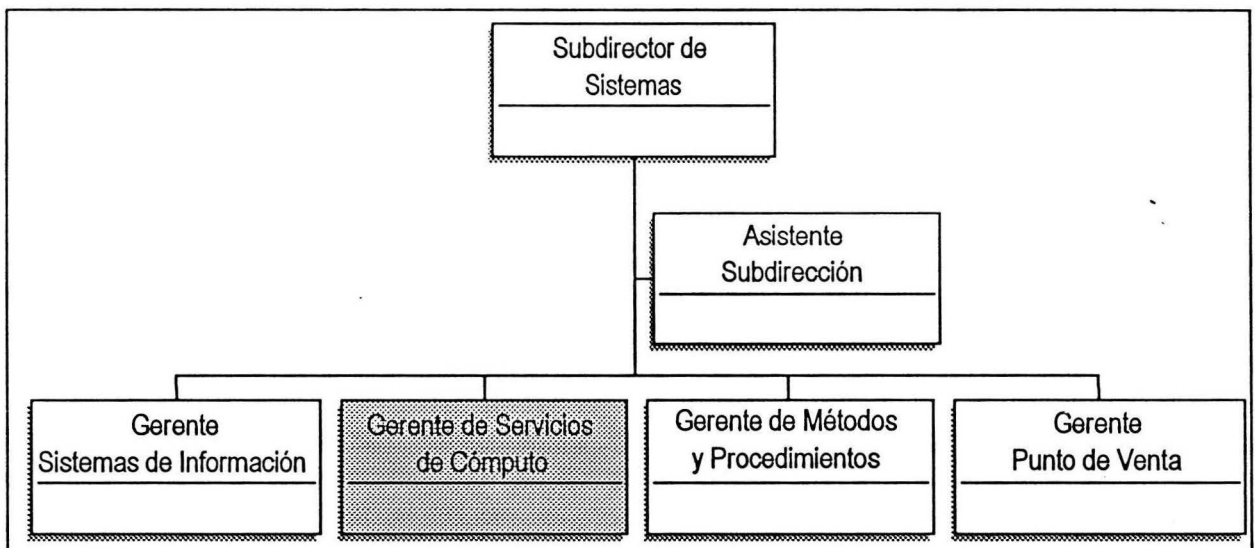


Figura 1.5. Organigrama de la subdirección de sistemas.

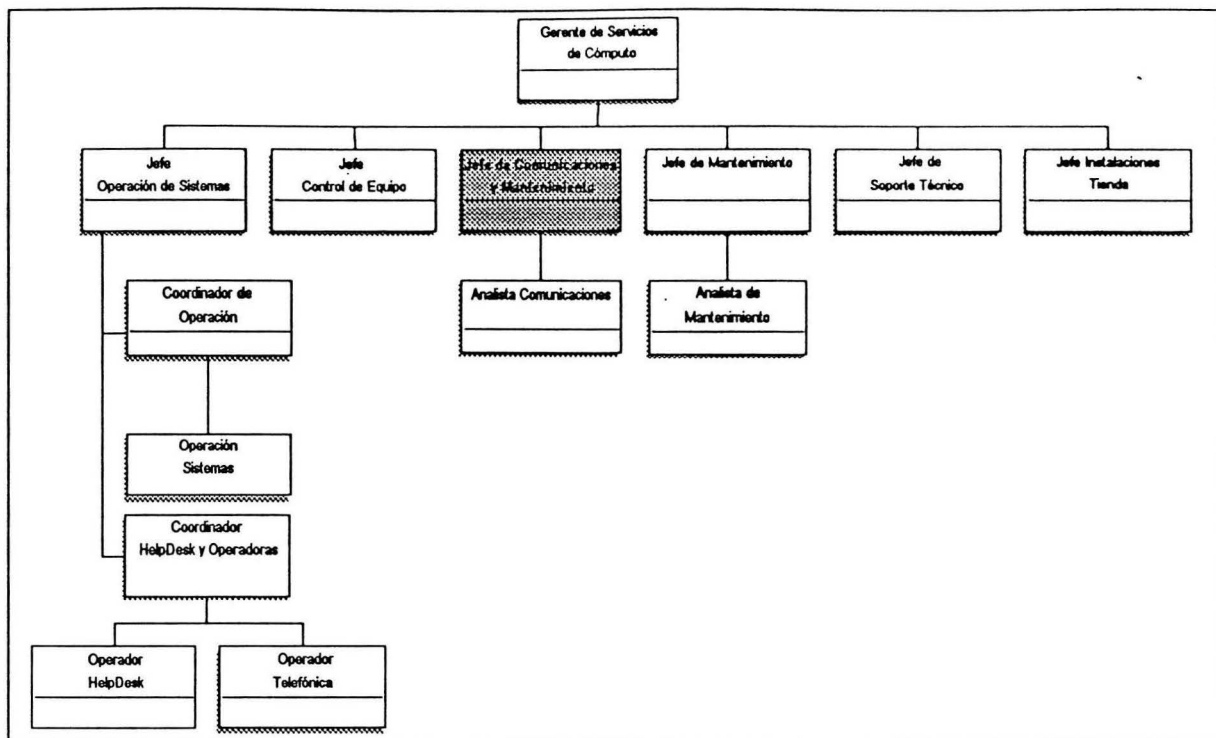


Figura 1.6. Organigrama de la gerencia de servicios de cómputo.

La figura 1.5. representa el organigrama de la subdirección de sistemas de H.E.B. México, esta es una subdirección de la dirección de administración y finanzas y cuenta con cuatro gerencias, la de sistemas de información, la de servicios de cómputo, la de métodos y procedimientos y la de punto de venta.

La gerencia de servicios de cómputo está encargada de dar soporte técnico, realizar instalaciones, dar soporte a la operación de sistemas y mantener la red de comunicaciones. Esta gerencia está integrada por varias jefaturas, operación de sistemas, control de equipo, comunicaciones y mantenimiento, mantenimiento, soporte técnico e instalaciones de tienda.

La jefatura de comunicaciones y mantenimiento tiene la labor de mantener la red de comunicaciones de HEB México. La persona encargada de esta jefatura será uno de los usuarios del sistema, además del resto de los socios que necesiten tener acceso a los reportes generados por el sistema.

1.5. Infraestructura computacional.

La infraestructura computacional de H.E.B. México se divide en las dos áreas siguientes:

- Infraestructura de las tiendas
- Infraestructura de las oficinas corporativas

Cada una de las sucursales de H.E.B. México cuenta con un site, al igual que sus oficinas corporativas. En el site de las oficinas corporativas está concentrado todo el equipo de cómputo y comunicaciones que soporta la operación del negocio. Además, se cuenta con el equipo de puntos de venta y los equipos computacionales de los socios que necesitan este tipo de herramientas para el desarrollo de sus actividades.

La siguiente tabla es un listado del equipo instalado en las sucursales y en las oficinas corporativas de H.E.B. México.

Tiendas o Sucursales (Datos por sucursal)	
Hardware	Software
4 HP NetServer LH3	Sistema STEP 2.1
40 PC's entre Pentium y Pentium II	Windows NT 4.0
	Windows 95
	MS Office 97 y 2000

Oficinas Corporativas o Centro de Apoyo	
Hardware	Software
2 HP-9000 E35	Sistema STEP 2.1
4 HP NetServer LH3	Windows NT 4.0
Piso 1 – 40 PC's entre Pentium y Pentium II	Windows 95
Piso 2 – 60 PC's entre Pentium y Pentium II	MS Internet Explorer 5.0
Piso 3 – 84 PC's entre Pentium y Pentium II	Visual Basic 6.0
Piso 4 – 45 PC's entre Pentium y Pentium II	SQL Server 6.5 y 7.0
Piso 5 – 22 PC's entre Pentium y Pentium II	Visual Interdev 6.0
	Net Charts 2.2
	MS Office 97 y 2000
	Power Play 6.2
	Impromptu 5.0
	Reflection 6.0

Comunicaciones por Sucursal
1 Enlace E1 Voz / Datos de cada sucursal al centro de apoyo
Tiendas 3-5 Líneas Analógicas
Centro de Apoyo 40 Líneas Digitales
1 conmutador Ericsson MD-110 por sucursal y centro de apoyo

Tabla 1.2. Infraestructura computacional de HEB México.

2. JUSTIFICACIÓN DEL PROYECTO

En el presente capítulo se explican los procesos actuales involucrados con el proyecto y sus problemas. Además se exponen las áreas de oportunidad encontradas, la solución propuesta y los objetivos del proyecto.

2.1. Situación actual.

2.1.1. Antecedentes.

Actualmente H.E.B. de México cuenta con una red de datos y de voz que une a todas sus sucursales y oficinas, la cual juega un papel muy importante en la operación de la empresa. A través de la red de datos se transmite toda la información de ventas, productos, precios, proveedores, etc., además de servir también como un medio de comunicación entre las tiendas y las oficinas. La red de voz es utilizada para la comunicación telefónica entre las sucursales y las oficinas así como la comunicación con entidades externas como proveedores y clientes.

En cada una de las tiendas se cuenta con un conmutador Ericsson MD-110, los cuales se enlazan con el Centro de Apoyo donde se encuentran las troncales primarias de salida a Telmex, y cada tienda cuenta con 3 a 5 troncales analógicas para tráfico de voz en una ruta de respaldo, como se muestra en la figura 2.1.

Los conmutadores realizan los enlaces entre las extensiones y las entidades externas. Actualmente estos conmutadores son configurados realizándose una conexión vía telnet y ejecutando una serie de comandos propios del lenguaje del conmutador que le indican la información que debe de contener y el modo de trabajo. Este tipo de conmutadores tienen además la facilidad de que al finalizar una llamada se envíen por un puerto serial los detalles de la llamada, la extensión que la realizó, el teléfono destino, la duración de la llamada, el tipo de llamada, la fecha y hora de la llamada, etc. Sin embargo, esta facilidad del conmutador aún no es utilizada para ninguna finalidad.

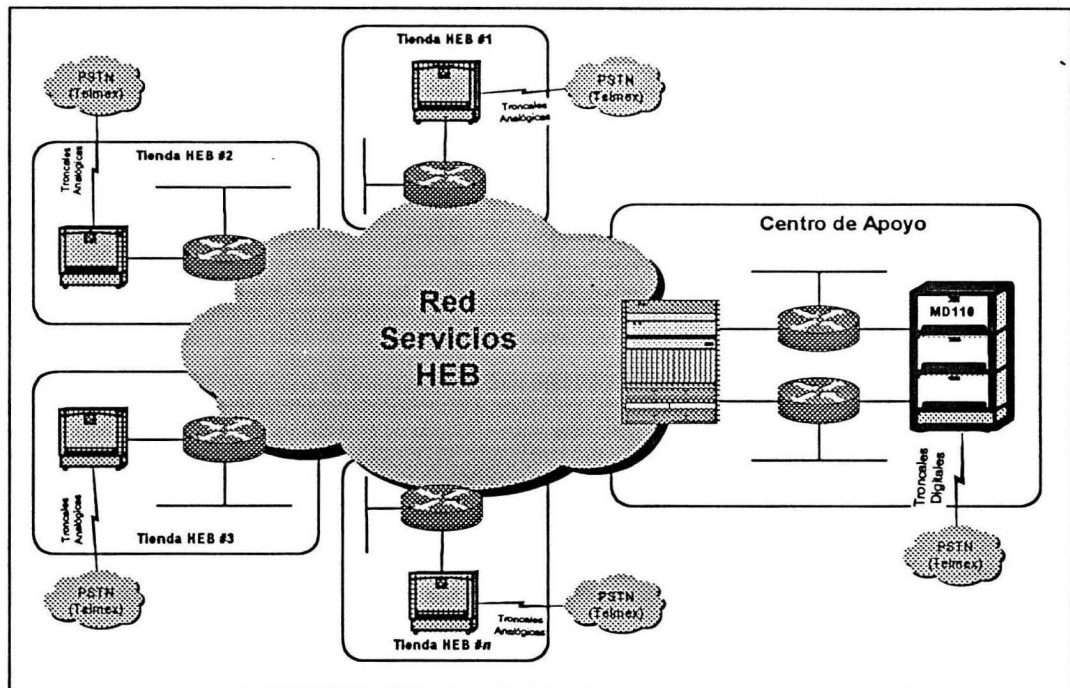


Figura 2.1. Red de voz de H.E.B. México.

Actualmente la red de voz no cuenta con ningún mecanismo mediante el cual se pueda tener conocimiento sobre la información de las llamadas realizadas, por lo menos en cuanto a los servicios otorgados por Telmex, ya que son los que generan un costo para la empresa, de modo que no se sabe ciertamente que llamadas se involucran directamente con el negocio y cuáles no. Telmex realiza una facturación mensual para toda la empresa y no por sucursal, debido a esto y a que no se tiene conocimiento de cuanto gasta cada tienda o cada departamento en llamadas, se tiene que realizar una distribución proporcional de los costos de las llamadas entre las tiendas y los departamentos de la empresa.

2.1.2. Descripción del proceso actual.

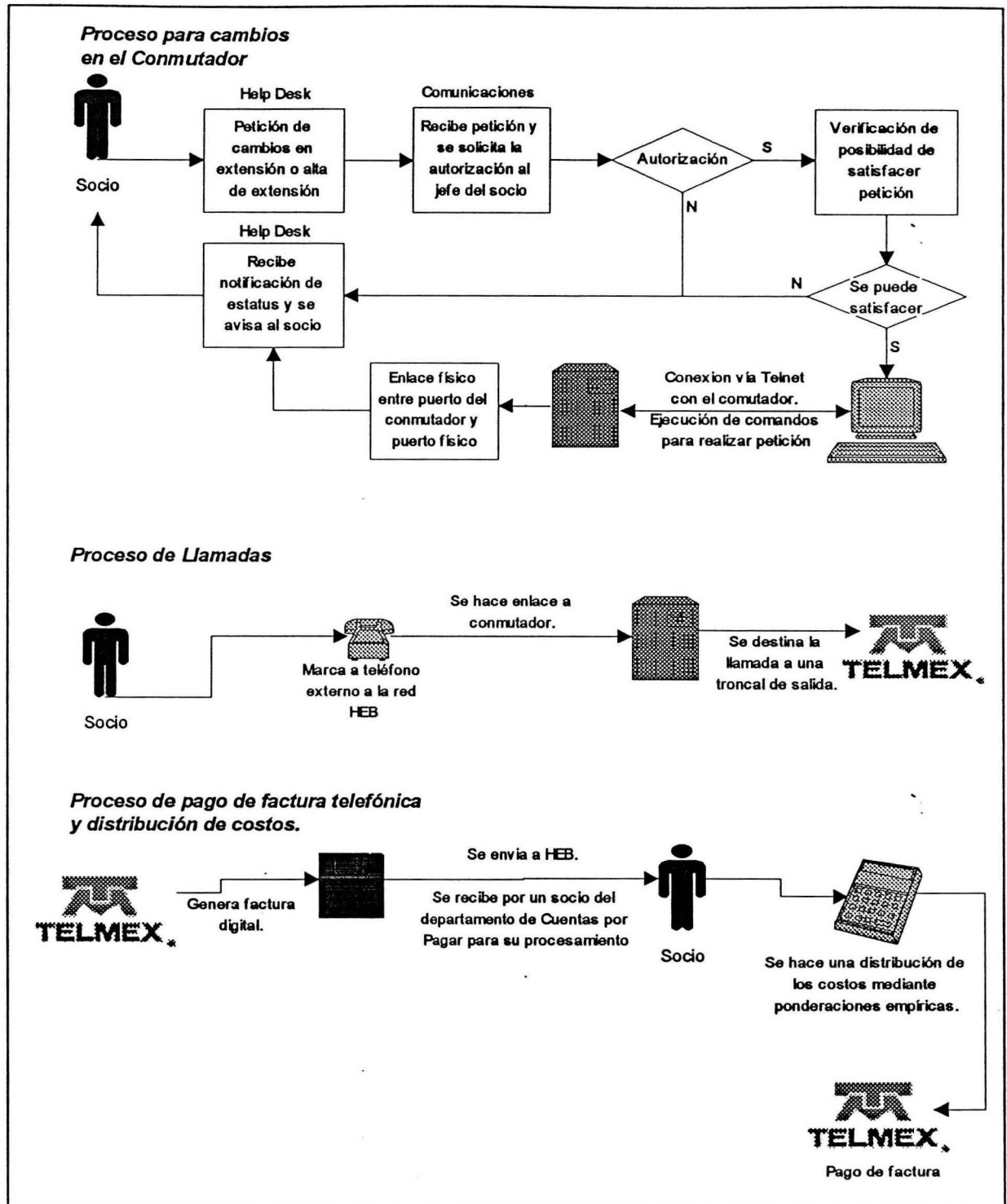


Figura 2.2. Procesos actuales involucrados con el proyecto.

2.1.2.1. Proceso para cambios en el conmutador.

Este proceso se genera cuando un socio requiere un cambio en los datos de su extensión o es necesario asignarle una extensión al socio, por ejemplo, cambiar de puerto la extensión actual, o cambiar el título de despliegue en los teléfonos digitales, entre otros. El socio realiza la petición al Help Desk de HEB. El Help Desk se encarga de enviar la solicitud de servicio al departamento correspondiente en caso de que las personas del Help Desk no puedan dar solución al problema. En este caso la solicitud es pasada directamente al departamento de comunicaciones. La persona del departamento de comunicaciones encargada de realizar los cambios en el conmutador antes de que la solicitud esté aceptada, pide una autorización al jefe inmediato del socio que realizó dicha solicitud. Si se autoriza, se hace una verificación para comprobar que exista disponibilidad en el conmutador para cumplir con la solicitud. Esto implicará la verificación de los puertos y extensiones vacantes ya sean digitales o analógicos, que tipos de aparatos telefónicos hay en existencia, etc. Si hay disponibilidad para cumplir la solicitud, se realiza una conexión vía telnet al conmutador y se ejecutan una serie de comandos propios del mismo, mediante los cuales se realizará la modificación. En todos los casos se le notifica al usuario el estatus de su petición mediante el Help Desk.

2.1.2.2. Proceso de llamadas.

Inicialmente, el socio marca desde una extensión de la red de voz de HEB a un teléfono externo a esta, ya sea local, de larga distancia nacional o internacional o un teléfono celular. El teléfono hace enlace con el conmutador y este identifica que el número marcado es un número externo y enruta la llamada hacia una de las salidas a Telmex, este tipo de conmutadores envía a través de un puerto serial un registro con los datos de la llamada terminada, pero actualmente no existe un mecanismo que recopile esos datos. Cabe destacar que no existe algún tipo de registro de las llamadas realizadas.

2.1.2.3. Proceso de pago de factura telefónica y distribución de costos.

Este proceso inicia con la generación de una factura digital por parte de Telmex, esta factura es enviada al departamento de cuentas por pagar de HEB. Esta es recibida por un socio, quien se encarga de revisar la facturación y hacer una distribución de los costos entre las unidades de negocio y departamentos de la empresa. La distribución de los costos entre las unidades de negocio y los departamentos se hace a través de un prorrateo. Posteriormente se realiza el pago de la factura a Telmex.

2.1.3. Áreas de Oportunidad.

Un área de oportunidad detectada es la facilidad que presentan los conmutadores Ericsson MD-110 de enviar la información de las llamadas a través de un puerto serial al terminar la llamada. Mediante esta facilidad es posible recolectar todas las llamadas que el conmutador realiza fuera de la red de voz de HEB. Contando con esta información y haciendo uso de la Intranet de la empresa, se podrá explotar la información desde cualquier explorador de Internet que se encuentre dentro de la red de datos de HEB México, sin necesidad de realizar una instalación previa del software de tarificación. Pudiendo además, realizar una conciliación con la facturación digital de Telmex y en caso de se detecte algún error en la facturación poder hacer una reclamación a Telmex.

La particularidad de los conmutadores Ericsson MD-110 de poder ejecutar comandos de configuración desde una conexión vía telnet, permitirá elaborar una interfase *visual* para la configuración de ciertas funciones del conmutador.

Aprovechando la estructuración de la red de datos de HEB México será posible tener centralizada la información de las llamadas de todas las sucursales.

Con esta información los usuarios del sistema tendrán la facilidad de:

- Tener la información de las extensiones centralizada.
- Distribuir entre las unidades de negocio los costos reales de las llamadas realizadas.
- Contar con una mejor visión de los gastos que no se involucran directamente con la empresa y de los que si se involucran.
- Realizar con mayor veracidad los presupuestos de gastos de las unidades de negocio.

2.2. Solución propuesta.

Para aprovechar las áreas de oportunidad antes mencionadas se propone desarrollar un sistema mediante el cual sea posible obtener los detalles de las llamadas realizadas por medio del conmutador y almacenarlos en una base de datos centralizada, de modo que la información pueda ser explotada haciendo uso de la Intranet de la empresa.

El conmutador del centro de apoyo estará conectado a una PC o servidor en el cual se estará ejecutando un agente que recibirá los detalles de las llamadas cuando el conmutador registre el término de cada una de ellas y las envíe a través de su puerto serial, estas se almacenarán en una base de datos en un servidor de SQL. Sólo en caso de que no exista conexión con la base de datos, se utilizará el almacenamiento local para que la información generada en el

lapso de restablecimiento de la conexión no se extravíe, y pueda ser enviada al momento de establecer nuevamente conexión con la base de datos.

Se han propuesto dos herramientas de desarrollo para este agente, Visual Basic y Java, las cuales serán evaluadas, para conocer cuál de ellas puede ser más eficiente en este tipo de procesos.

La información almacenada en la base de datos de SQL se explotará a través de reportes gráficos y de texto en la Intranet de la empresa. Estos reportes deben contemplar niveles organizacionales (unidad de negocio, sucursal, departamento, socio o extensión), tiempos (día, semana fiscal, mes, bimestre), y tipos de llamadas (servicio medido, larga distancia nacional, larga distancia internacional, celulares).

El sistema contará además con un proceso mediante el cual se pueda realizar un comparativo entre la información almacenada del conmutador y la información de facturación de Telmex. Esta información es entregada por Telmex en un archivo con los detalles de las llamadas facturadas, los cuales deben de ser conciliados con la información de los conmutadores y obtener así los costos de las llamadas y errores de facturación, los cuales serán evaluados por el departamento de cuentas por pagar para la canalización de los costos.

Además se incluirá una interfase mediante la cual sea posible administrar la información manejada por el conmutador como extensiones, puertos y usuarios, sin tener que utilizar los comandos del lenguaje del conmutador.

Se incluirá también un módulo mediante el cual las solicitudes de altas y cambios de extensiones sean capturadas. El departamento de comunicaciones podrá consultar estas solicitudes y el sistema presentará la información del conmutador que sea necesaria para satisfacer la solicitud y el usuario podrá seleccionar la información con la que desea que se realice el alta o el cambio en el conmutador de forma automática.

Con lo anteriormente mencionado los procesos actuales involucrados en el proyecto se realizarían como se muestra en la figura 2.3.

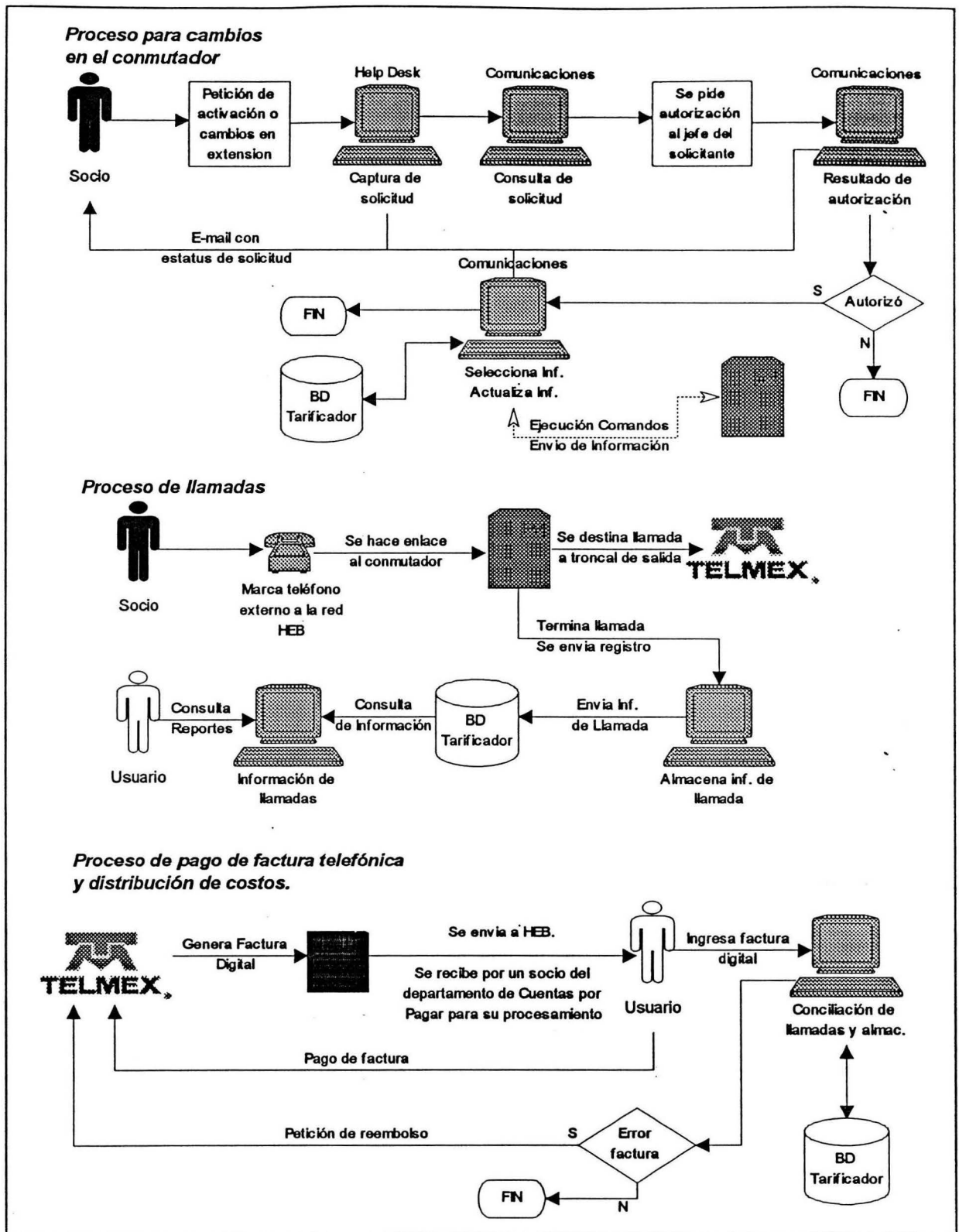


Figura 2.3. Procesos de la solución propuesta.

2.2.1. Objetivo general del proyecto.

Desarrollar un sistema computacional que almacene la información de las llamadas realizadas en la red de voz de H.E.B. registradas por los conmutadores en una base de datos de SQL, permitiendo explotar la información de modo que sea posible conocer a detalle los datos de las llamadas realizadas por cada una de las extensiones de la red. Permitir a través de una interfase amigable realizar la administración del conmutador, pudiendo realizar cambios en los datos generales de las extensiones. Realizar una conciliación de la información recopilada con los datos de facturación de Telmex para verificar su veracidad y los costos de las llamadas realizadas por las extensiones.

2.2.2. Objetivos particulares.

- Desarrollar el sistema utilizando la metodología Desarrollo Rápido de Aplicaciones (RAD). [McCONNELL, 1996].
- Diseñar y construir una base de datos para administrar la seguridad del sistema. [HARRINGTON, 1999]
- Diseñar y construir una base de datos con reglas de integridad bien definidas utilizando el modelo relacional que almacene los detalles de las llamadas registradas por los conmutadores de la red de voz de HEB. [HARRINGTON, 1999]
- Permitir al usuario restringir los accesos a las opciones del sistema, mediante la configuración de usuarios y perfiles. [SERVATI, 1998]
- Desarrollar un proceso para recolectar la información de las llamadas que generan los conmutadores de la red de voz y almacenar esta información en una base de datos centralizada, y si existe un error en su ejecución avisar al administrador por medio de un correo electrónico. [GUREWICH, 1998] [BIGUS, 1997]
- Determinar el lenguaje a utilizar para la programación del agente de recopilación de datos (Visual Basic o Java). [GUREWICH, 1998] [BIGUS, 1997]
- Desarrollar un proceso que permita almacenar la información recopilada por el agente de recolección de los detalles de las llamadas, en la base de datos de SQL o en caso de contingencia en un almacenamiento local para evitar el extravío de información. [GUREWICH, 1998] [BIGUS, 1997]
- Desarrollar un proceso para la depuración de los archivos de almacenamiento local. [GUREWICH, 1998] [BIGUS, 1997]
- Desarrollar una interfase y los procesos necesarios que permitan actualizar y consultar la información de extensiones, puertos y usuarios, almacenada en el conmutador. [GUREWICH, 1998] [BIGUS, 1997]
- Desarrollar un módulo que permita capturar las solicitudes de cambios en la información de extensiones, puertos y usuarios, y que mediante la información de la solicitud se presente la información del conmutador que pueda satisfacer la solicitud y permitir al usuario realizar los cambios.

- Desarrollar un proceso que permita avisar al interesado el estado de su solicitud por medio de un correo electrónico.[GUREWICH, 1998] [BIGUS, 1997]
- Desarrollar la interfase y los procesos necesarios para poder realizar una conciliación de los datos de facturación de Telmex contenidos en un archivo y los recopilados por el sistema.
- Desarrollar los procesos de seguridad para acceso a los reportes y a los actualizadores del sistema. [BROPHY, 1997] [GUREWICH, 1998]
- Generar los siguientes reportes con la información obtenida por el sistema, en modo texto y/o gráfico:
 - Directorio telefónico de extensiones. (texto)
 - Reporte de llamadas por socio. (texto y gráfico)
 - Reporte teléfonos mas marcados por socio. (texto y gráfico)
 - Reporte de teléfonos no involucrados con la empresa por socio. (texto y gráfico)
 - Reporte de llamadas por Departamento. (texto y gráfico)
 - Reporte de llamadas por Unidad de Negocio. (texto y gráfico)
 - Reporte de llamadas por Tienda o Sucursal. (texto y gráfico)
 - Reporte de costos para el departamento de cuentas por pagar. (texto)

Los reportes antes mencionados podrán ser obtenidos por las siguientes unidades de tiempo:

- Día.
- Semana Fiscal.
- Mes.
- Bimestre.

[BROPHY, 1997]

- Manejo de puntos de alerta en los reportes para identificar el mal uso de las extensiones asociando estas con los servicios de telefonía que se les permiten realizar. [BROPHY, 1997]
- Elaborar la documentación necesaria para el desarrollo y mantenimiento del sistema. [McCONNELL, 1996]

2.2.3. Alcances.

- Se generarán los reportes antes mencionados en los objetivos específicos, utilizando niveles (unidad de negocio, sucursal, departamento, socio o extensión), tiempos (día, semana, quincena, mes, semestre, bimestre), tipos (servicio medido, larga distancia nacional, larga distancia internacional, celulares).
- En cuanto a la administración del conmutador Ericsson MD-110, será posible habilitar o deshabilitar extensiones, cambiar los puertos de las extensiones, cambiar los datos de las extensiones (nombre del usuario y otros datos generales) y consultar la información contenida en el conmutador.

2.2.4. Análisis costo-beneficio.

2.2.4.1. Costos

Costos de Desarrollo	Inversión Inicial
1 Computadora Pentium II 350 Mhz.	\$15,000.00
Sueldo Analista de 4 meses.	\$22,000.00
Consumibles	\$500.00
Total de Costos de Desarrollo	\$37,500

Tabla 2.1. Costos de Desarrollo.

2.2.4.2. Beneficios

En las tablas 2.2. y 2.3. se analizan los costos de las llamadas de servicio medido y a celulares respectivamente. Para el análisis de las llamadas después de que sea implementado el sistema se utilizó la estimación del departamento de cuentas por pagar de una reducción de un 40% en los costos, esto al tener un registro y un control de las llamadas que se realizan por cada departamento.

Gastos en Llamadas Locales	Antes	Después
Promedio de llamadas locales diarias por socio	8.75 llamadas	5.25 llamadas
Costo por llamada local	\$1.444	\$1.444
Gasto diario en llamadas locales por socio	\$12.635	\$7.581
Gasto anual en llamadas locales por socio	\$4611.78	\$2767.07

Tabla 2.2. Costos de llamadas locales.

Gastos en Llamadas Celulares	Antes	Después
Promedio de minutos diarios por socio	0.91 llamadas	0.55 llamadas
Costo por minuto celular	\$4.002	\$4.002
Gasto diario en llamadas a celular por socio	\$3.641	\$2.201
Gasto anual en llamadas a celular por socio	\$1328.97	\$803.37

Tabla 2.3. Costos de llamadas a celulares.

Obteniendo con lo anterior un ahorro diario de \$6.494 pesos por socio.

Entre otros beneficios estarían:

- Tener la información de las extensiones centralizada.
- Distribuir mejor los costos.
- Presupuestos más exactos y confiables.
- Tener una visión real de los gastos involucrados con la empresa y los gastos personales por socio.
- El tiempo de respuesta a las solicitudes de activación y cambios en las extensiones del socio mejorará.
- Tener una pronta disponibilidad a la activación de una extensión.

2.3. Requerimientos de hardware y software.

2.3.1. Software.

	<i>Cliente</i>	<i>Servidor Web</i>	<i>Servidor de SQL</i>	<i>Recolector de Llamadas.</i>	<i>Ericsson MD-110</i>
Desarrollo	Windows 95 ó 98. Explorer 4.0 ó 5.0. Front Page 98 o InterDev. Java Visual Basic 5 ó 6. SQL Server 7.0 MS Office 97.	Windows NT 4.0. IIS 4.0. NetCharts 2.2.	Windows NT 4.0. SQL Server 7.0	Windows NT 4.0	Lenguaje Propio.
Operación	Windows 95 ó 98. Explorer 4.0 ó 5.0.	Windows NT 4.0. VBScript. IIS 4.0. NetCharts 2.2.	Windows NT 4.0. SQL Server 7.0	Windows NT 4.0.	Lenguaje Propio.

Tabla 2.4. Requerimientos de Software.

2.3.2. Hardware.

Desarrollo

Procesador Pentium 250MHz. Como mínimo.

RAM 32 MB como mínimo.

3 GB libre de espacio en disco duro.

Monitor SVGA o UVGA.

Infraestructura de Red Local.

Conmutador MD-110.

Servidor de Web.

Servidor de SQL Server 7.0.

Operación

Cliente: Procesador Pentium 250MHz. Como mínimo.
 RAM 32 MB como mínimo.
 1 GB libre de espacio en disco duro.
 Monitor SVGA o UVGA.

Recolector: Procesador Pentium 250MHz. Como mínimo.
 RAM 40 MB como mínimo.
 2 GB libre de espacio en disco duro.
 Monitor SVGA o UVGA.

Infraestructura de Red Local.
Conmutador MD-110.
Servidor de Web.
Servidor de SQL Server 7.0.

3. MARCO TEÓRICO

A continuación se expone la información teórica que respalda el desarrollo del presente proyecto. Se describe información sobre el conmutador, la metodología de desarrollo y las herramientas utilizadas.

3.1. PBX Ericsson Consono MD-110.

3.1.1. Descripción general.

Ericsson es el cuarto surtidor más grande de equipo de telecomunicaciones en el mundo. Cuenta con 70,000 obreros, fabrica el equipo en 43 países y con dirección del negocio en 100 países en los 5 continentes. Operando un 40% de las comunicaciones celulares mundiales, Ericsson es #1 para la telefonía inalámbrica.

A través de sus PBX's (Private Branch Exchange), lo que podríamos traducir literalmente como Intercambio de Ramificaciones Privado o switch de teléfonos da servicio a un negocio u organización. El PBX provee de servicios telefónicos incluyendo llamadas internas, y acceso a la red pública de telefonía. Este permite que un pequeño número de líneas de salida sean compartidas por todas las personas de la organización. Los PBX avanzados algunas veces proveen servicios de auto-asistencia, correo de voz, y distribución automática de llamadas.

El PBX provee la conexión física entre la línea externa utilizada por la persona quien llama, y el teléfono del empleado. Es privado porque pertenece a la organización y está localizado dentro de sus propiedades. Es un intercambio de ramificaciones porque realiza conexiones dentro del negocio del mismo modo que un teléfono normal. Este conecta todos los números telefónicos de la compañía telefónica.

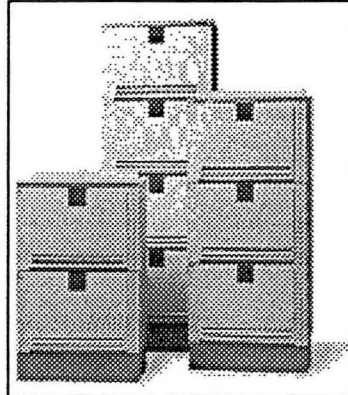


Figura 3.1. Módulos del PBX Ericsson Consono MD110.

El PBX Ericsson Consono MD110 (figura 3.1.) es una solución de comunicaciones totalmente integrada que da a una compañía flexibilidad al máximo. El PBX MD110 incorpora una arquitectura descentralizada otorgando una completa transparencia sin importar la distancia entre los módulos y el número de usuarios. Además, sin una configuración predefinida, el sistema se construye alrededor de las necesidades físicas de la organización. El PBX MD110 permite que los módulos sean empilados conforme se requiera.

El Ericsson MD110 es un PBX para voz y datos digitales de gran capacidad y utiliza estándares internacionales de software para el procesamiento y distribución de la información. Su infraestructura transmite transparentemente voz digital, datos y vídeo a través del mismo sistema de red. La red puede consistir en nodos alámbricos o inalámbricos, o bien como conexiones WAN entre la gente usando terminales alámbricas o móviles.

Las funciones del sistema están uniformemente disponibles para todos los usuarios, es decir, sin diferencias en las características o las funciones aun para los usuarios situados a 1,200 millas del PBX. Con su arquitectura de red, el PBX MD110 provee la más avanzada solución de sistema sencillo en EEUU. Ericsson dotó su PBX con capacidades altamente configurables, una característica da un buen desempeño en su nuevo papel como plataforma integrada para una red de multimedia.

La configuración flexible de los PBX MD110, más su capacidad para manejar hasta 26,000 puertos de voz y datos en un solo switch, permite a los propietarios

del sistema construir una red privada de procesamiento y distribución de información capaces de ampliarse y adaptarse a los requerimientos cambiantes. Incluso los sistemas originales MD110 se pueden actualizar al último producto sin la necesidad de hacer cambios en el hardware.

3.1.2. Lenguaje.

3.1.2.1. General.

La comunicación máquina-hombre se realiza mediante la ejecución de comandos y confirmación. Un comando es una instrucción de un operador de mantenimiento autorizado, y una confirmación es la respuesta del sistema al comando.

Para todos los comandos, el sistema realiza una revisión de los caracteres y del formato.

3.1.2.2. Conexión.

Para poder conectarse al sistema, es necesario entrar enviando caracteres al sistema desde la terminal. Guiado por la velocidad con la que los caracteres son enviados, el sistema determina la velocidad de transmisión. La cadena de caracteres que será enviada al sistema consiste en un conjunto de letras "S". Cuando el sistema ha determinado la velocidad de transmisión, el conjunto de letras "S" es regresada como eco a la terminal y es impresa. La cadena de "S" es terminada por un punto y coma (;) seguida de un 'CR', 'ENTER' o 'SEND' teniendo el mismo significado, sólo dependiendo del teclado de la terminal.

Si una terminal con velocidad baja ha sido conectada al sistema, y la conexión es cambiada a un terminal más veloz, puede traer dificultades para entrar al sistema. En este caso, la tecla "Break" debe ser presionada una vez y después intentar entrar con "S" como se realiza usualmente.

En este punto, el sistema preguntará por una contraseña. Cuando se ingresa una contraseña válida, el sistema responderá imprimiendo la clase de autoridad del operador de mantenimiento, copyright, la identidad de la estación, la fecha, la hora y la posición del equipo a la cual la terminal es conectada. Luego el sistema estará listo para recibir comandos.

Si el sistema recibe una contraseña inválida, este preguntará por otra contraseña. Después de tres contraseñas inválidas consecutivas desde la misma terminal, el procedimiento de entrada debe iniciarse otra vez. Para prevenir que una persona sin autorización entre al sistema, el siguiente intento se debe realizar después de haber pasado 20 minutos. Si el operador de

mantenimiento intenta entrar dentro de los siguientes 20 minutos la cuenta se reiniciará a otros 20 minutos.

El operador de mantenimiento automáticamente será desconectado si la terminal se desconecta después de haber entrado al sistema. Si la terminal se desconecta durante la ejecución de un comando, un nuevo intento de conexión será rechazado hasta que la ejecución del comando haya finalizado.

3.1.2.3. Contraseñas.

El sistema tiene la posibilidad de inicializar 8 palabras de autorización, llamadas contraseñas. Una contraseña consiste en un máximo de 16 caracteres alfanuméricos. A cada contraseña se le asigna un código de autorización de 0 a 7, donde 0 indica la menor y 7 la mayor autorización.

El sistema le asigna al código de autorización 7 la contraseña "HELP" automáticamente cuando este es iniciado por primera vez.

Para que un operador de mantenimiento pueda ejecutar un comando, es necesario que se le haya asignado una contraseña con un código de autorización mayor o igual al que tiene asignado el comando.

3.1.2.4. Ejecución de comandos.

Los comandos enviados al sistema se encuentran en formato MML (man-machine language, es decir, lenguaje hombre-máquina).

Los comandos pueden ser enviados desde tres fuentes distintas, llamadas:

- Terminal de E/S o módem. Mediante cualquier terminal o emulador de terminal dentro de la red de la empresa se podrá realizar una conexión a una dirección IP en modo telnet, o mediante un módem realizando una conexión directa con el conmutador.
- Archivo de comandos en la unidad de disco duro o disco flexible. Es posible almacenar archivos que contengan un conjunto de comandos para realizar cierta operación, estos archivos pueden ser ejecutados mediante un comando o por la inicialización del sistema.
- Internamente en el sistema. A través de cualquier puerto en el sistema utilizando un aparato telefónico especial, con el que generalmente cuenta el operador del sistema.

Se ejecuta una revisión de sintaxis para el comando ingresado. Si no existe error, se revisa una tabla para identificar las unidades de programa de la administración de comandos en el grupo de comandos dado por las primeras dos letras en el comando ingresado.

La unidad de programa que administra el comando toma los parámetros del comando y los analiza. Cuando el análisis de parámetros es exitoso el comando es ejecutado.

3.1.2.5. Sintaxis de los comandos.

Un comando consiste en un código de comando, el cual puede ir seguido de uno o más parámetros.

$$\text{Código_comando:Nombre_parámetro} \left\{ \begin{array}{l} ! \\ ; \end{array} \right\}$$

$$\text{Nombre_parámetro} = \left\{ \begin{array}{l} ALL \\ Param.valor \left[\begin{array}{l} \&Param.valor \\ \&\&Param.valor \end{array} \right] \end{array} \right\}$$

El máximo número de caracteres por línea de comando es 200 y se permiten todos los caracteres del código ASCII.

- [] Indica que esa parte es opcional (puede ser omitida).
- { } Indica que una de las alternativas tiene que ser elegida.
- ... Indica que uno o más valores pueden ser especificados para este parámetro.

3.1.2.6. Caracteres.

El lenguaje usado para la comunicación hombre-máquina en el sistema comprende los siguientes caracteres:

Caracteres comando.

- A-Z Letras mayúsculas.
- 0-F Dígitos hexadecimales.

Caracteres de Edición.

Durante la comunicación con el sistema, los siguientes caracteres pueden ser usados.

- Usado para denotar subelementos en el equipo de hardware.

- , Separador de parámetros.
- & Separa valores de parámetros. Por ejemplo 100&200, significa que 100 y 200 pueden ser usados, pero no los valores entre el rango.
- && Separa los valores iniciales y finales de una serie de valores de parámetros. Por ejemplo 100&&200, significa que todos los valores entre 100 y 200 inclusive pueden ser utilizados.
- ! Si un comando es terminado por un signo de admiración será ejecutado y después se imprimirá para que el operador agregue nuevos parámetros.
- : Separa el código de comando y la parte de parámetros.
- = Asigna un valor a un parámetro.
- /* Indica el inicio de un comentario.
- */ Indica el final de un comentario.
- H' Notación hexadecimal.
- O' Notación octal.
- B' Notación binaria.
- ' El siguiente carácter no será revisado.
- " Indica el inicio y el final de un valor de parámetro que es una cadena de caracteres.

3.1.2.7. Validación de comandos.

Además de la revisión de sintaxis, cada comando es revisado mas a fondo con respecto a:

- Validez de cada carácter.
- Validez del código de comando.
- Si el comando esta libre.
- Si el comando cambiará datos, se ejecuta una revisión para asegurar que otro comando como tal no esta en proceso.
- Número de parámetros.
- Si los valores de los parámetros corresponden a los tipos del comando.
- Si los valores de los parámetros están dentro de los limites permitidos.

3.1.2.8. Estructura de comandos.

Un comando consiste de cinco caracteres, XXYYZ.

XX	Indica el grupo de comandos al que pertenece.
YY	Indica la función que se ejecutará.
Z	Indica el código de operación.

Los siguientes son los códigos de operación existentes:

- C – Cambio.
- E – Fin.

- I – Inicialización.
- L – Carga.
- P – Imprimir.
- R – Remover.
- S – Asignar, afectar.
- T – Transferir.

3.1.2.9. Interrupción de un comando.

Durante la transmisión de datos en cualquier dirección, los operadores autorizados de mantenimiento tienen la posibilidad de enviar una señal de interrupción al sistema, la cual provoca que el sistema inmediatamente esté listo para recibir comandos.

La interrupción del comando es iniciada presionado la combinación de teclas CTRL-X. El sistema confirmará enviando los caracteres:

- ?
- <

Cuando la señal de interrupción ha sido inicializada las siguientes posibilidades son disponibles:

- B; 'CR' Para interrumpir una impresión.
- ; 'CR' Para continuar imprimiendo.
- F; 'CR' Para interrumpir un comando que ha sido previamente aceptado. Si el comando ha sido parcialmente ejecutado, los cambios realizados en los datos permanecerán.

3.1.2.10. Comandos peligrosos.

Un comando es clasificado como "PELIGROSO" si afecta el funcionamiento del sistema, por ejemplo, remover una extensión. Los comandos peligrosos son precedidos del siguiente despliegue:

SURE?(YES/NO)
<

Para aquel comando que será ejecutado, el operador de mantenimiento debe responder "Y" si esta seguro de la ejecución del comando, o "N" seguido de un retorno de carro ("CR") si no lo está, para que se ejecute o no el comando.

3.1.2.11. Comandos multilínea.

Cuando un comando con sus parámetros excede una línea de longitud, la línea es terminada por un valor de parámetro completo seguido de una coma (,). Después presionar la tecla de retorno de carro ('CR') para continuar con el comando.

Ningún comando puede exceder 2 líneas (200 caracteres) de longitud.

3.1.2.12. Comentarios.

Un comentario inicia con los caracteres `/*` y termina con los caracteres `*/`. Un comentario no debe exceder de dos líneas y no debe ser mezclado con comandos en la misma línea.

Los siguientes son unos ejemplos de comentarios:

```
/* Iniciar un número de serie individual para el operador del PBX */
NANSI:NUMTYP=OI,NUMSE=500&&502;
/* Inicializar operadores del PBX 500 y 501 */
OPERI:DIR=500&&501,EQU=1-0-41-0,TRAF=1515,OPC=0,PRG=1;
```

3.1.2.13. Despliegue o impresión.

El sistema tiene cuatro diferentes tipos de despliegue:

- Despliegue de verificación.
Un despliegue de verificación consiste de un comando que ha sido regresado como eco a la terminal ya que el comando es clasificado como peligroso.
- Despliegue de procedimientos.
Este tipo de procedimientos son usados para informar al operador de mantenimiento que comandos han sido ejecutados, parcialmente ejecutados o no ejecutados por completo. Si el comando ha sido aceptado, los siguientes despliegues pueden ocurrir.
EXECUTED – El comando ha sido ejecutado con éxito.
ORDERED – El comando ha sido aceptado por el sistema pero no ha sido ejecutado aún. El resultado del comando será impreso en la terminal que realizó la solicitud cuando se termine la operación.
NOT ACCEPTED – El comando se especificó erróneamente. No se realiza ninguna acción.
PARTLY EXECUTED – La acción especificada ha sido parcialmente ejecutada, pero la detección de un error ha interrumpido la ejecución.

Los mensajes NOT ACCEPTED y PARTLY EXECUTED los acompañará un despliegue especificando el error.

- Despliegue de respuesta inmediata.
Este tipo de despliegues se dan tan pronto el sistema haya aceptado el comando. Si el tiempo de respuesta para el despliegue es largo, el mensaje WAIT será desplegado.
Los despliegues de respuesta inmediata terminan con END.
- Despliegue de resultados.
Los despliegues de este tipo son aquellos que no aparecen inmediatamente cuando un comando es iniciado.
Los despliegues de resultados pueden aparecer después de que el operador ha terminado su sesión. Sin embargo, este puede aparecer dentro de la conexión con otro comando, si la tarea solicitada ha sido terminada.
El despliegue de resultados inicia con SPONTANEOUS PRINTOUT y termina con END.

3.1.2.14. Terminar la sesión.

La desconexión se puede ejecutar manual o automáticamente.

Desconexión manual.

Presionar la combinación de teclas "CTRL-X" de modo que se termine el administrador de comandos. El sistema responde con:

?

<

Teclear 0;'CR', a lo que el sistema responderá con:

"LOGGED OFF"

Desconexión automática.

Si no ha sido ingresado ningún comando durante un periodo de 10 minutos el sistema imprimirá lo siguiente:

"TIME OUT WARNING"

Si no se ingresa ningún comando en los siguientes 10 minutos, la terminal automáticamente será desconectada y el sistema imprimirá lo siguiente [ERICSSON, 1998]:

"LOGGED OFF"

3.1.3. Formato del registro de la llamada terminada.

La siguiente tabla muestra la descripción del registro enviado por el PBX a través de su tarjeta SIU (System Interface Unit) al finalizarse una llamada.

Descripción	Posiciones
Espacio	0-2
Fecha Fin	3-6
Hora Fin	7-10
Espacio	11
Tiempo de duración de la llamada en segundos	12-15
Espacio	16-17
Pulso "Tax"	19-21
Espacio	22
Código de condición	23-24
Espacio	25
Código de Acceso 1	26-30
Código de Acceso 2	31-32
Código de Acceso 2/ ISDN CBC	33-35
Espacio	36
Número marcado	37-56
(puede incluir el tiempo de cola de operadora)	37-38
Espacio	57
Extensión origen	58-67
Espacio	68
Código de cuenta	69-83
Espacio	84
Código de autorización	85-90
Espacio	91
Código de selección de red de tránsito (ISDN)	92-94
Espacio	95
Código de acceso del operador del sistema (ISDN)	96-97
Espacio	98
Tiempo de cola para la línea externa ocupada	99-100
Espacio	101
Identificador de línea externa	102-107
Espacio	108
Retorno de carro (CR)	109
Alimentación de línea (LF)	110-112
Espacio	113

Tabla 3.1. Formato del registro de la llamada terminada.

3.2. Desarrollo Rápido de Aplicaciones.

3.2.1. Definición.

El Desarrollo Rápido de Aplicaciones (RAD) se refiere a un ciclo de vida de desarrollo de sistemas que presenta un desarrollo mucho más rápido y con resultados de más alta calidad que el ciclo de vida de desarrollo de sistemas tradicional. [MARTIN, 1991]

La metodología de Desarrollo Rápido de Aplicaciones provee de medios para desarrollar sistemas rápidamente mientras se reducen los costos y se aumenta la calidad. Esto mediante la automatización de grandes porciones del ciclo de vida del desarrollo de sistemas, imponiendo rígidos límites en los tiempos de desarrollo y reutilizando componentes ya existentes. [MARTIN, 1991]

Algunas de las técnicas utilizadas por el RAD son:

- El uso de evolución de prototipos que se convierten en el sistema final.
 - La utilización de Talleres en vez de entrevistas para recolectar requerimientos y revisar el diseño.
 - El uso de un conjunto de herramientas que soporten la modelación, el prototipo, y código reusable.
 - La utilización de tiempos rígidos para la terminación de tareas o etapas.
- [MARTIN, 1991]

3.2.2. Fases del RAD.

La metodología de desarrollo rápido de aplicaciones (RAD) cuenta con 4 fases principales, cada una de las cuales tiene ciertas actividades a realizar y se obtienen ciertos resultados. Estas fases son:

- Fase de planeación de requerimientos, mejor conocida como JRP (siglas del inglés Joint Requirements Planning, que significa en español Planeación de Requerimientos en Grupo).
- Fase de diseño con el usuario, también conocida como JAD (siglas del inglés Joint Application Design, cuyo significado es Diseño de la Aplicación en Grupo).
- Fase de Construcción.
- Fase de Liberación.

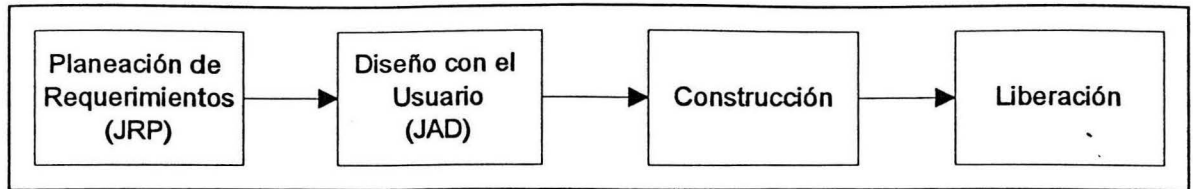


Figura 3.2. Fases del Desarrollo Rápido de Aplicaciones (RAD).

3.2.2.1. Planeación de requerimientos (JRP).

Esta es la fase inicial del ciclo de vida del RAD, en esta fase se hace una recopilación de todos los requerimientos del sistema, esta es elaborada a través de reuniones con los usuarios finales.

Una sesión de JRP es normalmente corta y sin detalles técnicos. En estas sesiones se establecen los requerimientos y la justificación del sistema así como las funciones detalladas que debe ejecutar este. [MARTIN, 1991]

La figura 3.3 muestra detalladamente los cinco pasos para el desarrollo del JRP, que son inicio, investigación, preparación de las sesiones del JRP, conducción de las sesiones del JRP y finalización.

El JRP tiene como resultados finales los siguientes:

- Lista de los departamentos que harán uso del sistema.
 - Lista de los objetivos del sistema.
 - Detalles de las posibles funciones del sistema:
 - Lista de las posibles funciones del sistema.
 - Lista de los beneficios de cada función (tangibles e intangibles).
 - Asignar prioridades a las funciones (utilizando tres o más categorías)
 - Cuáles funciones deben estar presentes en la primera versión del sistema.
 - Cuáles funciones pueden estar presentes en la primera versión del sistema si se pueden construir rápidamente.
 - Cuáles funciones podrían ser desarrolladas en una siguiente versión del sistema.
 - Un diagrama de descomposición de procesos del sistema.
 - Un diagrama de flujo de procesos del sistema.
 - Un diagrama de flujo que muestre las interfaces con otro sistema.
- [MARTIN, 1991]

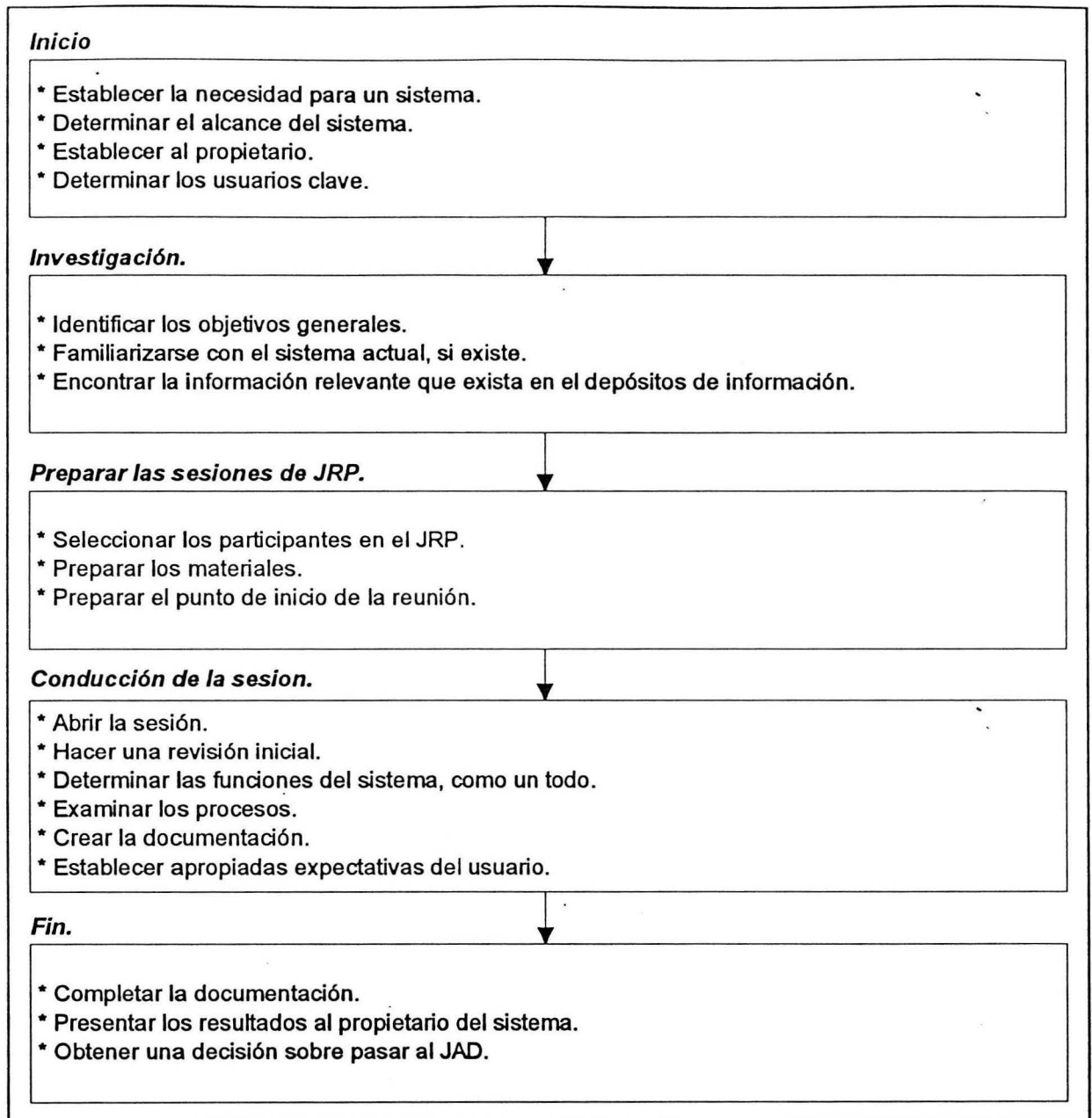


Figura 3.3. Pasos del JRP.

3.2.2.2. Diseño con el usuario (JAD).

Esta fase es la siguiente a la fase de JRP, en esta fase se realiza el diseño detallado del sistema, y es llamada diseño con el usuario porque es el usuario quien está muy involucrado en esta etapa, generalmente mediante las técnicas de prototipo.

El JAD también se lleva a cabo generalmente en dos largas sesiones. En la primer sesión de JAD se establece el diseño detallado del sistema, que consiste en un modelo de datos y un modelo de procesos, especificaciones detalladas, diseño de pantallas y de reportes, y posiblemente sencillos prototipos. En la segunda sesión se revisan los prototipos, se solidifica el diseño y se crea la documentación [MARTIN, 1991]

La figura 3.4 muestra detalladamente los pasos para el desarrollo del JAD.

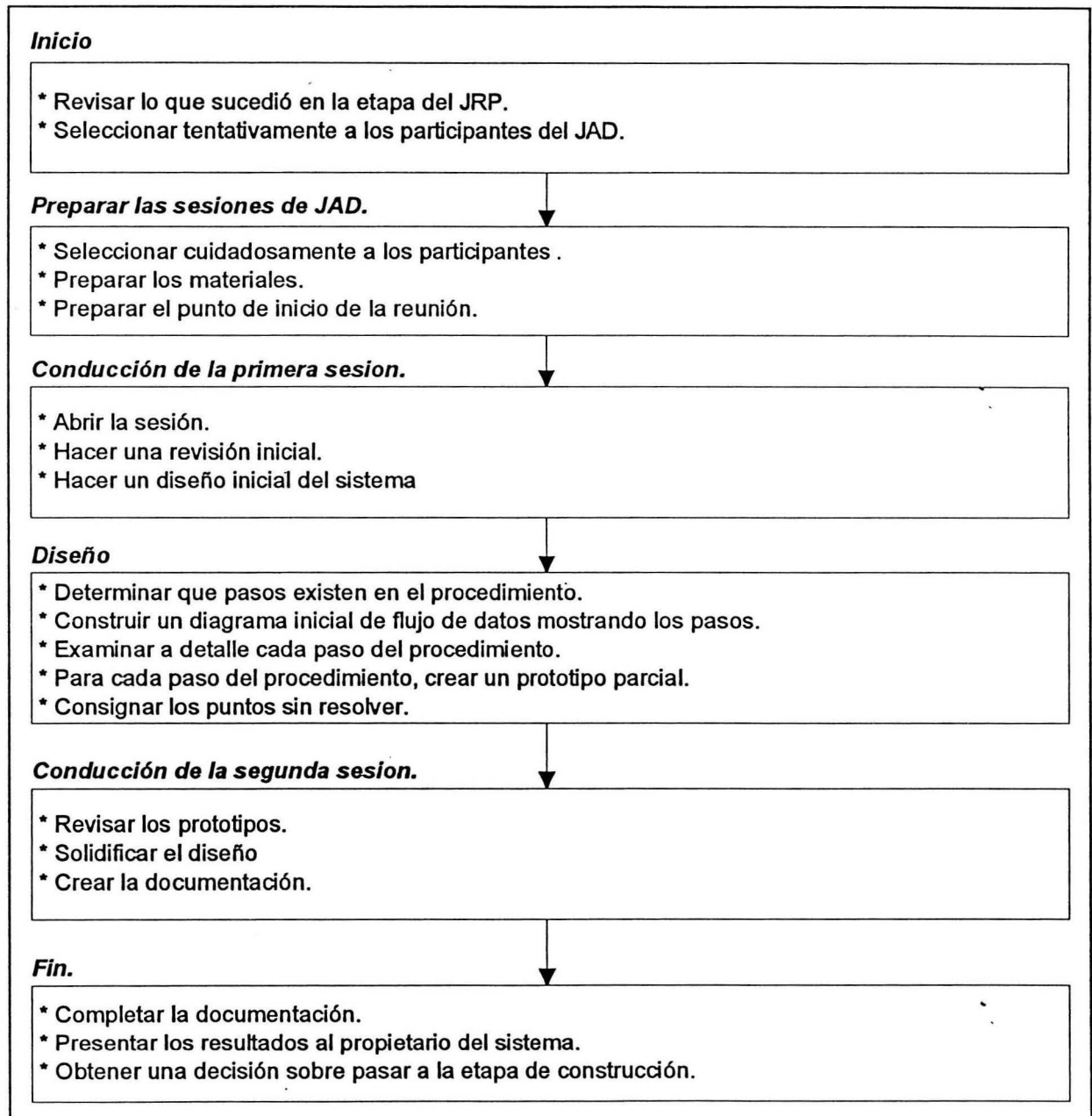


Figura 3.4. Pasos del JAD.

Los resultados finales del JAD son los siguientes:

- Arquitectura del sistema.
- Diseño de los procedimientos del sistema.
- Diseño de las interfases y los reportes del sistema.
- Modelo de datos.

3.2.2.3. Construcción.

En esta etapa se desarrollan físicamente los resultados de la etapa de JAD. Se construye físicamente la base de datos, se realiza la programación de los procedimientos, las interfases y los reportes del sistema. [MARTIN, 1991]

Los usuarios finales continúan involucrados en esta etapa, validando el diseño de pantallas y la información arrojada por los procesos. A lo largo de esta fase se van realizando las pruebas del sistema y a la vez se documentan. [MCCONNELL, 1996]

3.2.2.4. Liberación

En esta fase se realizan pruebas integrales al sistema, se da capacitación a los usuarios, se realizan los cambios organizacionales y de operación.

3.2.3. Beneficios del RAD.

Algunos de los beneficios que se obtienen al utilizar esta metodología son:

- Acerca a los altos ejecutivos al proceso de planeación del sistema.
- Liga la planeación del sistema con el análisis de metas, problemas, factores críticos de éxito y oportunidades estratégicas de sistemas.
- Fomenta la lluvia de ideas en lo que se refiere a como deben ser las funciones más importantes del sistema.
- Elimina funciones de valor cuestionable.
- Fomenta la creatividad de los ejecutivos para pensar acerca de cómo pueden utilizar los sistemas de información para elevar las oportunidades de negocio.
- Ayuda a obtener los requerimientos y el diseño correctos desde la primera entrevista. Es caro y peligroso cambiar los requerimientos después de que un sistema ha sido diseñado o implementado.
- La especificación y el diseño de sistemas toma mucho menos tiempo que con el análisis de sistemas tradicional.
- La productividad del proceso de desarrollo aumenta sustancialmente.
- Acerca a los usuarios finales al proceso del diseño y ayuda a eliminar la insatisfacción.

- Los sistemas resultantes generalmente tienen alta calidad y gran valor en el negocio.
- Ayuda a producir una comunidad de usuarios con una gran instrucción computacional. Esto hace que los usuarios sean más imaginativos e inventivos acerca de crear mejores procedimientos.
- Ahorra dinero evitando la necesidad de reprogramar o modificar sistemas con un diseño inadecuado. Los gastos en mantenimiento son menores.
- Ayuda a integrar y unificar las necesidades de distintas partes de la organización.
- Evita que los analistas de sistemas se encuentren en situaciones de tener que resolver conflictos políticos entre los usuarios finales.
- Aumenta la satisfacción del usuario. Debido a que los usuarios diseñaron el sistema, ellos toman interés en él, se sienten propietarios y ayudan en la fase de construcción. [MARTIN, 1991] [MCCONNELL, 1996]

3.2.4. La técnica de prototipo.

El prototipo es una técnica para construir una rápida versión de un sistema deseado o partes de ese sistema. El prototipo ilustra el sistema a los usuarios y diseñadores. Este les permite ver los defectos y proponer nuevas formas de mejorar el sistema. Este sirve como un vehículo de comunicación permitiendo a las personas que requieren el sistema que revisen la interacción prevista del usuario con el sistema. Por lo que el prototipo es mucho más efectivo que revisar especificaciones en papel. [MCCONNELL, 1996]

La técnica de prototipo cuenta con las siguientes etapas: recolección y refinamiento de requisitos, diseño rápido, construcción del prototipo, evaluación del prototipo por el usuario, refinamiento del prototipo y producto de ingeniería. Estas etapas se muestran en la figura 3.5.

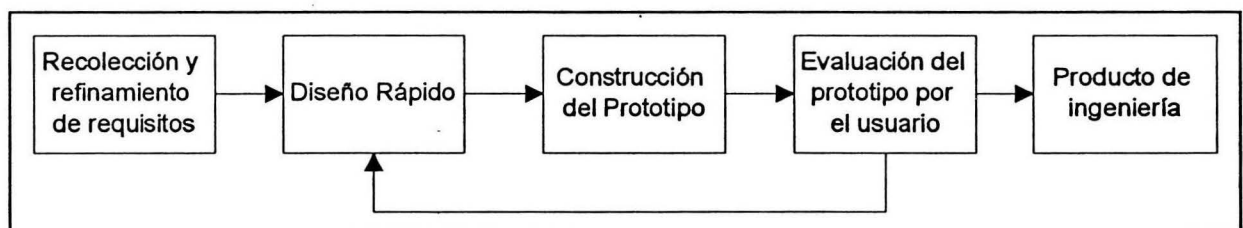


Figura 3.5. Etapas de la técnica de prototipo.

El prototipo en el desarrollo rápido de aplicaciones es fundamentalmente diferente del prototipo en otros ciclos de vida de desarrollo de sistemas, en los que los prototipos deben ser parte de la evolución del sistema. Algunas veces los prototipos son desarrollados con una herramienta diferente de la herramienta de desarrollo final y son eventualmente desechados. En el ciclo de vida del desarrollo rápido de aplicaciones los prototipos deben ser contruidos como la

herramienta de desarrollo final, de modo que puedan pasar directamente de la fase del JAD a la fase de construcción. [MCCONNELL, 1996]

3.3. Active Server Pages.

3.3.1. Las Páginas Activas de Servidor (ASP).

Las Páginas Activas de Servidor (del inglés Active Server Pages, ASP) es una extensión de ISAPI (Internet Server Application Programming Interface), la cuál se construye en el tope de la infraestructura del ISAPI para proveer un esqueleto de la aplicación del lado del servidor, haciendo más fácil la construcción de aplicaciones de Web dinámicas. [FRANCIS, 1998]

Un documento ASP puede contener sintaxis HTML y scripts de servidor. Cuando el servidor de Web recibe la petición HTTP para el documento ASP, se genera en memoria un archivo virtual de salida HTML usando una combinación de información HTML estática y HTML generado por el programa script. Detallando, inicialmente el servidor de NT, el cual es un servidor de Web debido a que tiene un servicio de IIS (Internet Information Service), recibe una petición http para un documento ASP. Esta petición es recibida por la interfase de Web del IIS y se envía la petición del documento al procesamiento de ASP. Este toma el documento del repositorio y lo ejecuta mediante los interpretadores de scripts obteniendo información de las bases de datos mediante los ADO (ActiveX Data Objects) y los ODBC (Open Database Connectivity) y a la vez generando el documento HTML. Al finalizar la ejecución del script se ha terminado de generar el documento HTML y es enviado a la interfase de Web para que sea recibido por el cliente y mostrado en su navegador (Figura 3.5.). [FRANCIS, 1998]

Las páginas activas cuentan con la habilidad de incluir en el programa script un gran número de objetos intrínsecos de ASP, los cuales automáticamente manejan muchas de las tareas básicas de desarrollo de sistemas, y por lo tanto simplifican la cantidad de desarrollo requerido. Estos objetos dan a las ASP el concepto de una sesión de usuario y permiten que las variables persistan a través de las páginas de Web, hasta que la sesión es abandonada por programa o el navegador es cerrado. Antes de las ASP, el manejo del contexto usuario a través de muchas peticiones http era un proceso complejo. [FRANCIS, 1998]

Además, el poder real de los scripts de las ASP está en su habilidad de hacer interfase con componentes de software tipo COM, ya sean los provistos por Windows NT o los desarrollos de otras compañías de software. [FRANCIS, 1998]

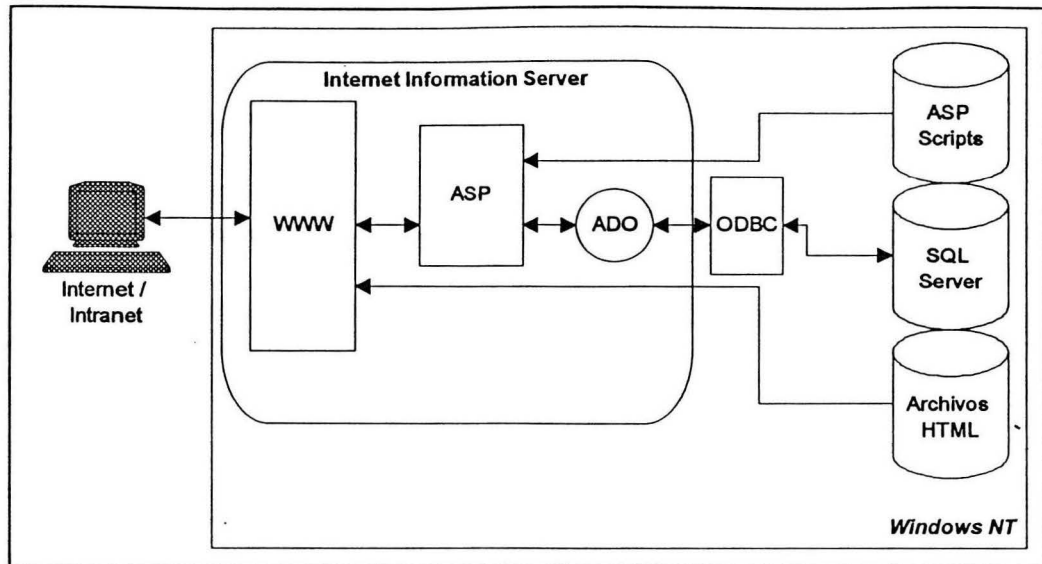


Figura 3.6. Modelo de las Active Server Pages.

3.3.2. Sintaxis de las ASP.

Las ASP son un ambiente que permite el procesamiento scripts incluidos en páginas HTML. Un script es una especie de lenguaje de programación que se emplea para tener el control en un ambiente host. El script no es compilado, sino interpretado. Es por esto, que el script debe ejecutarse en el ambiente que contiene el interprete del script, generalmente en el server.

Para que el documento ASP pueda ser identificado como tal, además de la extensión es necesario incluir la siguiente línea de comandos, con la cual se indica que se va a utilizar un lenguaje script por omisión.

```
<%@ Language=VBScript %>
```

Al igual que HTML, las ASP también hacen uso de etiquetas o tags. Los tags utilizados para que el interpretador identifique el lenguaje script son `<%` y `%>`. Por ejemplo para asignar un valor a una variable la instrucción seria como sigue:

```
<% variable = "valor" %>
```

Lo anterior hace que el interpretador del script realice la asignación. El lenguaje script también utiliza otro tipo de delimitadores que indican que el valor en el lenguaje script se imprima o despliegue el archivo HTML que es generado al interpretarse la ASP. Estos tags son `<%=` y `%>`. Por ejemplo si se desea colocar el valor de una variable como la leyenda de una liga seria como sigue:

```
<a href="http://www.udem.edu.mx"><%= variable %> </a>
```

Lo anterior entonces aparecería en el archivo HTML generado como:

```
<a href="http://www.udem.edu.mx"> valor </a>
```

3.4. NetCharts.

NetCharts es un conjunto de applets de Java que permite a los desarrolladores de Web y a los programadores de Java crear rápida y fácilmente gráficas en línea desde casi cualquier fuente de datos, con una codificación mínima.

3.4.1. Características.

- Es posible graficar simultáneamente en una misma gráfica múltiples conjuntos de datos de múltiples fuentes de datos.
- Las gráficas pueden ser desplegadas como applets, librerías de clases o beans.
- Se soportan graficas de los siguientes tipos: barra, pie, tira, caja, diagrama, tiempo, combinación, XY, valores.
- Características drill-down en cualquier parte de la gráfica.
- Es posible desplazarse, acercarse y alejarse de la gráfica.
- Cuenta con etiquetas dinámicas de estilo pop-up.
- Las graficas de pie se pueden rotar.
- Maneja gráficas en 2 y 3 dimensiones.
- Cuenta con texto internacionalizado.
- Realiza las escalas de las gráficas automáticamente. [VISUAL MINING,1996]

3.4.2. Beneficios.

- Provee un sistema rápido y fácil para la construcción sofisticada de gráficas en línea.
- Reduce el tiempo y costo de construcción de gráficas en línea.
- Las gráficas pueden ser dinámicamente actualizadas en tiempo real desde casi cualquier fuente de datos.
- Permite explotar el Internet para diseminar información poderosa para ayudar a la administración de los procesos de negocio y promover los productos y servicios.
- Se ejecuta en cualquier plataforma que soporte Java incluyendo UNIX, Windows NT, Windows, Macintosh, OS/2, PC's de red y estaciones de Java. [VISUAL MINING,1996]

3.4.3. Programación.

NetCharts da a los desarrolladores la habilidad de crear y manipular la demanda de gráficas a través de un vasto conjunto de atributos y comandos llamados CDL (del inglés Chart Definition Language, cuyo significado es Lenguaje de Definición de Gráficas). Con el uso de estas herramientas es posible ver el diseño de la gráfica, lo que evita programar sin observar los resultados. [VISUAL MINING,1996]

3.4.4. Integración.

NetCharts puede ser incorporado en documentos HTML, DHTML, XML, en herramientas de desarrollo de aplicaciones y en herramientas de desarrollo de aplicaciones de Web, como las siguientes: Visual Basic, Visual InterDev, Cold Fusion, Silver Stream, NetDynamics, Inprise Jbuilder, IBM Visual Age, Java Workshop, Visual Café. [VISUAL MINING,1996]

El siguiente es un ejemplo de una gráfica de barras incorporada a un documento HTML.

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Response.buffer = True %>
<HTML>

<center>
    <applet name=barchart
        codebase=../Netcharts/classes
        code=NFBarchartApp.class
        width=762 height=600>
        <param      name=NFPParamScript      value = '
Background      = (white, NONE, 8);
BottomTicks     = ("ON", black, "Tahoma", 12,90);
LeftTicks       = ("ON", black, "Tahoma", 12);
LeftFormat      = (FLOAT);
DwellLabel      = ("", black, "Tahoma", 20);
DwellBox        = (white, SHADOW, 3);
BarLabels       = <%=Barras %>;
BarWidth        = 80;
LeftTitle       = ("<%=Unidad %>", black, "Tahoma", 20,90);
LeftTitleBox    = (white, NONE, 5);
Legend          = ("", black, "Tahoma", 14);
LegendBox       = (white, NONE, 5);
LegendLayout    = (HORIZONTAL,BOTTOMRIGHT,0,0,LEFT,2);
```

```

GraphType = GROUP;
Bar3DDepth = 85;
DataSets = <%=ConDatos %>;
<%=ArrDatos %>
'>
</applet>
</center>
</HTML>

```

La siguiente figura muestra la gráfica en el navegador de Internet.

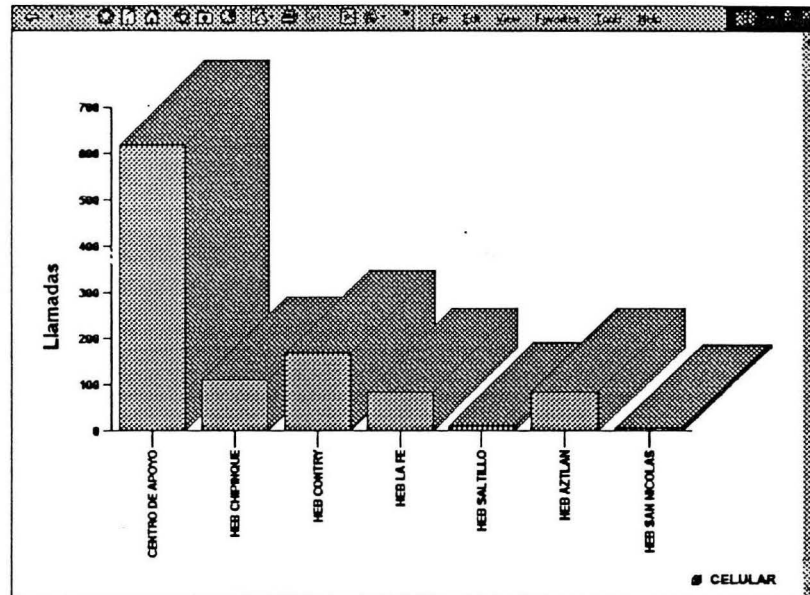


Figura 3.7. Ejemplo de gráfica con NetCharts.

3.4.5. Especificaciones.

NetCharts soporta cualquier plataforma Java, incluyendo PC's, UNIX y computadoras en red. Todos los applets pueden ser desplegados en cualquier navegador de Web compatible con Java y soporta el uso de la interfase JDBC para cargar definiciones de parámetros o valores de datos de uno o más servidores de bases de datos, como Oracle, Sybase, Informix, DB2, SQL Server, dBase, Access y Excel. [VISUAL MINING, 1996]

3.5. Visual Basic vs. Java.

3.5.1. Java.

3.5.1.1. Antecedentes.

Java fue desarrollado por Sun Microsystems en 1991. Este era parte de un proyecto de investigación de desarrollo de software para el control de dispositivos electrónicos dirigidos al consumidor final. Inicialmente los investigadores de este proyecto desarrollaron un prototipo de dispositivo llamado Star7, para el cual se pensaba desarrollar su sistema operativo en C++. Sin embargo, uno de los miembros del proyecto, James Gosling, no estaba satisfecho con el desempeño de C++ y se dedicó a la creación de Java. [LEMAY, 1999]

Originalmente Java no funcionó como herramienta para desarrollo de dispositivos electrónicos, pero funcionó muy bien para Web debido a que es pequeño, seguro y portable. Además, podría ser usado como lenguaje de propósito general para desarrollar programas que se pudieran ejecutar en distintas plataformas. [LEMAY, 1999]

En 1994, se creó el navegador HotJava que podía ejecutar applets de Java, esto con el fin de demostrar el potencial de Java. A pesar de que HotJava fue aceptado por la comunidad Web, Java no tomó fuerza sino hasta que Netscape licenció el lenguaje en 1995. [LEMAY, 1999]

3.5.1.2. Características.

Java es apropiado para el diseño de programas que funcionen en conjunto con Internet. También hace uso de la metodología orientado a objetos, cuya utilidad es creciente en el diseño de software. Además, Java es un lenguaje multiplataforma, por lo que puede ser ejecutado tanto en Windows, Apple o UNIX.

Java se parece más a lenguajes de programación populares como C++, Visual Basic y Delphi, que a HTML o JavaScript.

Java es mucho más conocido por su capacidad de correr en páginas de Web. Los navegadores de Web pueden obtener un programa de Java, llamado applet, y ejecutarlo localmente en el navegador del usuario. [LEMAY, 1999]

Aunque los applets son probablemente el uso más popular de Java, son un modo de usar el lenguaje. Como Visual C++, Visual Basic y Delphi, Java es un

lenguaje robusto para desarrollar una amplia gama de programas, soportar interfases gráficas de usuario, conectividad de redes, conectividad de bases de datos y otra funcionalidad sofisticada. [LEMAY, 1999]

3.5.2. Visual Basic.

3.5.2.1. Antecedentes.

Durante mucho tiempo sólo se dispuso de unas pocas herramientas para el desarrollo en las aplicaciones Windows. Antes de que Visual Basic fuera introducido en 1991, el desarrollo de las aplicaciones para Windows era mucho mas complicado que desarrollar aplicaciones para DOS. Los programadores tenían que preocuparse más de lo que estaba haciendo el ratón, de dónde estaba el usuario dentro de un menú y si se estaba realizando un clic o un doble-clic en un punto determinado. [CORNELL, 1997]

Fue entonces que surgió Visual Basic como un lenguaje de programación para Windows. Es derivado de las versiones de DOS de Basic, QBasic y QuickBasic. Fue originalmente construido como un servidor OLE, y fue uno de los primeros lenguajes en soportar componentes como los VBX. [CORNELL, 1997]

Actualmente con la última versión de Visual Basic las aplicaciones sofisticadas de Windows 9.x, Windows NT e incluso Windows 3.x pueden realizarse ahora en una fracción del tiempo necesario anteriormente. Los errores de programación no suceden con la misma frecuencia de antes y, en caso de suceder, son mucho más fáciles de detectar y solucionar. [CORNELL, 1997]

3.5.2.2. Características.

Conforme se desarrolló Visual Basic, desde la versión 1.0 hasta la actual versión 6.0 muchas cosas han sido mejoradas. La interfase se ha convertido en un ambiente muy poderoso de diseño. Conforme las tecnologías han cambiado, Visual Basic también, y ahora es posible desarrollar aplicaciones de alta velocidad, servidores OLE, controles ActiveX, objetos COM, aplicaciones DHTML y muchas cosas más. [GUREWICH, 1998]

Lo que ha hecho a Visual Basic uno de los lenguajes de programación más populares es probablemente la facilidad de crear. Todos los controles están previamente dibujados, solo es necesario agregar el código y compilar. No es necesario compilar las aplicaciones en EXE's, Visual Basic cuenta con un veloz "Ejecutador".

En la última versión de Visual Basic es posible compilar las aplicaciones para crear código maquina optimizado. Este código maquina se ejecuta mucho más

rápido que el código interpretado. Sin embargo, aun se cuenta con la opción de compilar al formato interpretado por Visual Basic, el código-p. Una nueva opción que puede ser de gran utilidad es la de mantener en memoria la aplicación hasta que el proceso termine, lo que puede ayudar realmente al desempeño de las aplicaciones. [GUREWICH, 1998]

3.5.3. Comparación de Java y Visual Basic.

Una de las ventajas esenciales de Java y Visual Basic es que los programadores pueden desarrollar un amplio rango de programas mucho más rápido que con otros lenguajes. Con relación a otros lenguajes de 3ª generación, como Ada, C/C++, Cobol, Fortran y Pascal, Java y Visual Basic tienen como ventajas:

- Nuevos, uniformes y completos API's para cada aspecto de la programación en Web, cliente/servidor, y n-capas.
- Diversos modos de despliegue, incluyendo: EXE's, componentes, servlets y applets.
- Ambientes visuales de desarrollo superiores, y con muchos proveedores de herramientas y componentes.

Además, el desempeño de ambos se ha mejorado constantemente hasta el punto de aproximarse del 10 al 30% de los programas más rápidos desarrollados en C/C++, Cobol, Fortran o Pascal. Finalmente, Java en particular, pero también VB, pueden ser usados para distribuir programas de Web. [KERNIGHAN, 1998] [SURVEYER, 1999]

Java y Visual Basic han venido destacando como lenguajes de programación porque fueron construidos para ajustarse al nuevo molde de desarrollo de sistemas, con un desarrollo rápido a través de sus ambientes de desarrollo interactivos y completos API's para abarcar los últimos requerimientos de programación, especialmente el Web. [SURVEYER, 1999]

Tal vez los métodos orientados a objetos y código de semi-componentes sean aun más importantes, estos pueden ejecutarse individualmente o ser ligados de muchas maneras a programas existentes y fuentes de datos, como servlet, aplicación de servidor o applet de Web. Esta versatilidad de despliegue es lo que hace tan atractivos a estos lenguajes. [SURVEYER, 1999]

Superficialmente, Java y Visual Basic tienen algunas características en común. Ambos son fáciles de aprender y es fácil empezar a hacer desarrollos útiles. Además, tienen completos API's con lo último en desarrollo de Web y n-capas. Realmente si no es posible encontrar lo necesario directamente en el lenguaje, existe una gran cantidad de herramientas de terceras partes, utilerías y componentes disponibles para completar su efectividad.

Sin embargo, existen tres grandes diferencias entre los dos lenguajes. Primero, Java ha sido construido desde sus raíces como un lenguaje orientado a objetos, mientras que Visual Basic está gradualmente adquiriendo capacidades completas de OO. Segundo, Java tiene algunas características de seguridad y confiabilidad construidas dentro del corazón de su diseño.

Pero la tercera, y tal vez su diferencia más crítica, es que Java tiene plataforma cruzada o multiplataforma, puede ejecutarse en cualquier combinación de hardware y sistema operativo, mientras que Visual Basic se ejecuta solamente en Windows, ya sea Windows 9x, NT, 2000, y CE. Sin embargo, Visual Basic no llega a soportar Windows 3.x, mientras que Java si lo hace. [SURVEYER, 1999]

Por otro lado, Visual Basic se ha convertido en el lenguaje para las aplicaciones de Windows, a través de VBA. Esto es importante porque cada vez más sistemas serán construidos directamente en aplicaciones clave como AutoCAD, Excel, Notes, Project, SAS, SAP, Word y otros cientos de programas de Windows.

Si se compara Java y Visual Basic en detalle, existen cuatro grandes criterios que se pueden considerar. El primero es la facilidad de desarrollo. Es donde VB parece que tiene ventaja. Sin embargo, como Java, Visual Basic ha crecido y tiene grandes cantidades de API's. Para poder agregar nuevas características de orientado a objetos y otras funcionalidades, VB ha tenido que extender su sintaxis básica por docenas de comandos. Aún peor, la documentación para Visual Basic se ha deteriorado en las recientes ediciones. La explicación de los comandos se extiende sobre tres largos manuales, cuya versión impresa cuesta más de \$100 dls.. Los ejemplos rápidos de cómo usar la gran cantidad de comandos y funciones de VB han decaído aun en la documentación electrónica. [SURVEYER, 1999]

Java es sólo levemente mejor en este aspecto. Afortunadamente, la gran cantidad de API's de Java están organizados en completas referencias. Pero como VB, los usuarios tienen que buscar en terceras partes para obtener documentación con ejemplos rápidos de código de Java. El problema fundamental es que aunque los desarrolladores de ambos lenguajes han trabajado fuertemente para lograr hacerlos fácil de acceder y de aprender, el tamaño absoluto y alcance de su funcionabilidad los hacen difíciles. [SURVEYER, 1999] [KERNIGHAN, 1998]

Es posible concluir que Java es mejor y es el lenguaje de programación que puede dar mejores resultados, por su ejecución en multiplataformas, su tecnología orientada a objetos, y por sus características de seguridad. Sin embargo, hay un criterio más que considerar, si la red de la empresa consiste solamente de servidores NT y clientes Windows o si la red tiene recursos que no están relacionados con Microsoft. Para el primer caso Java es una opción pobre. No se explotaría su capacidad multiplataforma, su curva de aprendizaje es

mucho mayor y además el desempeño de Visual Basic contra el de Java en un ambiente puramente Windows es mejor. Pero, en el segundo caso, Java es la mejor opción a largo plazo para el desarrollo de sistemas en la empresa.

A continuación se presenta en una tabla un análisis comparativo entre ambos lenguajes basados en las características mayormente evaluadas para cualquier lenguaje.

Característica	Java	Visual Basic
Facilidad de desarrollo	+ Muy buenos ambientes visuales de desarrollo. + Gran cantidad de <i>wizards</i> y ayuda.	+ Muy buenos ambientes visuales de desarrollo. + Gran cantidad de <i>wizards</i> y ayuda.
Número de elementos de sintaxis / rutinas	50 comandos / miles de rutinas.	Más de 300 comandos / miles de rutinas.
Confiabilidad y estabilidad	- Desventaja por sus constantes y rápidos cambios.	- Desventaja por sus constantes y rápidos cambios.
Mayores debilidades	- Numerosas nuevas clases o API's. - Rápida desaprobación de clases. - Los API's son mucho más grandes que los de VB.	- Numerosas nuevas rutinas o sintaxis. - Apenas inicia a moverse hacia la tecnología OO. - Siempre existen errores en nuevas versiones. - Las rutinas de acceso a datos son un llo.
Robustez	+ Limpio manejo de excepciones y modelos para espera de eventos. + Revisión de límites. + Manipulación de memoria restringida.	+ Revisión automática de sintaxis en el editor. + Uso directo de API's de Windows. - Abierto para dirigir cambios de la memoria.
Seguridad	++ Mayor ventaja con su modelo de seguridad.	- Los ActiveX siguen siendo problemáticos.
Completa confiabilidad	+ Razonablemente confiable al inicio. - Errores en la versión 1.1 del SDK. + Problemas de compatibilidad en el GUI (Graphical User Interface). + Mayores ISV's (Independent Software Vendor) como Oracle, IBM, Sun, Sybase, etc. confían en él.	+ Seis versiones para limpiar errores. - Continúan islas de inestabilidad o errores. - Difícil de encontrar buenos consejos para evitar o corregir ciertos errores. - La herramienta favorita de Microsoft.
Prueba y depuración	= Es buena la depuración del lado del cliente. = Bien pero mezclado con la depuración en el servidor. = Buenas herramientas de prueba.	++ Mejor depurador en el cliente. = Perfeccionando en el lado del servidor. = Algunas buenas herramientas de prueba.
Tamaño del proyecto	+ De mediana escala con DBMS. + Servidor o n-capas gran escala / Web	+ Pequeño, rápido y sucio. + Escala mediana con muchos GUI's.
Otras	+ Su ventaja es que es mas seguro y tiene componentes de gran alcance como los JavaBeans, EJB's (Enterprise JavaBeans) y <i>servlets</i> . = Herramientas, librerías, etc.	+ Su gran ventaja es la inmensa cantidad de desarrolladores. = Herramientas, librerías, etc.
Plataforma cruzada	++ Mayor ventaja	- Puede ser negada si VB emite código byte.
Opciones de despliegue	= EXE's y código byte. = Componentes Beans y EJB. = <i>Servlets</i> de Web. + Applets.	= EXE's y código-p. = Componentes ActiveX. = <i>Servlets</i> de Web.
Velocidad	+ Ventaja para Java en el servidor. + Optimizadores de código byte, empiezan a usarse confiables aplicaciones compiladas.	+ Ventaja para VB en el cliente.

Tabla 3.2. Comparación de características de Java y VB. (++Muy Bueno, + Bueno, = Justo, - Débil)

4. ANÁLISIS

En este capítulo se presentan los resultados obtenidos de la etapa de análisis o JRP de la metodología de desarrollo rápido de aplicaciones (RAD).

4.1. Usuarios clave.

Puesto	Nombre
Supervisor de Cuentas por Pagar	Miguel Gavito
Jefe de Comunicaciones y Mantenimiento	Felipe López
Analista de Comunicaciones	Antonio Hernández
Gerente de Sistemas de Información	Macedonio Garza

Tabla 4.1. Usuarios clave involucrados en el sistema.

4.2. Expectativas de los usuarios.

Supervisor de Cuentas por Pagar.

- Consultar las cantidades, tiempos y costos de las llamadas realizadas por las unidades de negocio y departamentos de la empresa agrupadas por unidades de tiempo como día, semana, mes, año o un periodo de fechas y clasificadas por el tipo de servicio (servicio medido, larga distancia, larga distancia internacional y celular).
- Obtener reportes gráficos de las cantidades, tiempos y costos de las llamadas realizadas por las unidades de negocio y departamentos de la empresa agrupadas por periodos de tiempo y clasificadas por el tipo de servicio.

- Consultar una conciliación entre las llamadas realizadas por la empresa y las facturadas por el proveedor por el período de la factura.
- Mostrar puntos de alerta en los reportes en modo texto que indiquen que se ha hecho mal uso de la extensión.

Jefe de Comunicaciones y Mantenimiento.

- Activar y desactivar extensiones análogas y digitales.
- Actualizar y consultar los permisos de las extensiones.
- Actualizar y consultar la información general de las extensiones.
- Consultar en forma de listado la información general de las extensiones.
- Presentar a los socios un reporte del directorio telefónico de la empresa con la posibilidad de realizar búsquedas por el nombre del socio.
- Actualizar reglas por extensión de modo que sirvan de puntos de alerta para los reportes generados.

Analista de Comunicaciones.

- Activar y desactivar extensiones análogas y digitales.
- Actualizar y consultar los permisos de las extensiones.
- Actualizar y consultar la información general de las extensiones.
- Consultar en forma de listado la información general de las extensiones.
- Capturar las solicitudes de activación y modificación de extensiones.
- Consultar las solicitudes de activación y modificación de extensiones.
- Cancelar solicitudes de activación y modificación de extensiones.

Gerente de Sistemas de Información.

- Consultar las cantidades, tiempos y costos de las llamadas realizadas de la unidad de negocio, sus departamentos y sus socios agrupadas por unidades de tiempo como día, semana, mes, año o un período de fechas y clasificadas por el tipo de servicio.
- Obtener reportes gráficos de las cantidades, tiempos y costos de las realizadas de la unidad de negocio, sus departamentos y sus socios agrupadas por períodos de tiempo y clasificadas por el tipo de servicio.
- Consultar los teléfonos más marcados por los socios en un período de tiempo en forma de listado y en forma gráfica y clasificarlos por el tipo de servicio.
- Actualizar la información de los teléfonos que se involucran con la empresa.
- Consultar los teléfonos que son marcados por un socio y no se involucran con la empresa en forma de listado y en forma gráfica.
- Mostrar puntos de alerta en los reportes en modo texto que indiquen que se ha hecho mal uso de la extensión.

4.3. Objetivos y Alcances.

Los objetivos y alcances que son necesarios definir en la etapa del JRP son los mismos que se han definido en el segundo capítulo del presente documento.

4.4. Funciones del sistema.

A continuación se describen las funciones que realizará el sistema así como su prioridad y beneficios.

Función	Descripción
Actualizar la información de los usuarios del sistema y sus privilegios.	Permitir al administrador del sistema actualizar la información de los usuarios y otorgarles privilegios de acceso a las opciones del sistema.
Almacenar la información de las llamadas externas.	Recopilar todas las llamadas que son enviadas por el puerto serial de los conmutadores de cada una de las sucursales, clasificarlas por el tipo de servicio y almacenarlas en una base de datos. En caso de contingencia utilizar el almacenamiento local.
Enviar un correo electrónico al administrador.	Permitir que se envíe un correo electrónico cuando ocurra un error en la recopilación de los datos de las llamadas o el almacenamiento de estas.
Actualizar los costos de las llamadas mediante conciliación de información.	Permitir al departamento de cuentas por pagar hacer una conciliación entre las llamadas recopiladas y el detalle de llamadas del archivo de facturación de Telmex y actualizar los costos de las llamadas almacenadas.
Consultar la forma en que se conciliaron las llamadas.	Permitir al departamento de cuentas por cobrar observar como se relacionaron las llamadas recopiladas con el detalle de llamadas de la facturación de Telmex.
Actualizar la relación socio y extensión	Permitir al departamento de comunicaciones asociar las extensiones con los socios.
Actualizar la información de los teléfonos involucrados con la empresa.	Permitir a los usuarios del sistema ingresar la información de teléfonos que están directamente involucrados con la empresa, para que estos se tomen en cuenta en los reportes como gastos reales de la empresa.
Actualizar la relación entre las extensiones y los servicios de telefonía.	Permitir al departamento de comunicaciones relacionar las extensiones con los servicios de telefonía que pueden ofrecer para poder identificar el mal uso de estas.
Consultar las llamadas realizadas por las unidades de negocio.	Permitir a los directores y al departamento de cuentas por pagar observar reportes de las cantidades, tiempos y costos de las llamadas realizadas por las unidades de negocio, indicando puntos de alerta y clasificando las llamadas por tipo de servicio.
Graficar la cantidad de llamadas realizadas por las unidades de negocio.	Permitir a los directores y al departamento de cuentas por pagar observar reportes gráficos de la cantidad de llamadas realizadas por las unidades de negocio.
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los departamentos.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes de las cantidades, tiempos y costos de las llamadas realizadas por los departamentos, indicando puntos de alerta y clasificando las llamadas por tipo de servicio.
Graficar la cantidad de llamadas realizadas por los departamentos.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes gráficos de la cantidad de llamadas realizadas por los departamentos.
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los socios.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes de las cantidades, tiempos y costos de las llamadas realizadas por los socios, indicando puntos de alerta y clasificando las llamadas por tipo de servicio.
Graficar la cantidad de llamadas realizadas por los socios.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes gráficos de la cantidad de llamadas realizadas por los socios.

Función	Descripción
Consultar los teléfonos más marcados por los socios, indicando los teléfonos que no se involucran con la empresa.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar un reporte de los números que son marcados por los socios y indicando cuales no se involucran con la empresa.
Graficar las cantidades y tiempos de las llamadas a los teléfonos más marcados por los socios.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes gráficos de las cantidades y tiempos de las llamadas de los socios a teléfonos que no se involucran directamente con la empresa.
Graficar las cantidades y tiempos de los teléfonos marcados por los socios y que no se involucran con la empresa.	Permitir a los directores, gerentes y al departamento de cuentas por pagar observar reportes gráficos de las cantidades y tiempos de las llamadas a los teléfonos más marcados por los socios.
Consultar el directorio de extensiones de la empresa.	Permitir a los usuarios del sistema consultar el directorio de extensiones de la empresa y permitir realizar búsquedas por socio.
Actualizar la información general de las extensiones en el conmutador.	Permitir al departamento de comunicaciones actualizar la información de los puertos, número de extensión y usuario en el conmutador.
Consultar la información general de las extensiones del conmutador.	Permitir al departamento de comunicaciones observar un reporte con la información general de las extensiones que tiene el conmutador ya sea individual o en forma de listado.
Actualizar los permisos de las extensiones en el conmutador.	Permitir al departamento de comunicaciones activar o hacer cambios en los permisos que tiene cada extensión para poder acceder a los servicios.
Actualizar solicitudes de activación y cambios de extensiones.	Permitir al departamento de Help Desk la captura de las solicitudes de activación y cambios de extensiones, para que sean procesadas por el departamento de comunicaciones.
Consultar las solicitudes que requieran de activación o cambios de extensiones.	Permitir al departamento de comunicaciones consultar las solicitudes de activación y cambios de extensiones.
Consultar la información para poder satisfacer una solicitud de activación o cambios de extensiones.	Permitir al departamento de comunicaciones observar la información del conmutador que sea necesaria para satisfacer una solicitud de activación o cambios de extensiones.
Cancelar solicitudes de activación o cambios de extensiones.	Permitir al departamento de comunicaciones cancelar las solicitudes de activación y cambios de extensiones.
Enviar un correo electrónico al socio.	Permitir que se envíe un correo al socio que realizó la solicitud de activación o cambio en extensión cuando esta haya cambiado de estatus, de modo que tenga conocimiento del avance de su solicitud.
Activar extensiones análogas.	Permitir al departamento de comunicaciones activar extensiones análogas ya sea por el proceso de solicitud (cambiar el estatus de la solicitud a terminada) o cuando se requiera ante otras situaciones.
Activar extensiones digitales.	Permitir al departamento de comunicaciones activar extensiones digitales ya sea por el proceso de solicitud (cambiar el estatus de la solicitud a terminada) o cuando se requiera ante otras situaciones.
Desactivar extensiones digitales.	Permitir al departamento de comunicaciones desactivar extensiones digitales manualmente.
Desactivar extensiones análogas.	Permitir al departamento de comunicaciones desactivar extensiones análogas manualmente.
Depurar los archivos de almacenamiento local.	Depurar los archivos de almacenamiento local cuando la información almacenada no sea necesaria.
Depurar la base de datos.	Depurar la base de datos en un período de tiempo determinado.
Actualizar la información de conmutadores, tipos de extensión, tipos de llamada, claves de larga distancia, tipos de solicitudes, áreas genéricas y teléfonos permitidos.	Actualizar la información de los catálogos de la base de datos del sistema.

Tabla 4.2. Funciones del sistema.

Función	Prioridad
Actualizar la información de conmutadores, tipos de extensión, tipos de llamada, claves de larga distancia, tipos de solicitudes, áreas genéricas y teléfonos permitidos.	1
Actualizar la información de los usuarios del sistema y sus privilegios.	1
Actualizar la relación socio y extensión	1
Almacenar la información de las llamadas externas.	1
Actualizar la información de los teléfonos involucrados con la empresa.	2
Actualizar la relación entre las extensiones y los servicios de telefonía.	2
Actualizar los costos de las llamadas mediante conciliación de información.	2
Consultar la forma en que se conciliaron las llamadas.	2
Actualizar solicitudes de activación y cambios de extensiones.	3
Cancelar solicitudes de activación o cambios de extensiones.	3
Consultar la información para poder satisfacer una solicitud de activación o cambios de extensiones.	3
Consultar las solicitudes que requieran de activación o cambios de extensiones.	3
Activar extensiones análogas.	4
Activar extensiones digitales.	4
Actualizar la información general de las extensiones en el conmutador.	4
Actualizar los permisos de las extensiones en el conmutador.	4
Consultar el directorio de extensiones de la empresa.	4
Consultar la información general de las extensiones del conmutador.	4
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los departamentos.	4
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los socios.	4
Consultar las llamadas realizadas por las unidades de negocio.	4
Consultar los teléfonos más marcados por los socios, indicando los teléfonos que no se involucran con la empresa.	4
Desactivar extensiones análogas.	4
Desactivar extensiones digitales.	4
Depurar la base de datos.	5
Depurar los archivos de almacenamiento local.	5
Enviar un correo electrónico al administrador.	5
Enviar un correo electrónico al socio.	5
Graficar la cantidad de llamadas realizadas por las unidades de negocio.	5
Graficar la cantidad de llamadas realizadas por los departamentos.	5
Graficar la cantidad de llamadas realizadas por los socios.	5
Graficar las cantidades y tiempos de las llamadas a los teléfonos más marcados por los socios.	5
Graficar las cantidades y tiempos de los teléfonos marcados por los socios y que no se involucran con la empresa.	5

Tabla 4.3. Prioridad de las funciones del sistema.

Función	Beneficio u oportunidad
Activar extensiones análogas.	Contar con una interfase gráfica para poder activar extensiones análogas sin necesidad de ejecutar comandos en una terminal.
Activar extensiones digitales.	Contar con una interfase gráfica para poder activar extensiones digitales sin necesidad de ejecutar comandos en una terminal.
Actualizar la información general de las extensiones en el conmutador.	Contar con una interfase gráfica para actualizar la información general de las extensiones sin necesidad de ejecutar comandos en una terminal. De este modo es posible configurar rápidamente las extensiones.
Almacenar la información de las llamadas externas.	Contar con la información centralizada de las llamadas que se realizan fuera de la red de voz.
Consultar la información general de las extensiones del conmutador.	Contar con una interfase gráfica para consultar la información general de las extensiones sin necesidad de ejecutar comandos en una terminal. De este modo es posible conocer rápidamente la configuración de las extensiones.
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los departamentos.	Conocer los gastos generados por los departamentos, al igual que el tiempo invertido en el uso del teléfono permitiendo hacer presupuestos más exactos y verificar la productividad del departamento.
Consultar las cantidades, tiempos y costos de las llamadas realizadas por los socios.	Conocer los gastos generados por los socios, al igual que el tiempo invertido en el uso del teléfono permitiendo hacer presupuestos más exactos y verificar la productividad del socio.
Consultar las llamadas realizadas por las unidades de negocio.	Conocer los gastos generados por las unidades de negocio, al igual que el tiempo invertido en el uso del teléfono permitiendo hacer presupuestos más exactos y verificar la productividad de la unidad de negocio.
Desactivar extensiones análogas.	Contar con una interfase gráfica para poder desactivar extensiones análogas sin necesidad de ejecutar comandos en una terminal.
Desactivar extensiones digitales.	Contar con una interfase gráfica para poder desactivar extensiones digitales sin necesidad de ejecutar comandos en una terminal.
Actualizar la relación socio y extensión	Mantener actualizada la información de pertenencia de las extensiones mediante una interfase gráfica, de modo que sea posible relacionar la información de las llamadas con los socios, departamentos y unidades de negocio.
Actualizar los costos de las llamadas mediante conciliación de información.	Conocer los costos reales de las llamadas registradas conciliándolas con el detalle de llamadas de la factura telefónica, permitiendo también identificar errores en la facturación.
Actualizar los permisos de las extensiones en el conmutador.	Contar con una interfase gráfica para poder actualizar los permisos de las extensiones sin necesidad de ejecutar comandos en una terminal.
Actualizar solicitudes de activación o cambios de extensiones.	Conocer los datos necesarios para atender la necesidad del socio de manera que pueda ser atendida su solicitud más rápidamente.
Cancelar solicitudes de activación o cambios de extensiones.	En caso de que no se autoricen las solicitudes será posible cancelar la solicitud.
Consultar el directorio de extensiones de la empresa.	Los socios podrán consultar las extensiones de otros socios mediante la Intranet.
Consultar la información para poder satisfacer una solicitud de activación o cambios de extensiones.	Contar con una interfase gráfica en la que se muestre la información del conmutador que puede satisfacer esa solicitud y permita al usuario seleccionarla y terminar la solicitud con los datos seleccionados, lo cual hará más sencilla la atención de las solicitudes.
Consultar las solicitudes que requieran de activación o cambios de extensiones.	Poder dar seguimiento a las solicitudes de los socios mediante la consulta de estas en una interfase gráfica.
Consultar los teléfonos más marcados por los socios, indicando los teléfonos que no se involucran con la empresa.	Conocer el uso que esta dándole el socio a la extensión telefónica y mediante estos será posible evaluar la productividad de este.
Graficar la cantidad de llamadas realizadas por las unidades de negocio.	Poder observar gráficamente y de manera global la forma en que se está utilizando la red de voz.

Función	Beneficio u oportunidad
Graficar la cantidad de llamadas realizadas por los departamentos.	Poder observar gráficamente y de manera global y detallada la forma en que se está utilizando la red de voz.
Graficar la cantidad de llamadas realizadas por los socios.	Poder observar gráficamente y de manera global y detallada la forma en que se está utilizando la red de voz.
Graficar las cantidades y tiempos de los teléfonos marcados por los socios y que no se involucran con la empresa.	Poder observar gráficamente y de manera detallada la forma en que se está utilizando la red de voz y el tiempo que invierte el socio en hablar a cierto número telefónico.
Actualizar la información de los teléfonos involucrados con la empresa.	Permitir contar con la información de los teléfonos a los que se pueden justificar las llamadas telefónicas y con esta información poder clasificar las llamadas de los reportes.
Actualizar la información de los usuarios del sistema y sus privilegios.	Permitir dar acceso a los usuarios a las opciones que le corresponden.
Actualizar la relación entre las extensiones y los servicios de telefonía.	Permitir asociar las extensiones con los servicios de telefonía que pueden realizar, para poder identificar en los reportes las extensiones a las que se les está dando mal uso.
Consultar la forma en que se conciliaron las llamadas.	Poder verificar como coincidieron las llamadas, que llamadas no se les asignó el costo y que llamadas de la facturación no se registraron.
Graficar las cantidades y tiempos de las llamadas a los teléfonos más marcados por los socios.	Poder observar gráficamente y de manera detallada la forma en que se está utilizando la red de voz y el tiempo que invierte el socio en hablar a cierto número telefónico.
Depurar la base de datos.	Permitir eliminar la información que no es necesaria para mantener un cierto volumen de información.
Depurar los archivos de almacenamiento local.	Eliminar la información que no es necesaria para evitar que se eleve el volumen de información.
Enviar un correo electrónico al administrador.	Facilitar el monitoreo del funcionamiento de la aplicación.
Enviar un correo electrónico al socio.	Permitir al socio monitorear el avance del cumplimiento de solicitud.
Actualizar la información de conmutadores, tipos de extensión, tipos de llamada, claves de larga distancia, tipos de solicitudes, áreas genéricas y teléfonos permitidos.	Permitir a los administradores hacer cambios en la configuración de los catálogos del sistema

Tabla 4.4. Beneficios de las funciones del sistema.

4.5. Procesos del Sistema.

4.5.1. Descomposición de Procesos.

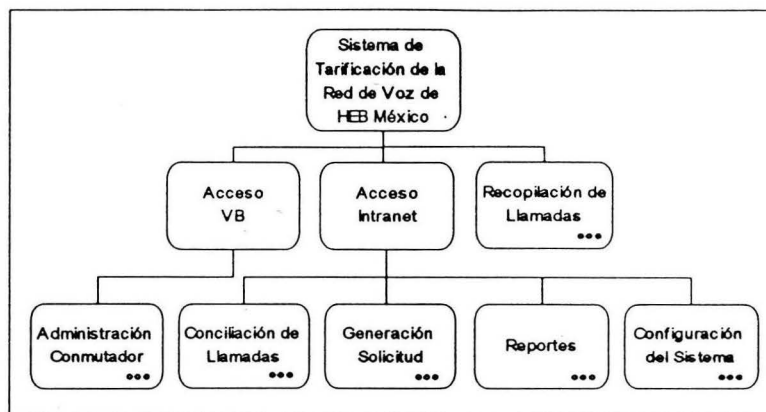


Figura 4.1. Descomposición de procesos (Sistema).

El sistema se compondrá de dos procesos de acceso, uno para los procesos de Intranet y otro para los procesos en Visual Basic, los cuales controlarán el ingreso a las opciones y a la información. Además existirán seis procesos principales (Figura 4.1), administración del conmutador, conciliación de llamadas, generación de solicitudes, recopilación de llamadas, generación de reportes y la configuración del sistema, cuya función detallada será explicada en los siguientes párrafos.

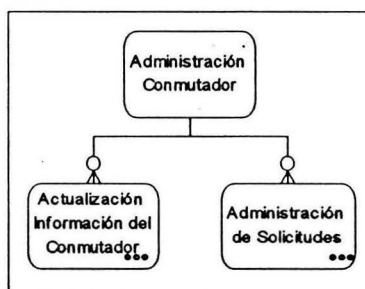


Figura 4.2. Descomposición de procesos (Administración Conmutador).

El proceso de administración del conmutador está compuesto por el proceso de actualización de la información del conmutador y el de administración de solicitudes (Figura 4.2). A través del proceso de actualización de la información del conmutador se realizarán movimientos propios de la configuración del conmutador como activaciones y permisos. Por medio del proceso de administración de solicitudes se podrán satisfacer las solicitudes de activación y cambios en las extensiones que generen los socios.

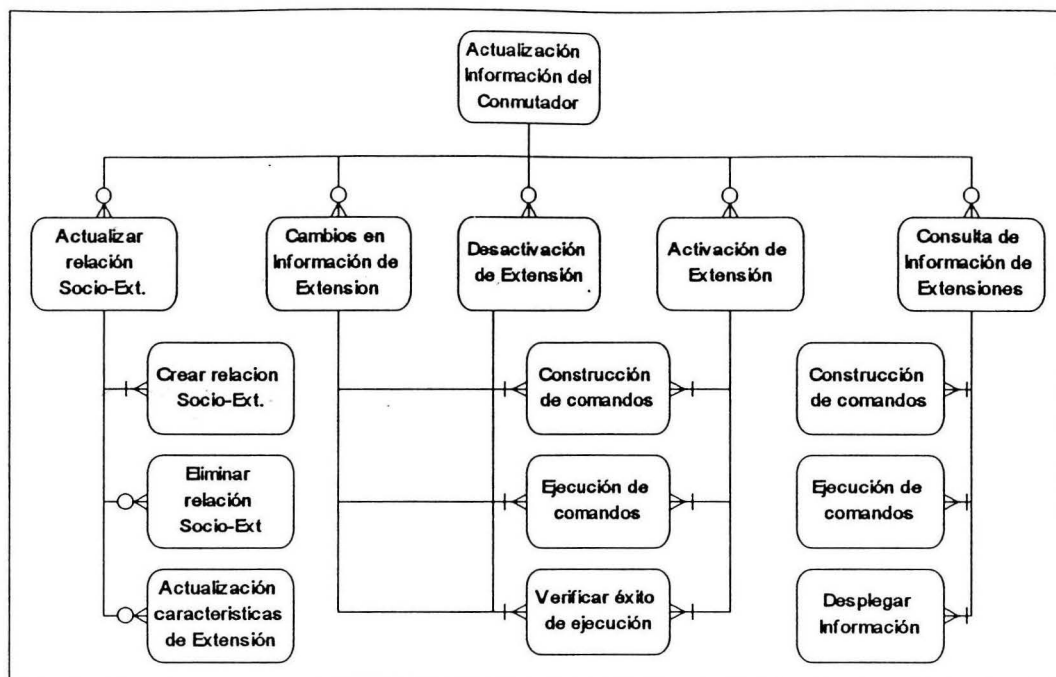


Figura 4.3. Descomposición de procesos (Actualización Información del Conmutador).

El proceso de actualización de información del conmutador se compone de cinco procesos, actualización de la relación socio-extensión, cambios en la información de la extensión, activación de extensión, desactivación de extensión y la consulta de la información de la extensión (Figura 4.3).

Mediante el proceso de actualización de la relación socio-extensión será posible crear y eliminar la relación socio-extensión para poder establecer un propietario de la extensión que se mostrará después en los reportes del sistema. Además se cuenta con el subproceso para modificar las características de la extensión en el cual se consultará y modificará información técnica de las extensiones sólo para referencia.

Los procesos de cambios en la información de la extensión, activación de extensión y desactivación de extensión consisten en funciones similares por lo que se componen de los mismos subprocesos. Los subprocesos por los que están compuestos son el de construcción de comandos, ejecución de comandos y el de verificación de ejecución. Mediante estos subprocesos se podrán realizar las modificaciones necesarias a la configuración del conmutador para su administración.

El proceso de consulta de información de extensiones realizará funciones similares a los procesos anteriores. Sin embargo, este proceso necesita del despliegue de la información obtenida por la ejecución de los comandos, por lo que se compone de los subprocesos de construcción de comandos, ejecución de comandos y despliegue de información.

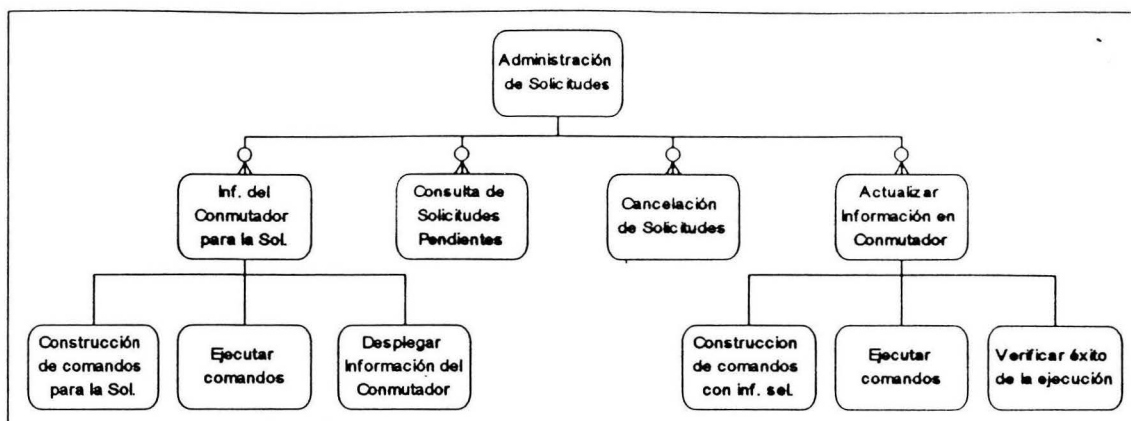


Figura 4.4. Descomposición de procesos (Administración de Solicitudes).

El proceso de administración de solicitudes se divide en cuatro procesos mediante los cuales se manejará la información de las solicitudes de activación y cambios en las extensiones. Estos procesos son la consulta de solicitudes pendientes, la obtención de información del conmutador para la satisfacción de las solicitudes, la actualización de la información del conmutador y la cancelación de solicitudes (Figura 4.4).

Mediante el proceso de consulta de solicitudes pendientes se podrán revisar las solicitudes de los socios que aun no han sido atendidas. El proceso de obtención de información del conmutador construirá los comandos mediante los cuales se pueda obtener la información del conmutador que pueda satisfacer la solicitud previamente seleccionada, para posteriormente ejecutarlos en el conmutador y desplegar esta información. A través del proceso de actualización de información en el conmutador y sus subprocesos (construcción de comandos, ejecución de comandos y verificación de ejecución) se realizarán los cambios en el conmutador con la información previamente seleccionada y se cumplirá con la solicitud de socio. Mediante el proceso de cancelación de solicitudes se rechazarán aquellas solicitudes que no han sido autorizadas.

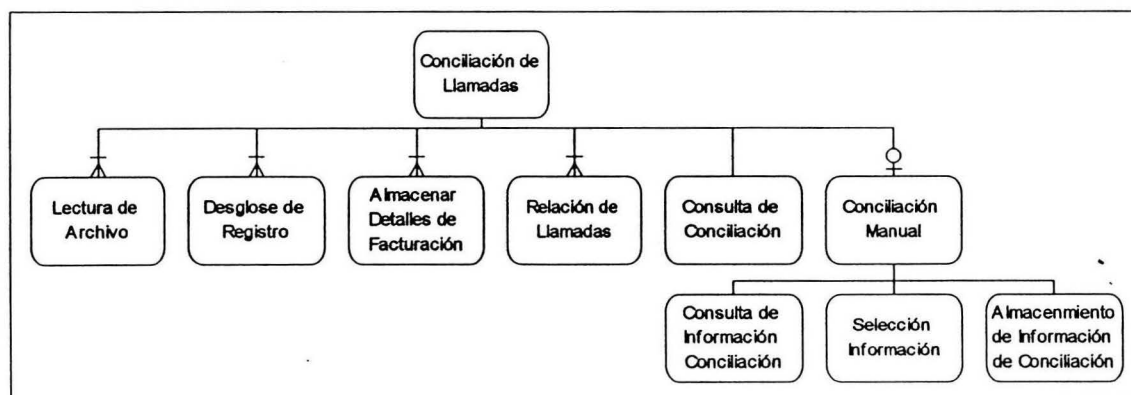


Figura 4.5. Descomposición de procesos (Conciliación de Llamadas).

Mediante el proceso de conciliación de llamadas se hará coincidir las llamadas registradas y las llamadas detalladas en la factura del proveedor de servicios telefónicos. Este proceso se compone de los procesos de lectura de archivo, desglose de registro, almacenamiento de detalles de facturación, relación de llamadas, consulta de conciliación y conciliación manual (Figura 4.5).

El proceso de lectura de archivo extraerá línea por línea los detalles de las llamadas del archivo de facturación. El desglose de registro consistirá en dividir en campos la línea leída del archivo. Los campos del archivo que sean necesarios para la conciliación serán almacenados en la base de datos por el proceso de almacenamiento de detalles de facturación. Los detalles de la facturación almacenados se tratarán de conciliar con las llamadas recopiladas mediante el proceso de relación de llamadas, obteniendo así los costos y los tiempos asignados por el proveedor a las llamadas recopiladas. El proceso de consulta de conciliación mostrará al usuario como fueron relacionadas las llamadas y a cuales no se les pudo encontrar relación. Si el usuario desea relacionar las llamadas restantes o hacer cambios en las relaciones se utilizará el proceso de conciliación manual. Este proceso cuenta con tres subprocesos con los que se consultará la información de conciliación, se permitirá al usuario seleccionar la información de las llamadas recopiladas y las de la facturación y relacionarlas y almacenar los cambios en la conciliación de las llamadas.

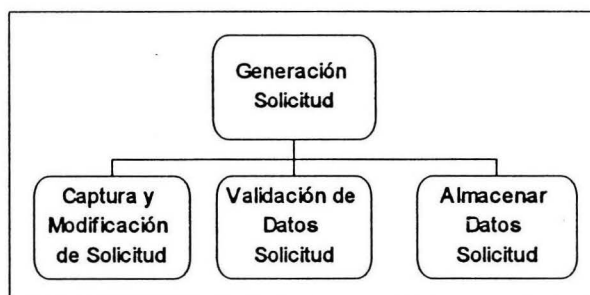


Figura 4.6. Descomposición de procesos (Generación Solicitud).

El proceso de generación de solicitudes permitirá a los usuarios crear solicitudes de activación y cambios en las extensiones. Este proceso se compone de tres subprocesos, la captura y modificación, la validación de los datos y el almacenamiento de los datos de la solicitud. Mediante el subproceso de captura y modificación podrán ser ingresados los datos de la solicitud para que sea creada o modificada. Los datos deben estar completos, lo que verificará el subproceso de validación de datos. Cuando los datos de la nueva solicitud o los cambios realizados sean válidos se almacenarán en la base de datos para que pueda ser atendida la solicitud.

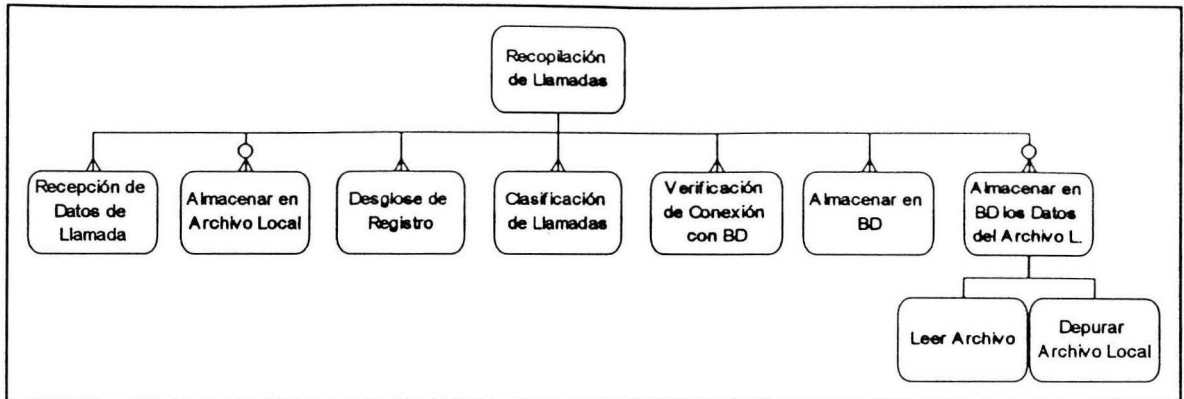


Figura 4.7. Descomposición de procesos (Recopilación de Llamadas).

El proceso de recopilación de llamadas será quien esté recibiendo las llamadas del conmutador y almacenándolas. Este proceso se compone de siete subprocesos (Figura 4.7).

El subproceso de recepción de datos de llamada se encargará de recolectar los registros que son enviados por el conmutador al puerto serial de la computadora donde se ejecuta. El segundo subproceso, el de almacenamiento en archivo local, almacenará todos los registros recopilados en un archivo local mientras exista una contingencia. El subproceso de desglose de registro se encargará de dividir en campos el registro enviado por el conmutador. El cuarto subproceso, el de clasificación de llamadas, analizará la llamada recibida para poder ubicarla en el tipo de servicio que le corresponda. El subproceso de verificación de conexión con la base de datos se encargará de revisar que la llamada se pueda almacenar en la base de datos. Mediante el subproceso de almacenamiento en la base de datos se grabará la llamada recibida en la base de datos y el subproceso de almacenar en la base de datos los datos del archivo local se encargará de procesar los registros que hayan sido almacenados en el archivo local.

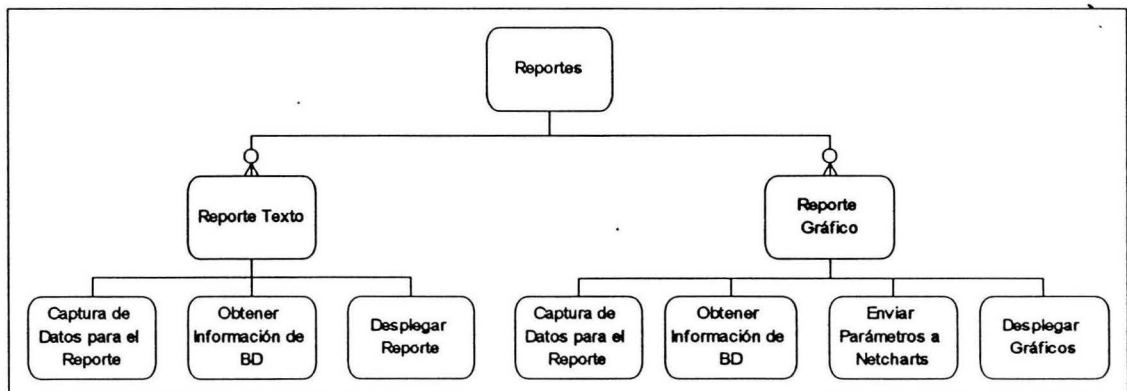


Figura 4.8. Descomposición de procesos (Reportes).

El proceso de reportes estará compuesto por dos procesos que permitirán generar reportes de tipo texto y reportes gráficos. El proceso de reporte de texto se subdividirá en tres subprocesos. El primero, captura de datos para consultar la información de la base de datos para posteriormente generar el reporte. El subproceso de obtención de información de la base de datos realizará la consulta en la base de datos generando el reporte solicitado.

El proceso de reporte gráfico será el encargado de generar gráficas para el análisis de información y este se compondrá de cuatro subprocesos. El primero, captura de datos para el reporte, recibirá la información con la que se quiere generar el reporte gráfico. El subproceso de obtención de información de la base de datos realizará la consulta en la base de datos de para poder generar la gráfica mediante el subproceso de envío de parámetros a NetCharts. El subproceso de despliegue de gráficos mostrará al usuario el reporte gráfico solicitado.

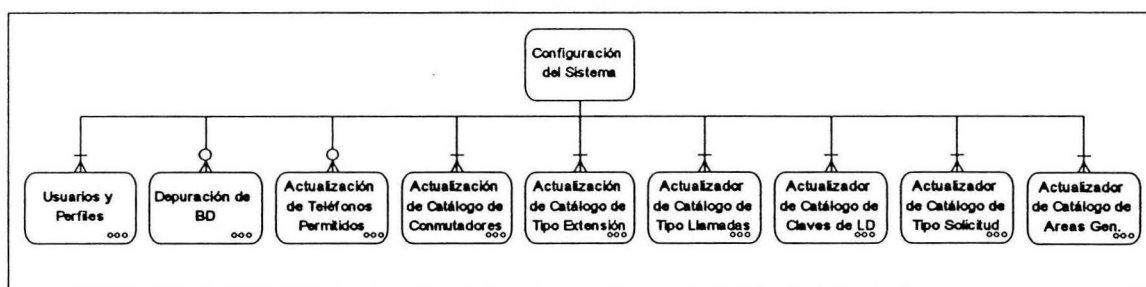


Figura 4.9. Descomposición de procesos (Configuración del Sistema).

El proceso de configuración estará formado por nueve procesos que son usuarios y perfiles, actualizador de teléfonos permitidos (involucrados con la empresa), la depuración de la base de datos, la actualización del catálogo de conmutadores, tipos de extensión, tipos de llamadas, claves de larga distancia, tipos de solicitud y de áreas genéricas (Figura 4.9).

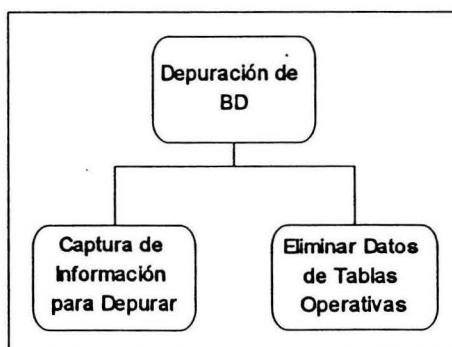


Figura 4.10. Descomposición de procesos (Depuración BD).

El proceso de depuración de la base de datos se dividirá en dos subprocesos la captura de la información para depurar, el cual permitirá ingresar las fechas para

eliminar posteriormente la información por el subproceso de eliminación de datos de las tablas operativas.

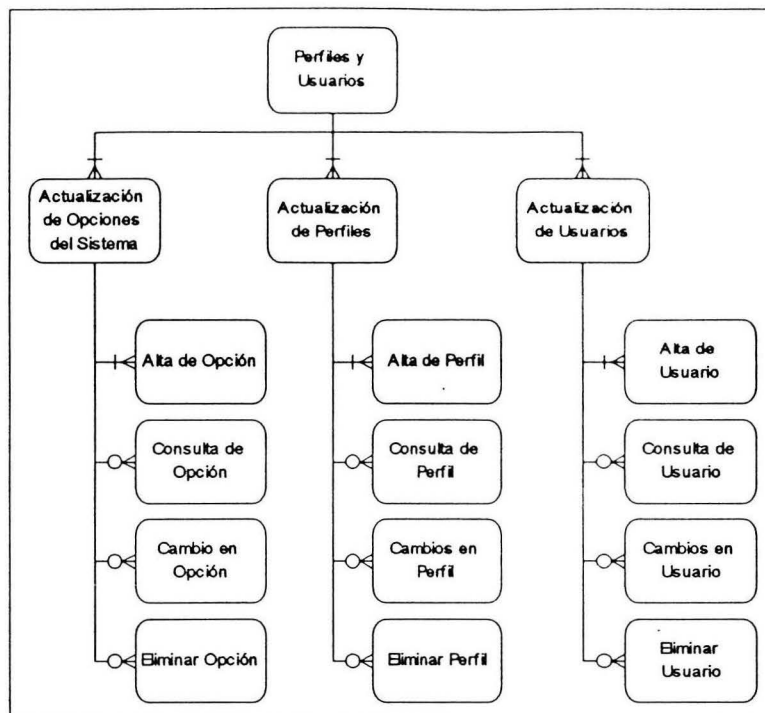


Figura 4.11. Descomposición de procesos (Perfiles y Usuarios).

El proceso de perfiles y usuarios se compondrá de tres procesos, la actualización de opciones del sistema, la actualización de perfiles y la actualización de usuarios (Figura 4.11). Mediante el proceso de actualización de opciones del sistema se podrán crear, modificar, consultar y eliminar opciones que representarán los módulos del sistema y los tipos de accesos que puede tener un perfil. A través del proceso de actualización de perfiles será posible agrupar opciones del sistema en un mismo perfil, este proceso cuenta con subprocesos para crear, consultar, modificar y eliminar los perfiles. El proceso de actualización de usuarios permitirá crear, consultar, modificar y eliminar usuarios con una clave de acceso y una contraseña y asociarlos a un determinado perfil para controlar los accesos a las opciones y a la información del sistema.

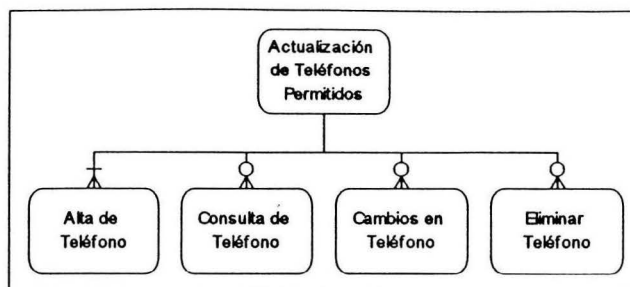


Figura 4.12. Descomposición de procesos (Actualizador de Teléfonos Permitidos).

El proceso de actualización de teléfonos permitidos podrá a través de sus subprocesos crear, consultar, modificar y eliminar teléfonos que están involucrados con la empresa de modo que en los reportes de las llamadas recopiladas sea posible identificar las llamadas realizadas que tienen relación con la empresa y cuáles no están involucradas.

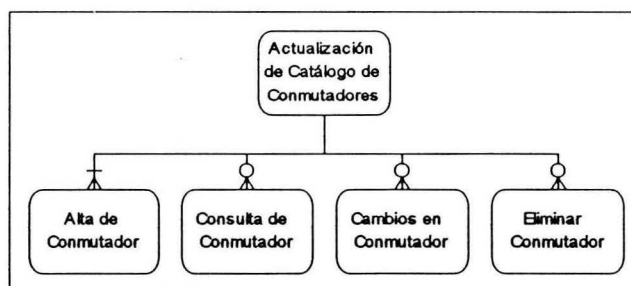


Figura 4.13. Descomposición de procesos (Actualización de Catálogo de Conmutadores).

Mediante el proceso de actualización de catálogo de conmutadores será posible crear, consultar, modificar y eliminar los conmutadores con los que cuenta la empresa para después incluirlos en los reportes.

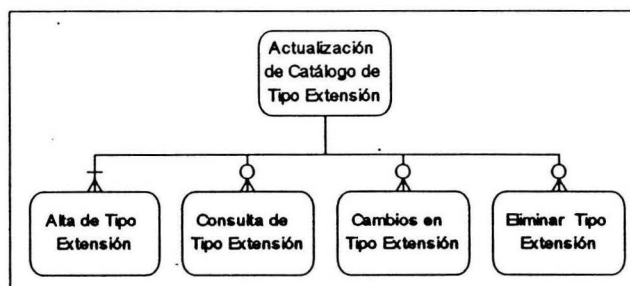


Figura 4.14. Descomposición de procesos (Actualización de Catálogo de Tipo Extensión).

El proceso de actualización de catálogo de tipo de extensión podrá a través de sus subprocesos crear, consultar, modificar y eliminar tipos de extensión en caso de que se realicen cambios en la forma de operación de las extensiones

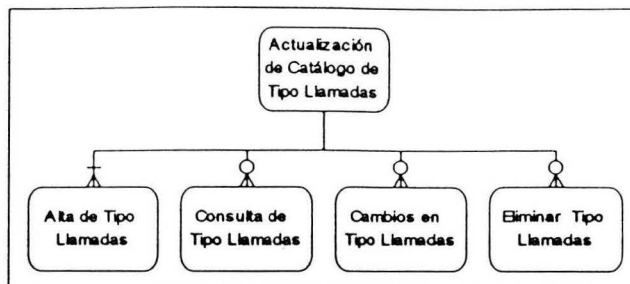


Figura 4.15. Descomposición de procesos (Actualización de Catálogo de Tipo Llamadas).

Mediante el proceso de actualización del catálogo de tipo de llamadas se podrá a través de sus subprocesos crear, consultar, modificar y eliminar tipos de llamada cuando los servicios de telefonía sufran algún cambio.

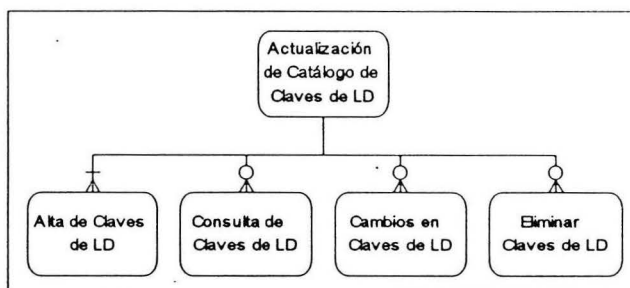


Figura 4.16. Descomposición de procesos (Actualización de Catálogo de Tipo Llamadas).

El proceso de actualización del catálogo de claves de larga distancia permitirá crear, consultar, modificar y eliminar claves de larga distancia para cualquier tipo de servicio de telefonía cuando las claves de larga distancia sufran algún cambio.

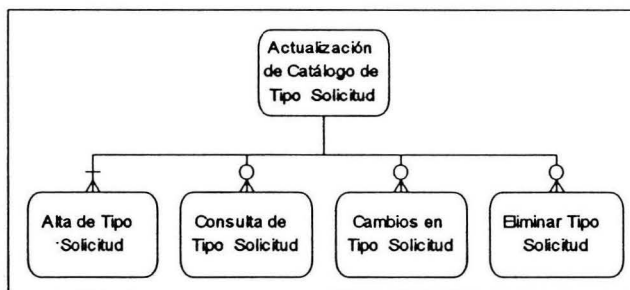


Figura 4.17. Descomposición de procesos (Actualización de Catálogo de Tipo Solicitud).

Mediante el proceso de actualización del catálogo de tipos de solicitud será posible crear, consultar, modificar y eliminar tipos de solicitudes de modo que si sufre cambios la solicitud de activación y cambios en extensión puedan ajustarse.

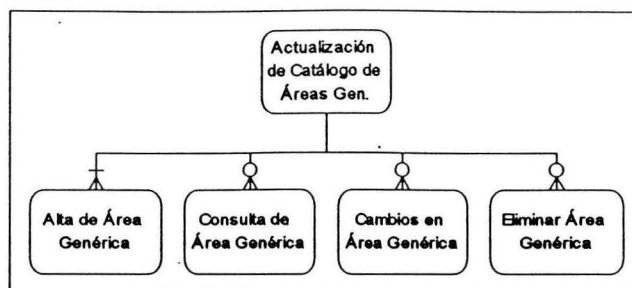


Figura 4.18. Descomposición de procesos (Actualización de Catálogo Áreas Genéricas).

A través del proceso de actualización del catálogo de áreas genéricas se podrán crear, consultar, modificar y eliminar áreas genéricas para posteriormente poder relacionar estas con la extensión que les corresponda.

4.5.2. Dependencia de procesos.

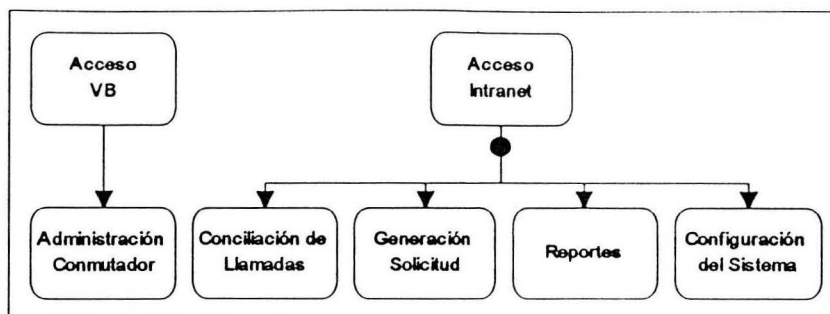


Figura 4.19. Dependencia de procesos (Acceso).

Los procesos de acceso permitirán o no la ejecución de los procesos de administración del conmutador, conciliación de llamadas, generación de solicitud, reportes y configuración del sistema, capturando la clave de acceso y la contraseña del usuario para validar los permisos de ejecución dependiendo de la configuración previa del perfil del usuario.

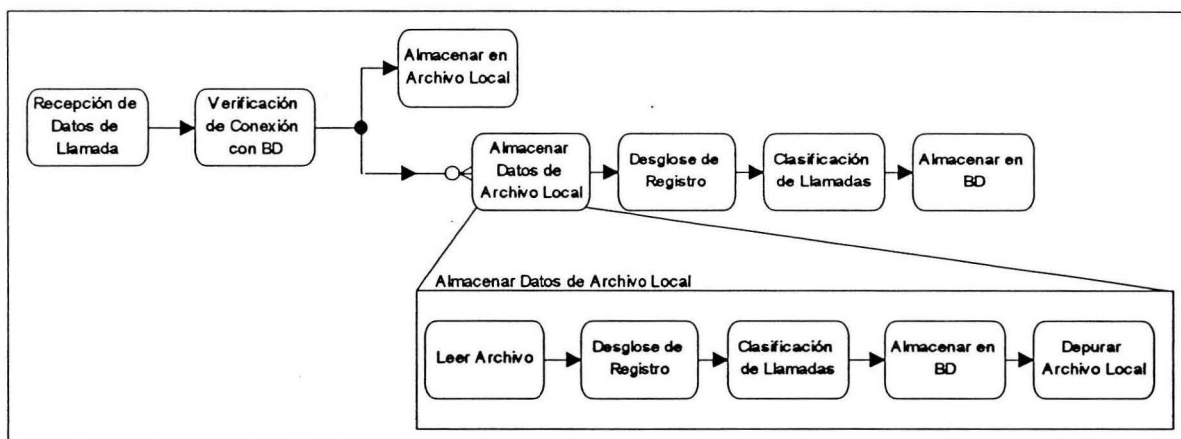


Figura 4.20. Dependencia de procesos (Recopilación de Llamadas).

El proceso de recopilación de llamadas (Figura 4.19) consistirá en recibir los datos de la llamada enviados por el conmutador, antes de procesarlos es necesario verificar si existe conexión con la base de datos, ya que de no ser así los datos serán almacenados en un archivo local mientras se reestablece la conexión. En caso de que exista conexión con la base de datos será necesario verificar si existen registros en el archivo local que no han sido almacenados. En este caso se leerá registro por registro el archivo, cada registro se desglosará en campos, se clasificará el tipo de servicio de la llamada y se almacenará en la base de datos, finalmente se depurará el archivo local. Al terminar de procesar el archivo local se desglosará el registro enviado por el conmutador,

posteriormente clasificará la llamada por el tipo de servicio y se almacenará en la base de datos.

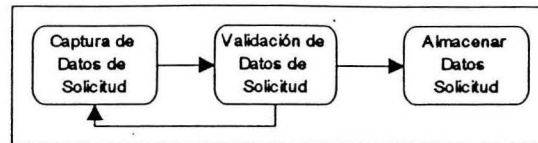


Figura 4.21. Dependencia de procesos (Generación Solicitud).

El proceso de generación de solicitud consistirá en la captura de los datos de la solicitud para después validar que se hayan capturado los datos necesarios para el tipo de solicitud y que estén correctos. Si los datos capturados pasaron la validación se almacenarán en la base de datos para que la solicitud pueda ser procesada (Figura 4.20).

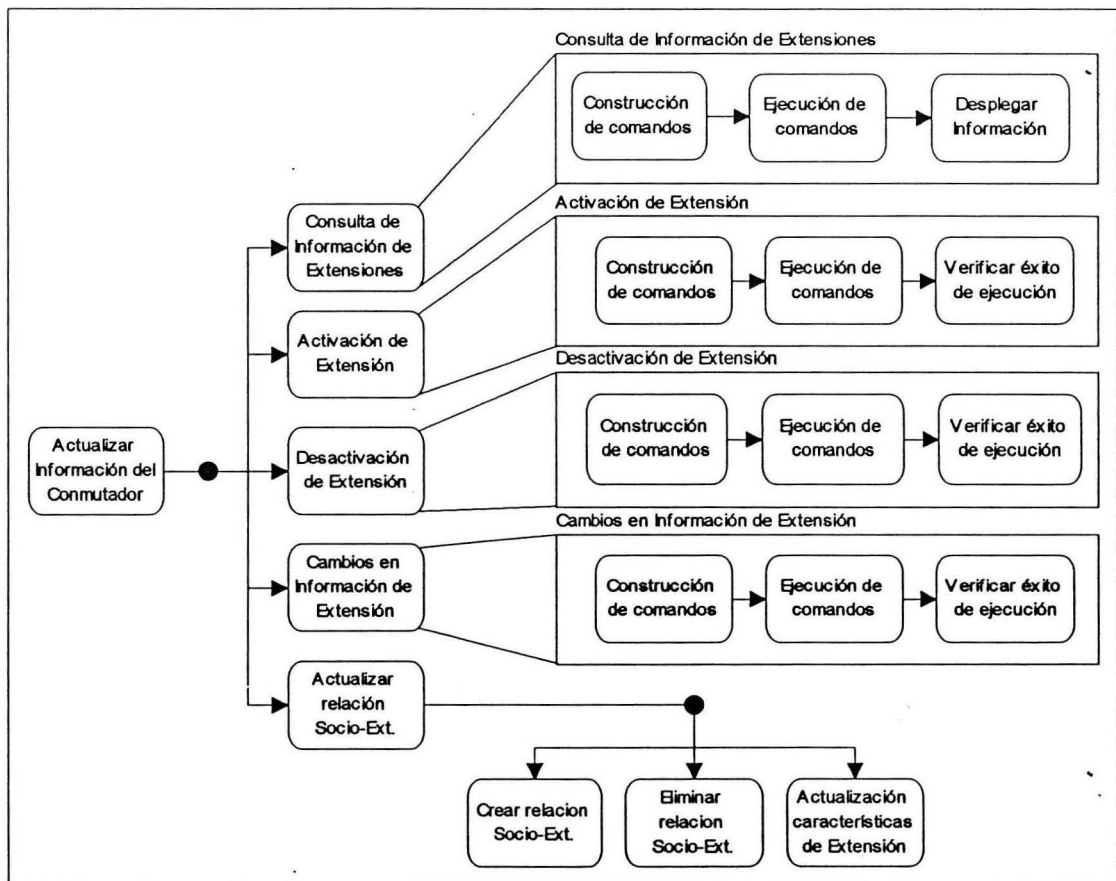


Figura 4.22. Dependencia de procesos (Actualizar Información del Conmutador).

El proceso de actualización de información del conmutador será la ejecución de cualquiera de sus procesos, consulta de información de extensiones, activación de extensiones, desactivación de extensiones, cambios en información de extensión y actualización de la relación socio-extensión (Figura 4.21).

El proceso de consulta de información de extensiones consultará en el conmutador información general sobre las extensiones del conmutador. Esto lo logrará construyendo y ejecutando los comandos adecuados para la consulta y desplegando la información al usuario.

Mediante los procesos de activación, desactivación y cambios en la información de la extensión se podrán habilitar y deshabilitar extensiones además de realizar modificaciones en la información general de estas. Para que esto se pueda realizar será necesario construir, ejecutar y verificar la ejecución de comandos del lenguaje del conmutador.

El proceso de actualización de la relación socio-extensión servirá para otorgarle propiedad a la extensión y se pueda identificar quien es el dueño de ésta. Será posible crear, eliminar la relación. Además también se podrán actualizar las características de las extensiones almacenadas en la base de datos.

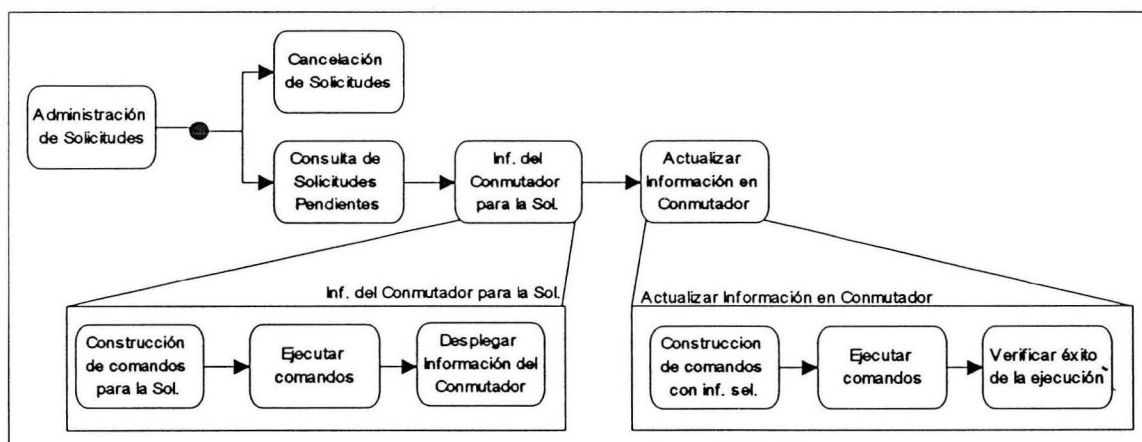


Figura 4.23. Dependencia de procesos (Administración de Solicitudes).

El proceso de administración de solicitudes (Figura 4.22) consistirá en la ejecución de la cancelación de solicitudes o la consulta de solicitudes pendientes. El proceso de consulta de solicitudes pendientes verificará en la base de datos qué solicitudes de activación y cambios en las extensiones no han sido procesadas. Posteriormente, se extraerá la información del conmutador que pueda satisfacer la solicitud que haya sido seleccionada. Esta información se obtendrá mediante la construcción y ejecución de comandos y el despliegue de la información enviada por el conmutador. Con esta información será posible satisfacer la solicitud mediante el proceso de actualización de información en el conmutador. Proceso que consistirá en la construcción de comandos con la información seleccionada previamente, su ejecución y la verificación de su ejecución.

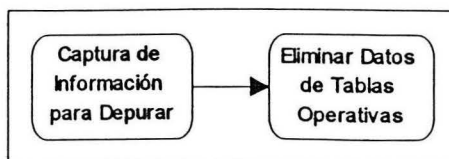


Figura 4.24. Dependencia de procesos (Depuración BD).

El proceso de depuración de la base de datos (Figura 4.23) implicará la captura de la información para realizar la depuración. Esta información consistirá en las fechas a partir de las cuales se desea realizar la depuración de las tablas operativas. Después de la captura se realizará la eliminación de los datos de las tablas operativas.

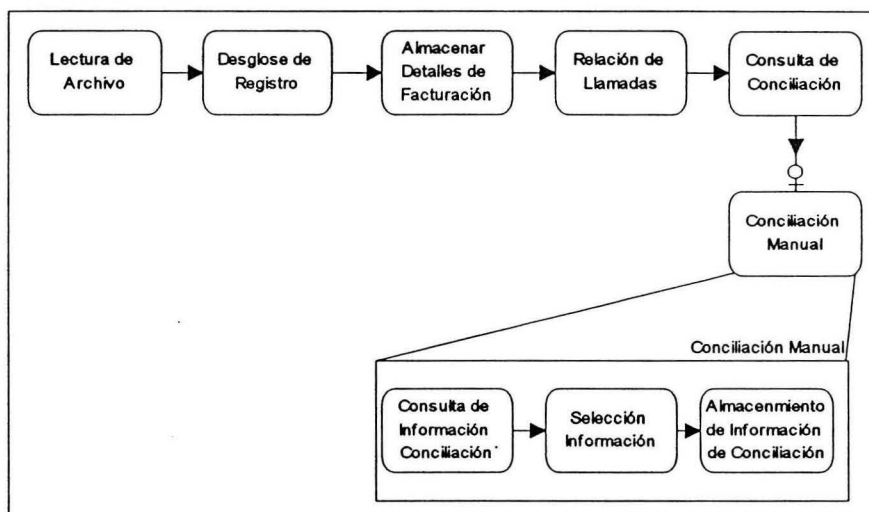


Figura 4.25. Dependencia de procesos (Conciliación de Llamadas).

El proceso de conciliación de llamadas (Figura 4.24) llevará a cabo la lectura del archivo del detalle de las llamadas de facturación, el cual se leerá línea por línea. Posteriormente la línea leída del archivo se desglosará en campos para ser almacenada en la base de datos. Al tener todos los datos almacenados se realizará la relación de los detalles de la facturación con las llamadas recopiladas. Después de haber relacionado las llamadas se ejecutará una consulta para mostrar como han sido conciliadas las llamadas y cuales no han podido ser relacionadas. En caso de que existan llamadas sin relacionarse se podrá ejecutar un proceso de conciliación manual, que será la consulta de la información de conciliación, la selección de información para realizar la asociación de las llamadas recopiladas con los detalles de la facturación y por último almacenar los cambios realizados en la conciliación.

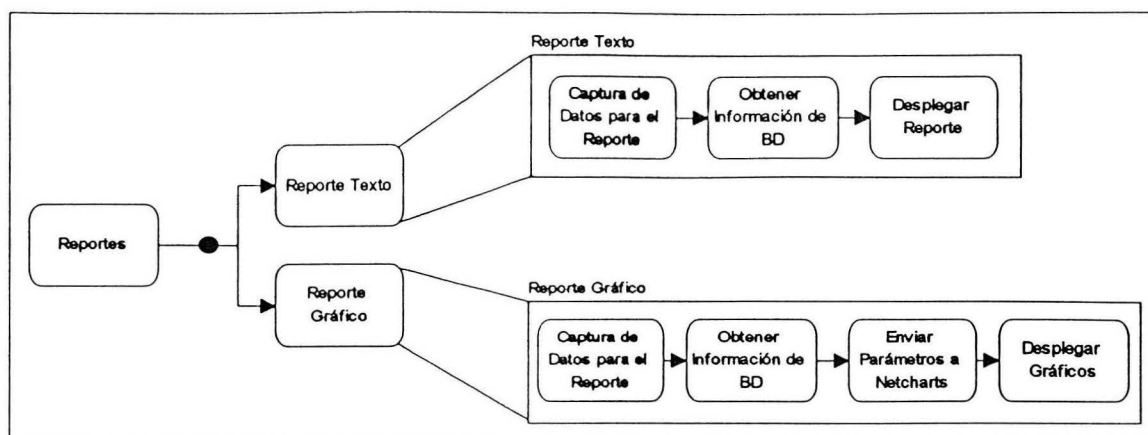


Figura 4.26. Dependencia de procesos (Reportes).

El proceso de reportes será la ejecución del proceso que permitirá ejecutar reportes tipo texto o gráfico (Figura 4.25). El proceso de reporte de texto se integrará por la captura de datos para la generación del reporte de modo que con la información capturada se pueda extraer la información de la base de datos para desplegar el reporte solicitado por el usuario. El proceso de reporte gráfico estará formado por casi los mismos procesos que el reporte de texto, solo que después de obtener la información será necesario darle formato para que el controlador de NetCharts pueda procesarla genere la gráfica y luego se despliegue.

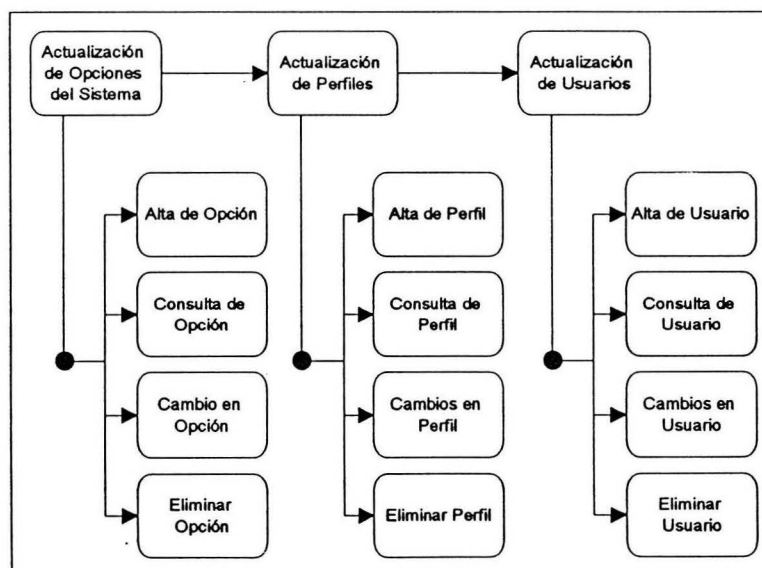


Figura 4.27. Dependencia de Procesos (Perfiles y Usuarios).

El proceso de perfiles y usuarios permitirá configurar los accesos a las opciones y a la información del sistema para cada usuario y sus necesidades. Inicialmente será necesario que sea creada una opción del sistema para poder configurar algún perfil. El proceso de actualización de las opciones del sistema permitirá ejecutar funciones ya sea para crear, consultar, modificar o eliminar opciones del

sistema. Después de contar con una opción, será necesario crear un perfil antes de poder asignarle accesos a un usuario. Para esto se cuenta con el proceso de actualización de perfiles que permitirá también crear, consultar, modificar o eliminar perfiles para el acceso de opciones. Por último, será necesario crear un usuario y asignar el perfil que ajuste a sus necesidades permitiendo o no acceder las opciones del sistema. Esto se logra mediante la actualización de usuarios que permitirá crear, modificar, consultar o eliminar a los usuarios del sistema y asociarlos con el perfil correspondiente.

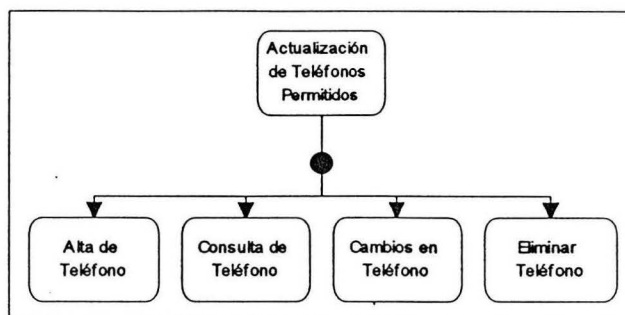


Figura 4.28. Dependencia de Procesos (Actualizador de Teléfonos Permitidos).

El proceso de actualización de teléfonos permitidos serán las opciones de crear, consultar, modificar y eliminar los teléfonos que se involucran con la empresa para que estos puedan ser identificados en los reportes de las llamadas recopiladas.

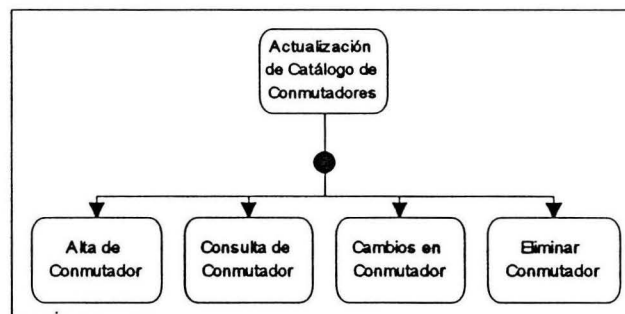


Figura 4.29. Dependencia de Procesos (Actualizador de Catálogo de Conmutadores).

El proceso de actualización del catálogo de conmutadores estará formado por las opciones de crear, consultar, modificar y eliminar conmutadores para poder identificar cada uno de los conmutadores instalados en la red de voz de HEB México.

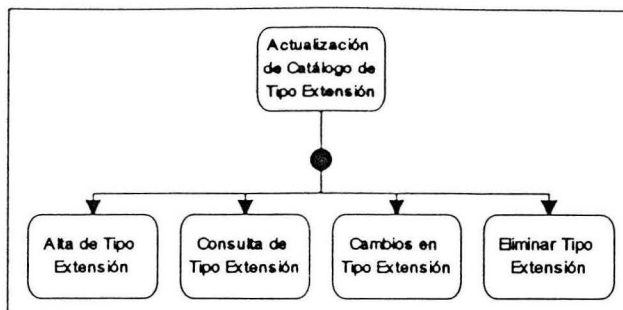


Figura 4.30. Dependencia de Procesos (Actualizador de Catálogo de Tipo Extensión).

El proceso de actualización del catálogo de tipo de extensión consistirá en las opciones de crear, consultar, modificar y eliminar tipos de extensión de forma que si la operación normal de las extensiones cambia, esta pueda ser reflejada.

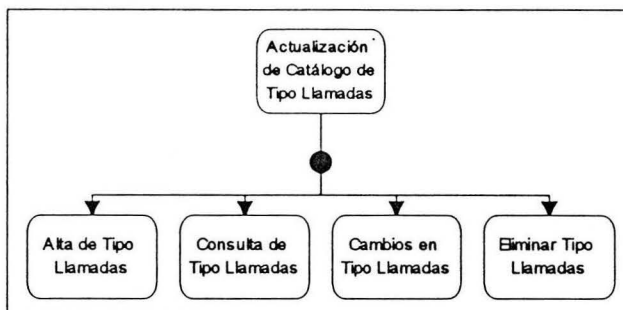


Figura 4.31. Dependencia de Procesos (Actualizador de Catálogo de Tipo Llamadas).

El proceso de actualización del catálogo de tipo de llamadas consistirá en las opciones de crear, consultar, modificar y eliminar tipos de llamadas de modo que si existe un cambio en los servicios de telefonía estos puedan ser configurados.

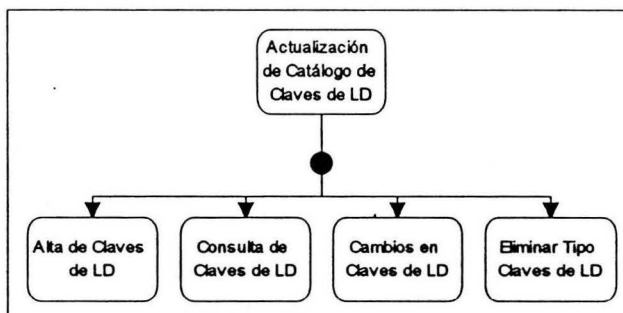


Figura 4.32. Dependencia de Procesos (Actualizador de Catálogo de Claves de LD).

El proceso de actualización del catálogo de claves de larga distancia estará formado por las opciones de crear, consultar, modificar y eliminar claves de larga distancia para que puedan ser identificadas por el sistema las localidades a donde se realizan las llamadas.

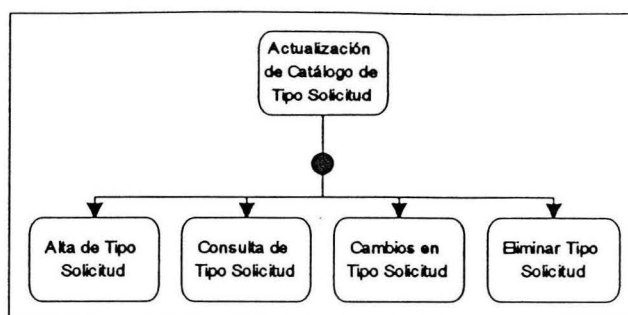


Figura 4.33. Dependencia de Procesos (Actualizador de Catálogo de Tipo Solicitud).

El proceso de actualización del catálogo de tipo de solicitud se formará por las opciones de crear, consultar, modificar y eliminar tipos de solicitudes de modo que si existe un cambio en la forma de realizar las solicitudes se pueda configurar en este catálogo.

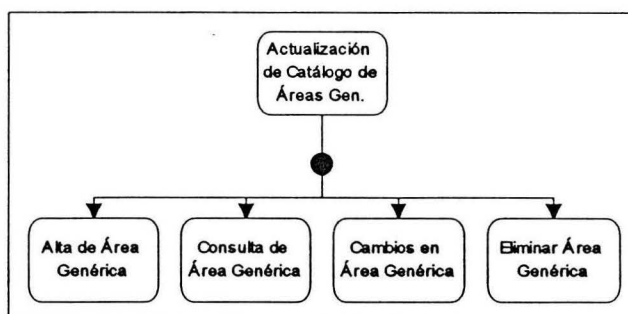


Figura 4.34. Dependencia de Procesos (Actualizador de Catálogo de Áreas Genéricas).

El proceso de actualización del catálogo de áreas genéricas estará formado por las opciones de crear, consultar, modificar y eliminar áreas genéricas para que estas puedan asociarse con su extensión correspondiente.

A continuación se presenta una relación de los procesos y las entidades con los que cada uno tiene interacción. Además la tabla muestra el tipo de interacción que tiene el proceso con la entidad.

Entidades Procesos																
	Conmutador	Extensión	Unegocio	Departamento	Socio	Llamada	ClaveLD	Tipo_Llamada	Llamada_Telmex	Tipo_Extensión	Ubicación	Tipo_Solicitud	Solicitud	Perfil	Usuario	Opc_Sistema
Acceso														L	L	L
Admon. Conmutador	CM LE	CM LE		L	L		CM LE	CM LE		CM LE		ML E	ML E	L	L	L
Actualización Información Conmutador	CM LE	CM LE		L	L		CM LE	CM LE		CM LE				L	L	L
Admon. de Solicitudes	L	L		L	L					L		ML E	ML E	L	L	L
Actualizar Relación Socio-Ext.	CM LE	CM LE		L	L			L		L				L	L	L
Crear Relación Socio-Ext.	L	L		L	L			L		L				L	L	L
Eliminar Relación Socio-Ext.	L	L		L	L			L		L				L	L	L
Modificar Caract. de Extensión	CM LE	CM LE						CM LE		L				L	L	L
Cambios en información de Extensión														L	L	L
Desactivar Extensión														L	L	L
Activación de Extensión														L	L	L
Construcción de comandos														L	L	L
Ejecución de Comandos														L	L	L
Verificar éxito de ejecución														L	L	L
Desplegar Información del Conmutador														L	L	L
Información del Conmutador para Solicitud	L	L		L	L			L		L		L	L	L	L	L
Construcción de comandos para solicitud														L	L	L
Consulta de Solicitudes Pendientes	L	L		L	L			L		L		L	L	L	L	L
Cancelación de Solicitudes	L	L		L	L			L		L		L	ML E	L	L	L
Actualizar Información Conmutador	L	L		L	L			L		L		L	L	L	L	L
Construcción comandos con Inf. Sel.														L	L	L
Conciliación de Llamadas						ML	L	L	CL					L	L	L
Lectura de Archivo														L	L	L
Desglose de Registro														L	L	L
Almacenar Detalles de Facturación									C							
Relación de Llamadas						M	L	L	L					L	L	L
Consulta de Conciliación						L	L	L	L					L	L	L
Conciliación Manual						ML	L	L	L					L	L	L
Consulta de Información Conciliación						L	L	L	L					L	L	L
Selección de Información						L	L	L	L					L	L	L
Almacenar Información Conciliación						ML	L	L	L					L	L	L
Generación de Solicitud	L	L		L	L					L		L	CM L	L	L	L
Captura y Modificación de Solicitud	L	L		L	L					L		L	CM L	L	L	L
Validación de Datos de Solicitud	L	L		L	L					L		L	L	L	L	L
Almacenar datos de Solicitud	L	L		L	L					L		L	CM L	L	L	L
Recopilación de Llamadas						C	L	L								
Recepción de datos de Llamada																
Almacenar en Archivo Local																

Entidades Procesos	Commutador	Extension	Unegocio	Departamento	Socio	Llamada	ClaveLD	Tipo_Llamada	Llamada_Telmex	Tipo_Extensión	Ubicación	Tipo_Solicitud	Solicitud	Perfil	Usuario	Opc_Sistema	AreaGenerica	Tel_permitido
Clasificación de Llamadas							L	L										
Verificación de Conexión con BD																		
Almacenar en BD						C	L	L										
Almacenar en BD los Datos del Archivo Loc.						C	L	L										
Leer Archivo Local																		
Depurar Archivo Local																		
Reportes	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Reporte Texto	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Captura de Datos para el Reporte	L	L	L	L	L			L		L	L			L	L	L		L
Obtener Información de BD	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Desplegar Reporte	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Reporte Gráfico	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Enviar Parámetros a Netcharts	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Desplegar Gráficos	L	L	L	L	L	L	L	L	L	L	L			L	L	L		L
Config. del Sistema					LE				LE				LE	CM LE	CM LE	CM LE		CM LE
Usuarios y Perfiles														CM LE	CM LE	CM LE		
Actualizador de Teléfonos Permitidos																		CM LE
Depuración de BD						LE			LE				LE	L	L	L		
Captura de Información a Depurar														L	L	L		
Eliminar Datos de Tablas Operativas						E			E				E	L	L	L		
Actualización de Perfiles														CM LE		CM LE		
Alta de Perfil														CL		CL		
Consulta de Perfil														L		L		
Cambios en Perfil														ML		ML		
Eliminar Perfil														EL		EL		
Actualización de Usuarios															CM LE			
Alta de Usuario															CL			
Consulta de Usuario															L			
Cambios en Usuario															ML			
Eliminar Usuario															EL			
Asociación de Perfiles y Usuarios														ML E	ML E	L		
Agregar Usuario a Perfil														ML	ML	L		
Eliminar Usuario de Perfil														LE	LE	L		
Actualizador de Teléfonos Permitidos																		CM LE
Alta de Teléfono																		CL
Consulta de Teléfono																		L
Cambios en Teléfono																		ML
Eliminar Teléfono																		EL
Actualizador de Áreas Genéricas																		CM LE
Alta de Área Genérica																		CL
Consulta de Área Genérica																		L
Cambios en Área Genérica																		ML
Eliminar Área Genérica																		EL

Tabla 4.5. Relación de Entidades y Procesos. (Crear, Modificar, Leer, Eliminar).

5. DISEÑO

A continuación se describen los resultados obtenidos en la etapa de diseño o JAD de la metodología de desarrollo rápido de aplicaciones (RAD). Se incluye la arquitectura del sistema, la dependencia entre procesos y entidades, el modelo de entidad-relación y los resultados de la evaluación de los prototipos, además de otra información relevante.

5.1. Arquitectura.

A continuación se describen los módulos que componen la arquitectura del sistema representada en la figura 5.1.

Conmutador

Cada una de las oficinas y sucursales de la empresa cuenta con un conmutador Ericsson MD-110, mediante los cuales se enlazan las llamadas de las extensiones a entidades externas u otras extensiones de la empresa. Este dispositivo registra los datos de las llamadas realizadas, y éstos pueden ser registrados mediante un puerto serial RS-232 haciendo uso de un proceso que los recopile.

BD Corporativa

En la base de datos corporativa se encuentra almacenada la información de los empleados de la empresa, su localización, su extensión, el área a la que pertenecen, su unidad de negocios, y otros datos generales. Esta información será necesaria para la elaboración de los reportes.

BD Sistema de Tarificación

La base de datos del sistema de tarificación contendrá la información extraída de los conmutadores, dentro de un modelo de datos que contemple el volumen de la información y las búsquedas de los reportes.

Módulo de Recopilación de Datos

Este módulo recopilará la información enviada por el conmutador y la enviará al módulo de almacenamiento.

Módulo de Almacenamiento

Este módulo registrará en la base de datos del sistema de tarificación la información enviada por el módulo de recopilación de datos. Sólo en casos de contingencia utilizará el almacenamiento local para que la información no se extravíe, y será enviada cuando se restablezca el enlace con la base de datos del sistema de tarificación.

Módulo de Depuración de Archivos

Mediante este módulo se depurarán los archivos de almacenamiento local evitando que estos tengan un volumen no deseado.

Módulo de Depuración de BD

Mediante este módulo se depurarán las tablas operativas de la base de datos eliminando datos con una antigüedad determinada para controlar el crecimiento de la base de datos y eliminar información innecesaria.

Módulo de Solicitudes

Este módulo registrará las solicitudes de alta y cambios en las extensiones que tengan los socios.

Módulo de Administración del Conmutador

A través de este módulo será posible administrar la información que es manejada por el conmutador como extensiones, puertos y usuarios, sin tener que utilizar los comandos del lenguaje del conmutador que se utilizan actualmente. Además presentará la información necesaria para satisfacer las solicitudes de alta y cambios en las extensiones.

Módulo de Reportes

Mediante este módulo será posible explotar la información almacenada en la base de datos del sistema de tarificación incluyendo niveles organizacionales, tiempos y tipos de llamadas. También se incluirán en los reportes puntos de alerta basados en reglas del negocio.

Módulo de Conciliación con Telmex

Este módulo permitirá realizar una conciliación de los datos de facturación de Telmex con los datos obtenidos por el sistema y evaluar así, los montos de las llamadas realizadas y los errores de facturación.

Módulo de Seguridad

Este módulo administrará la seguridad del sistema permitiendo a los usuarios consultar solamente la información que les involucra.

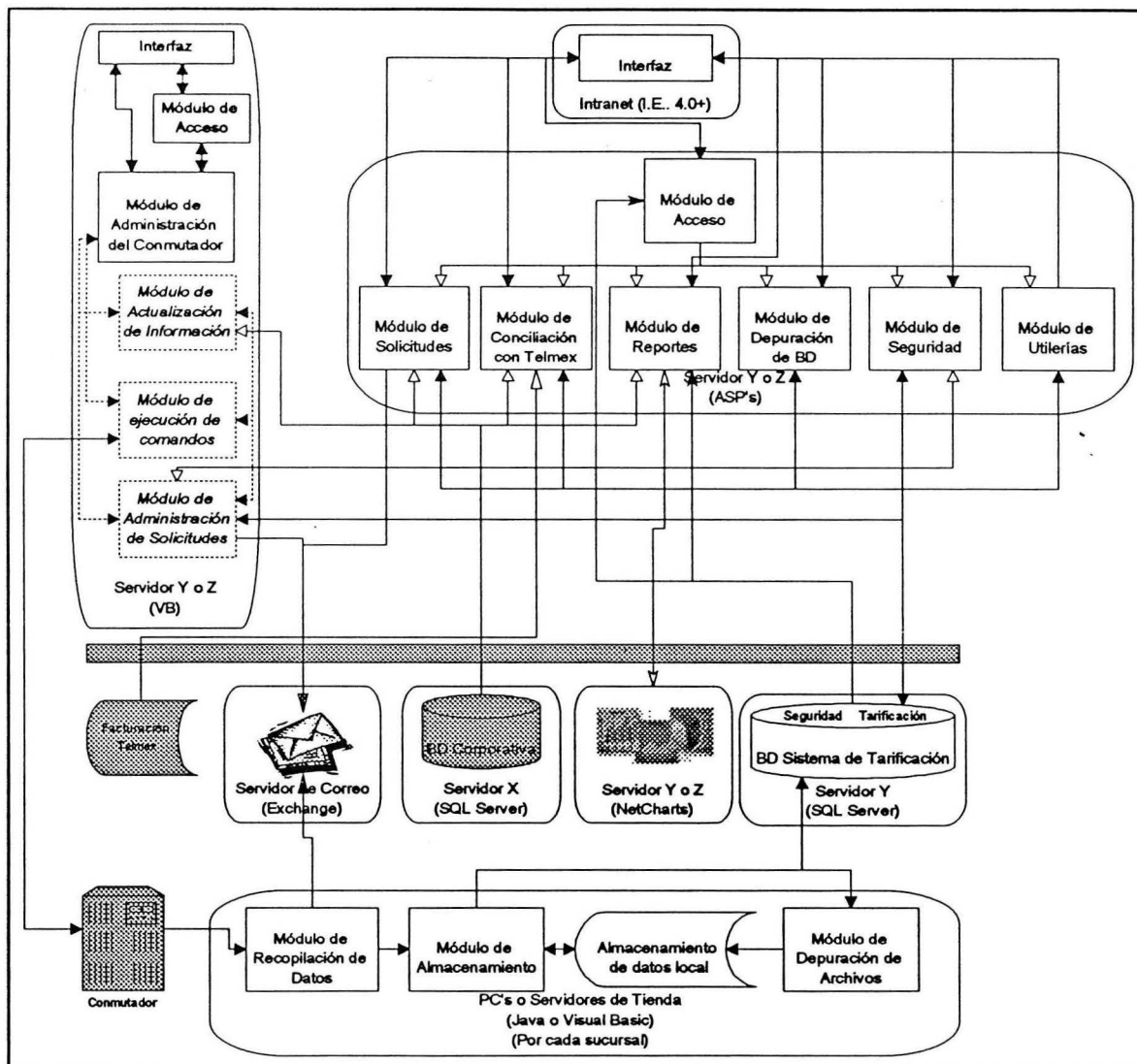


Figura 5.1. Arquitectura del Sistema.

5.2. Dependencia entre procesos y entidades.

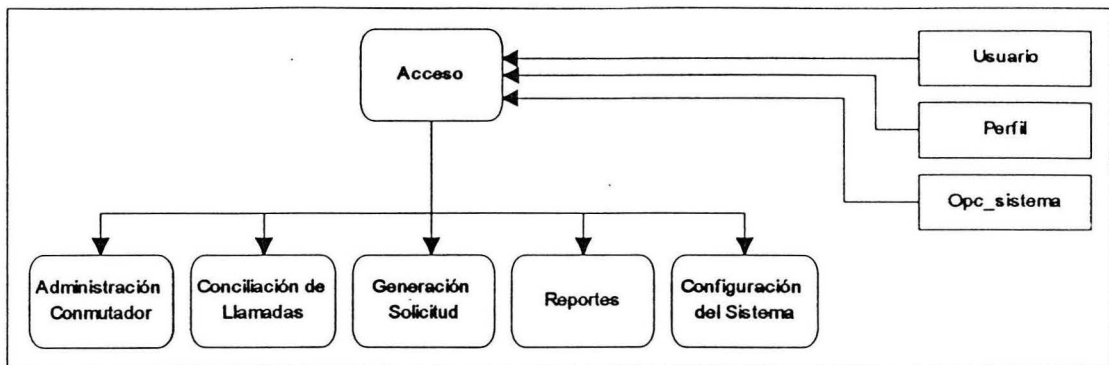


Figura 5.2. Relación Procesos y Entidades (Acceso).

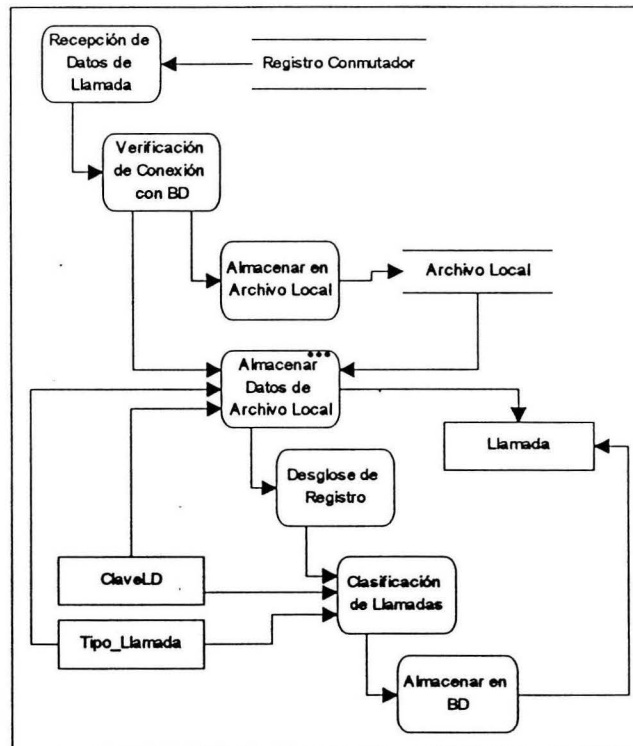


Figura 5.3. Relación Procesos y Entidades (Recopilación de Llamadas).

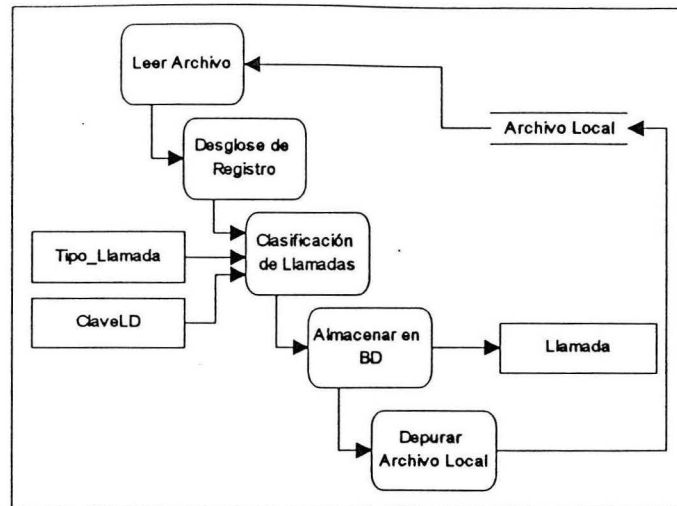


Figura 5.4. Relación Procesos y Entidades (Almacenar Datos de Archivo Local).

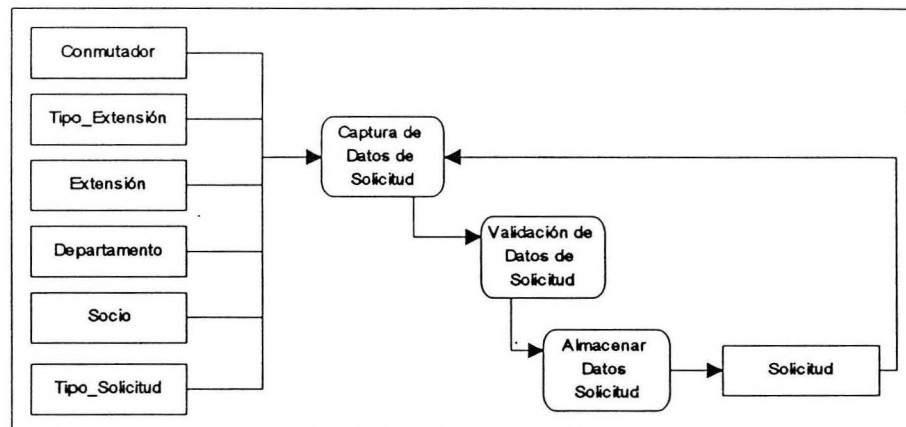


Figura 5.5. Relación Procesos y Entidades (Generación Solicitud).

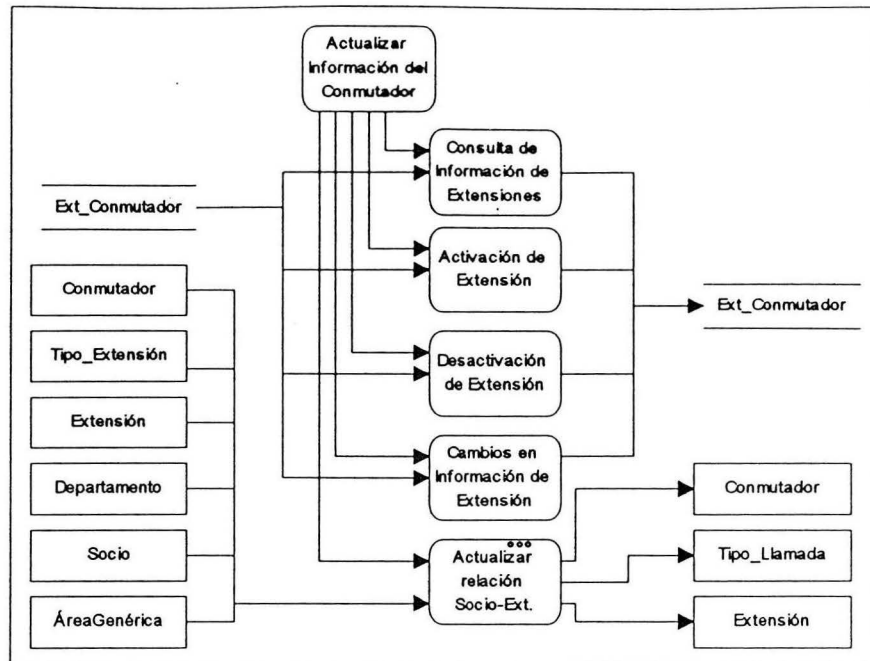


Figura 5.6. Relación Procesos y Entidades (Actualizar Información del Conmutador).

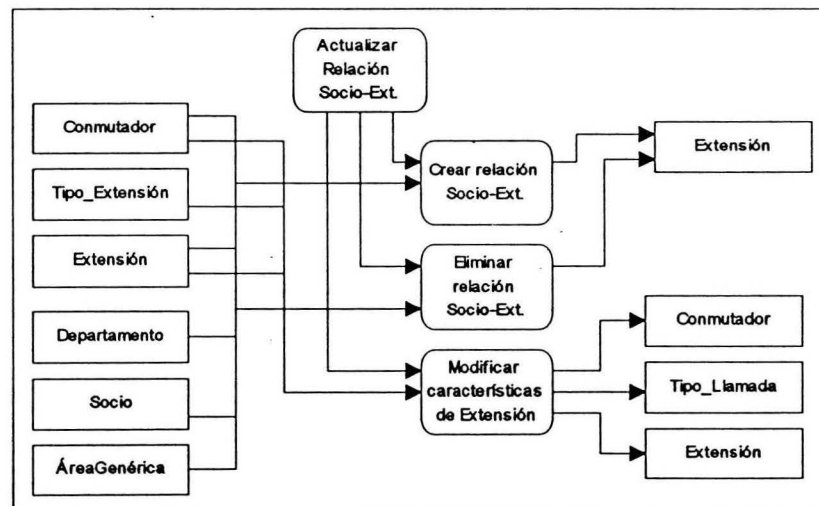


Figura 5.7. Relación Procesos y Entidades (Actualización de Relación Socio-Extensión).

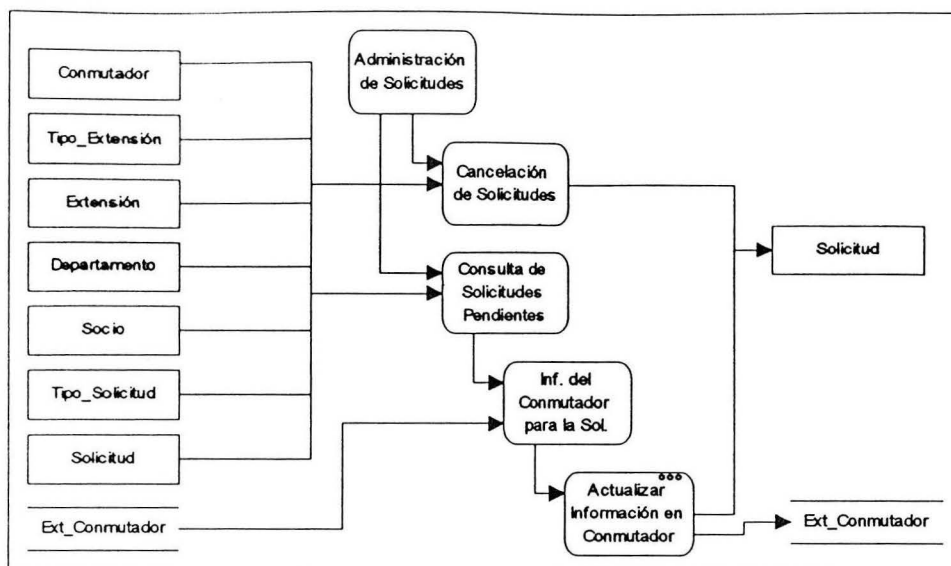


Figura 5.8. Relación Procesos y Entidades (Administración de Solicitudes).

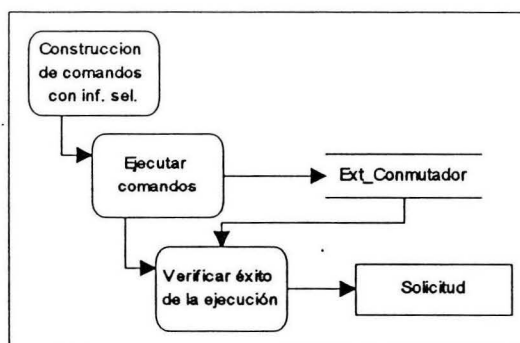


Figura 5.9. Relación Procesos y Entidades (Actualizar Información en Conmutador).

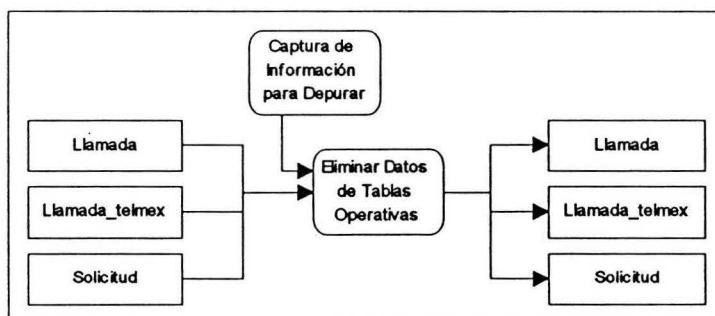


Figura 5.10. Relación Procesos y Entidades (Depuración BD).

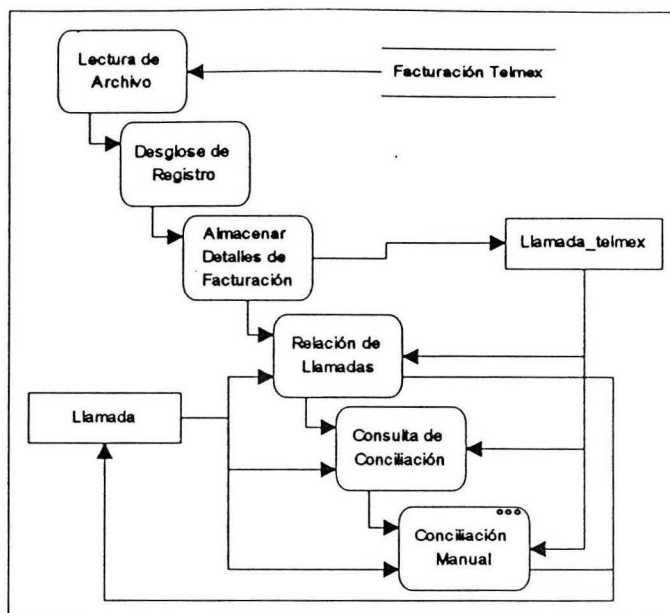


Figura 5.11. Relación Procesos y Entidades (Conciliación de Llamadas).

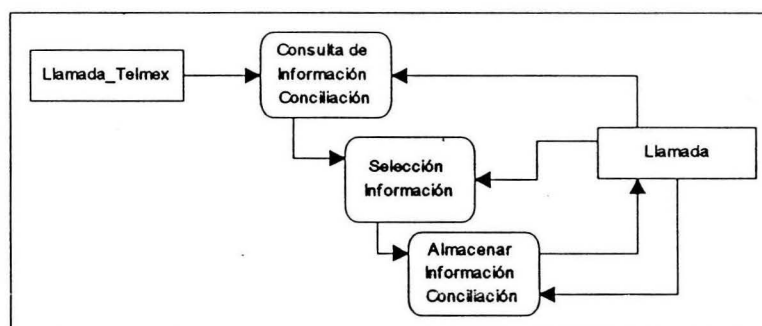


Figura 5.12. Relación Procesos y Entidades (Conciliación Manual).

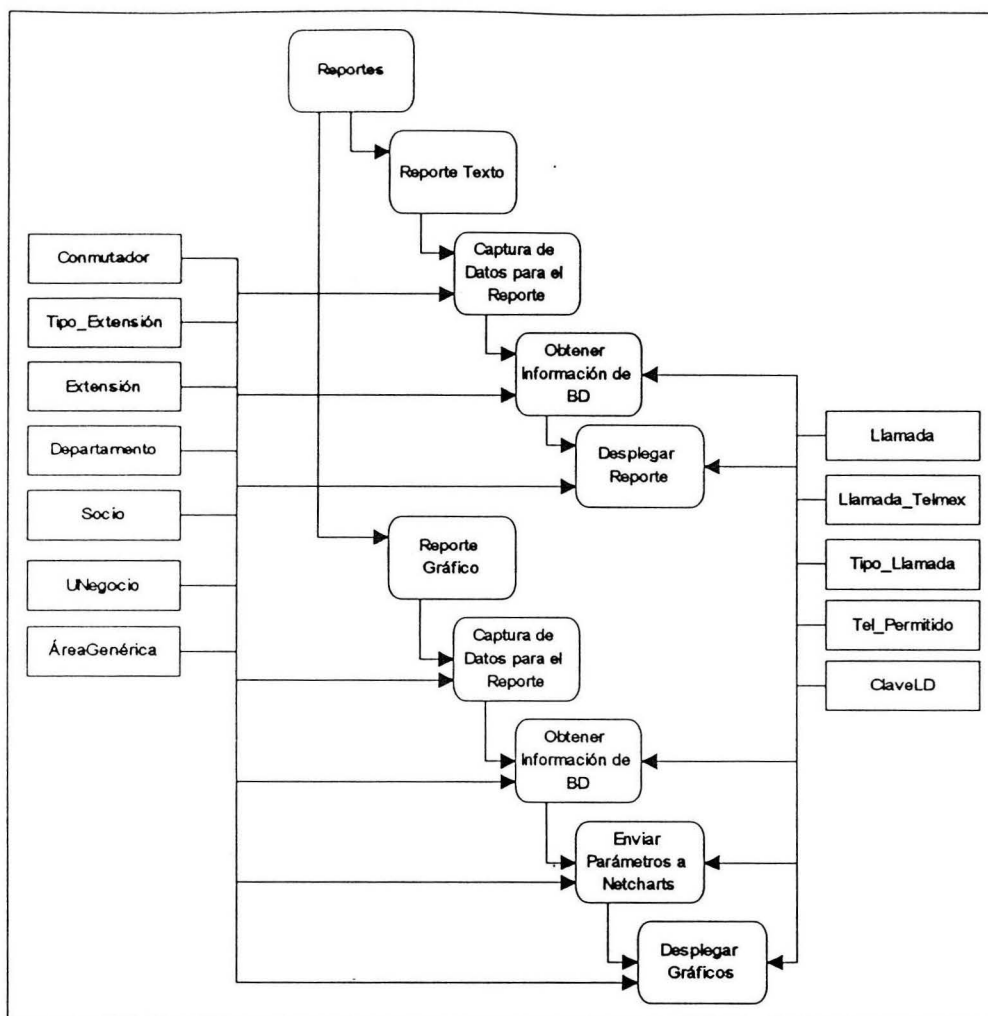


Figura 5.13. Relación Procesos y Entidades (Reportes).

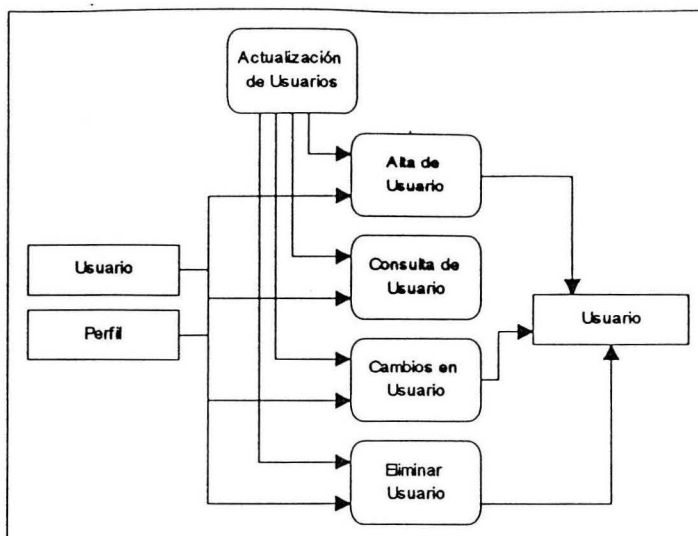


Figura 5.14. Relación Procesos y Entidades (Actualización de Usuarios).

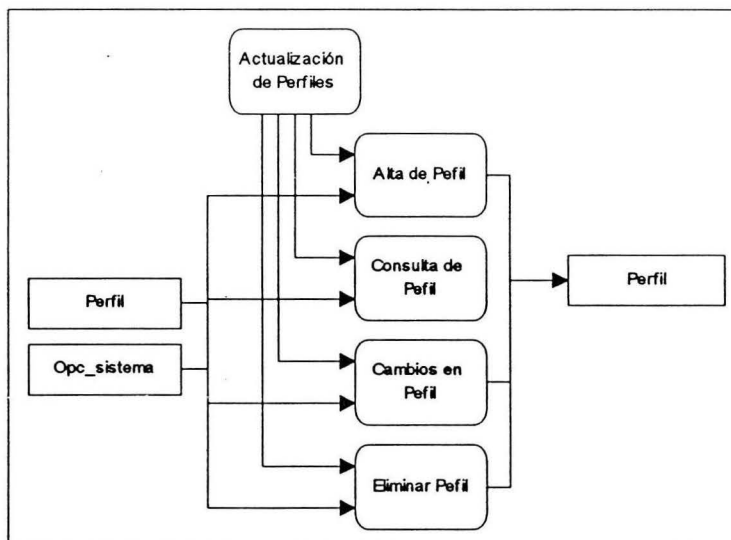


Figura 5.15. Relación Procesos y Entidades (Actualización de Perfiles).

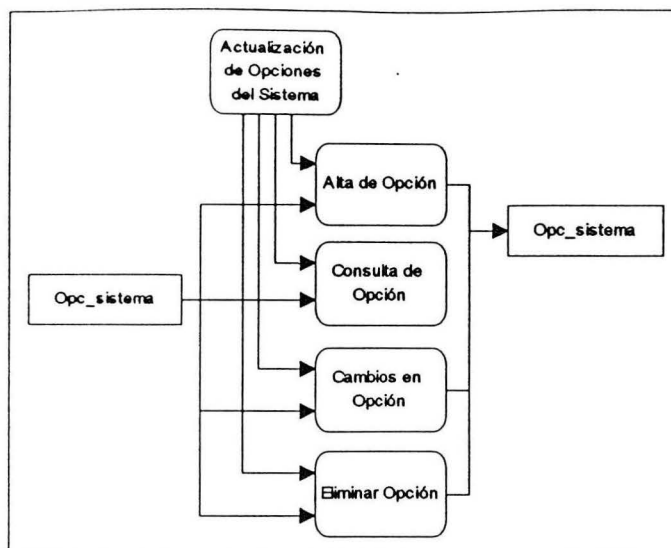


Figura 5.16. Relación Procesos y Entidades (Asociación de Perfiles y Usuarios).

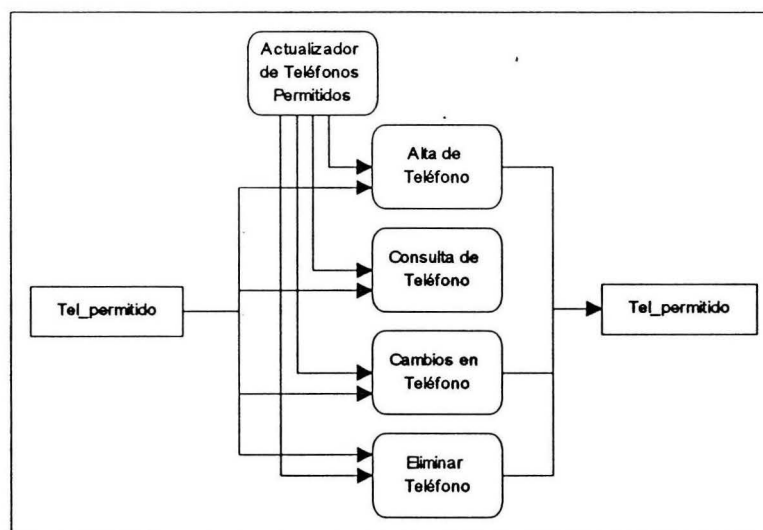


Figura 5.17. Relación Procesos y Entidades (Actualizador de Teléfonos Permitidos).

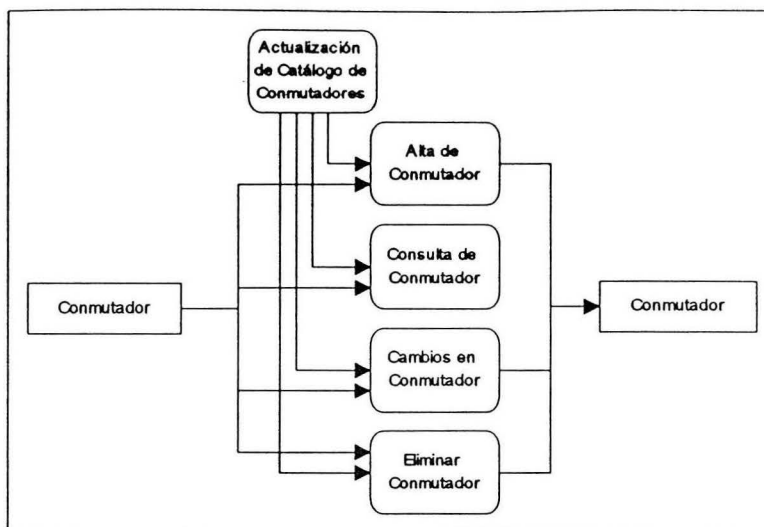


Figura 5.18. Relación Procesos y Entidades (Actualización de Catálogo de Conmutadores).

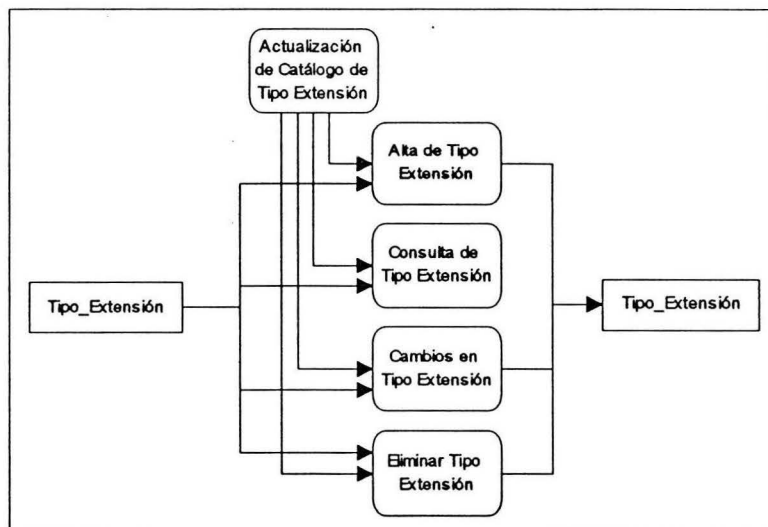


Figura 5.19. Relación Procesos y Entidades (Actualización de Catálogo Tipo Extensión).

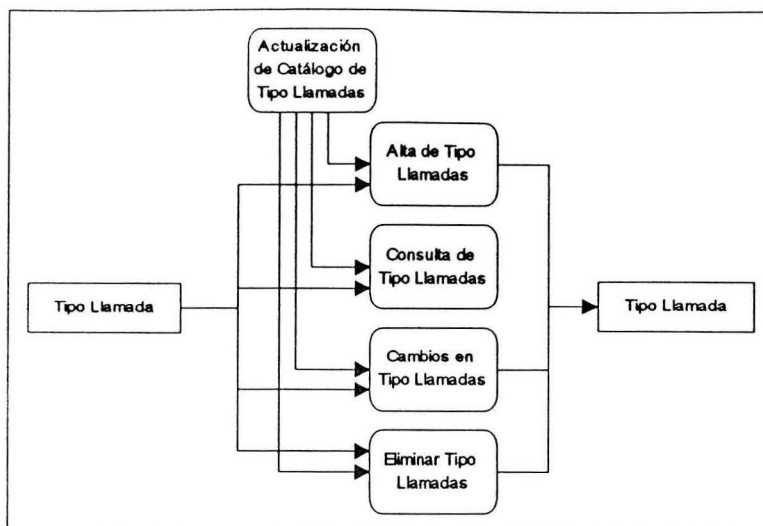


Figura 5.20. Relación Procesos y Entidades (Actualización de Catálogo de Tipo Llamadas).

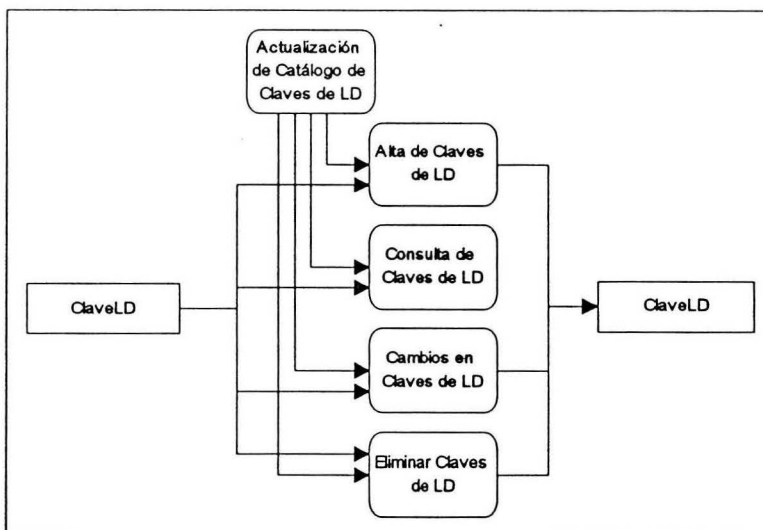


Figura 5.21. Relación Procesos y Entidades (Actualización de Catálogo de Claves de LD).

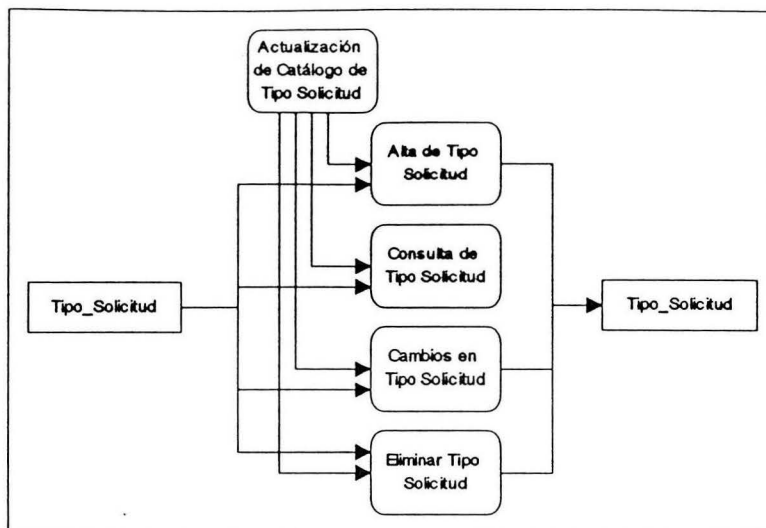


Figura 5.22. Relación Procesos y Entidades (Actualización de Catálogo de Tipo de Solicitud).

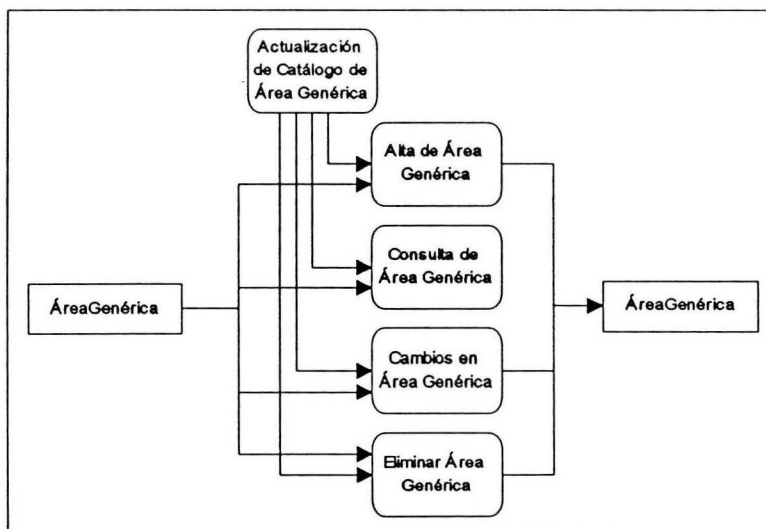


Figura 5.23. Relación Procesos y Entidades (Actualización de Catálogo de Áreas Genéricas).

5.3. Modelo entidad-relación.

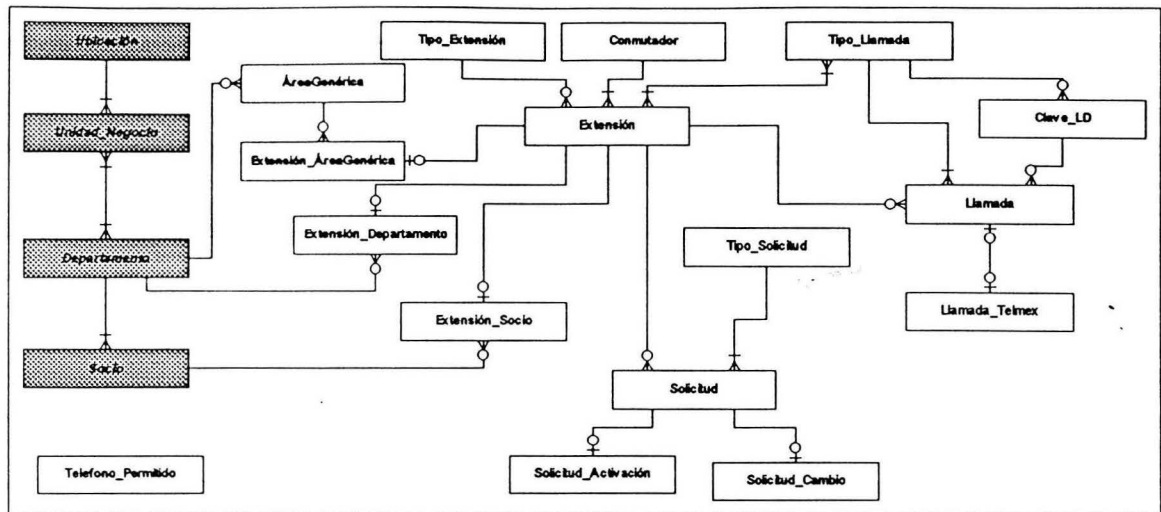


Figura 5.24. Diagrama de entidad-relación del sistema.

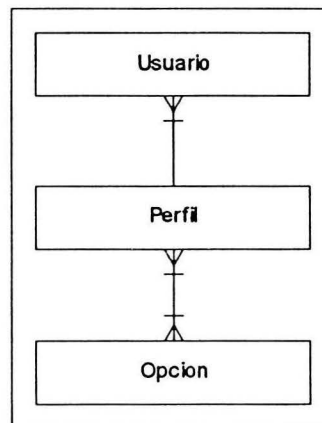


Figura 5.25. Diagrama de entidad-relación para la seguridad del sistema.

5.4. Estándares.

A continuación se explicarán los estándares de diseño utilizados por el departamento de sistemas de HEB México para la aplicación en Intranet y Visual Basic.

5.4.1. Estándares de diseño en Intranet.

A continuación se muestran algunos ejemplos de diseños de pantallas de uso común.

Encabezado de reporte.

Es el disparador del reporte, es decir, la pantalla en la que se capturan todos los datos necesarios para generar el reporte. En la Intranet este disparador se debe colocar como el frame superior de dos frames horizontales, en el inferior se desplegará el reporte, de modo que si se desea generar otro reporte no se necesite cambiar de pantalla.

Figura 5.26. Ejemplo de encabezado de reporte.

Cuerpo del reporte.

Es el frame inferior que se presenta inicialmente antes de que el reporte sea generado.



Figura 5.27. Ejemplo de cuerpo del reporte.

H·E·B
ADMINISTRACION DE LOCALES COMERCIALES
CATALOGO DE CLIENTES

No. Cliente:

Nombre:

Giro: Tipo Cliente:

Razón Social:

Calle: Colonia:

Ciudad: Código Postal:

Correo:

Teléfono 1: Teléfono 2: No de Fax:

Extension 1: Extension 2:

Email: RFC:

Figura 5.29. Ejemplo de Actualizador (pantalla inicial).

Actualizador (pantalla de consulta).

En esta pantalla se presentará la información consultada desde la pantalla inicial. Se deben mostrar todos los campos de captura habilitados, excepto los campos llaves. En esta pantalla son necesarios los botones de actualizar, eliminar y regresar, sin embargo pueden ser agregados más de estos si se requiere de otra funcionalidad.

H·E·B
ADMINISTRACION DE LOCALES COMERCIALES
CATALOGO DE CLIENTES

No. Cliente:

Nombre:

Giro: Tipo Cliente:

Razón Social:

Calle: Colonia:

Ciudad: Código Postal:

Correo:

Teléfono 1: Teléfono 2: No de Fax:

Extension 1: Extension 2:

Email: RFC:

Figura 5.30. Ejemplo de Actualizador (pantalla consulta).

Colores, tamaños y tipos de letra.

A continuación se muestra una tabla con los colores, tamaños y tipos de letra que deben tener cada una de las pantallas básicas.

	Tipo de Letra	Tamaño de Letra	Color de Letra	Fondo
Encabezados	Tahoma	3-Títulos 2-Normal	Blanco	Azul Hex. #336699
Cuerpo	Tahoma	3-Títulos 2-Normal	Negro	Beige Hex. #E7E7D6
Reporte	Tahoma	3-Títulos 2-Normal	Negro	Blanco Hex. #H00FFF
Actualizadores	Tahoma	3-Títulos 2-Normal	Negro-Normal Azul-Llaves	Beige Hex. #E7E7D6
Menús	Tahoma	3-Títulos 2-Normal	Negro	Beige Hex. #E7E7D6
Accesos	Tahoma	3-Títulos 2-Normal	Negro	Beige Hex. #E7E7D6

Tabla 5.1. Estándares de interfases para Intranet.

5.4.2. Estándares de diseño en Visual Basic.

Existen diferentes estilos de aplicaciones dentro de los ambiente GUI (Graphics User Interface). Para confirmar esta clasificación se consideran muy diversos aspectos funcionales. Sin embargo, el elemento mas importante es la cantidad y el tipo de ventanas o formas que se pueden editar en un momento dado dentro de la aplicación.

De acuerdo a los criterios ya expuestos se pueden mencionar los siguientes estilos de aplicaciones:

- Accesorios de escritorio.
- Aplicaciones SDI (Single Document Interface)
- Aplicaciones MDI (Multiple Document Interface)
- Workspace.

El workspace es el estilo de aplicación que más se ha adaptado a las necesidades de HEB México y es por tanto alrededor del cual gravitan los estándares de Visual Basic. Esto no quiere decir que no se puedan utilizar los otros estilos o que no exista ningún estándar para ellos. Los elementos de los estándares del workspace que sean transportables a los otros estilos deberán conservar su mismo estilo.

La figura 5.20 es un ejemplo de workspace típico. Los elementos que se pueden incluir en un workspace se listan en la tabla 5.2, así como las características con las que deben de contar.

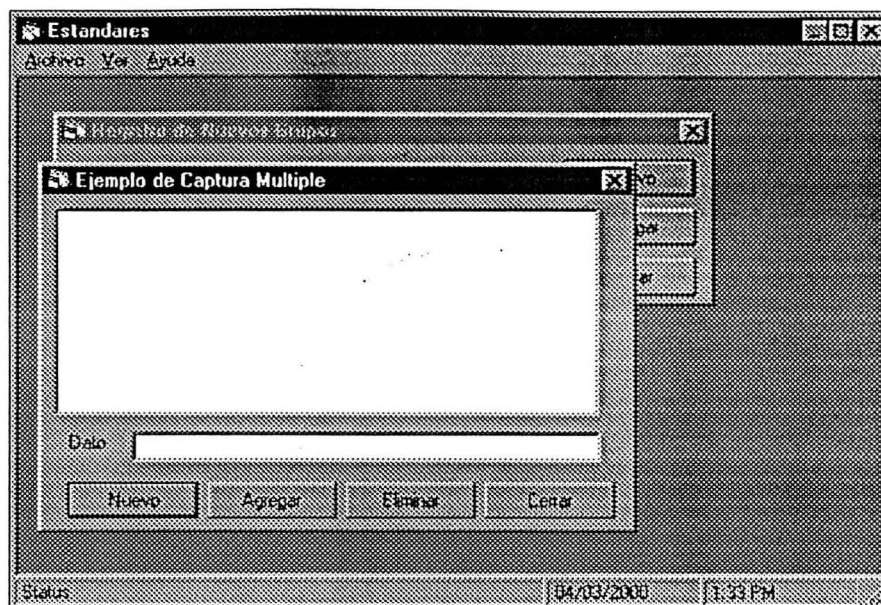


Figura 5.31. Ejemplo clásico de workspace.

Elemento/ Característica	System Menú o Control Box	Maximizar	Minimizar	Resize	BackColor	ForeColor	BevelInner	BevelOuter	BevelWidth	BorderWidth	FontName	FontSize	Bold	Italic	Alto x Ancho	Alineación
Ventana Principal	✓	✓	✓	✓												
Formas	✓	✗	✓	✗												
Panel3D					H00808080		Inset	Raised	1	3						
Diálogos Modales	✓	✗	✗	✗												
Diálogos No Modales	✓	✗	✗	✗												
Labels					H00000000 H0C0C0C0	H0C0C0C0					MS Sans Serif	8	✓			Izq. del campo
Botones					H00000000 H0C0C0C0	H0C0C0C0					MS Sans Serif	8	✗		360 x 1230 Der. de la forma	

Tabla 5.2. Elementos y características para un workspace.

5.5. Jerarquía de formas.

Las interfaces del sistema se pueden clasificar en niveles, donde cada nivel tiene una aplicación. La siguiente tabla muestra los niveles existentes y la aplicación en cuanto a las interfaces en Intranet y Visual Basic.

Forma	Aplicación
Nivel 1	Forma de acceso.
Nivel 2	Menú general.
Nivel 3	Consulta general.
Nivel 4	Actualización y consulta detallada.

Tabla 5.3. Niveles de Interfaces de Visual Basic.

Forma	Aplicación
Nivel 1	Forma de acceso.
Nivel 2	Menú general.
Nivel 3.1	Disparador de reporte.
Nivel 3.2	Pantalla inicial de actualizador.
Nivel 4.1	Reporte.
Nivel 4.2	Pantalla de consulta de actualizador.

Tabla 5.4. Niveles de Interfaces de Intranet.

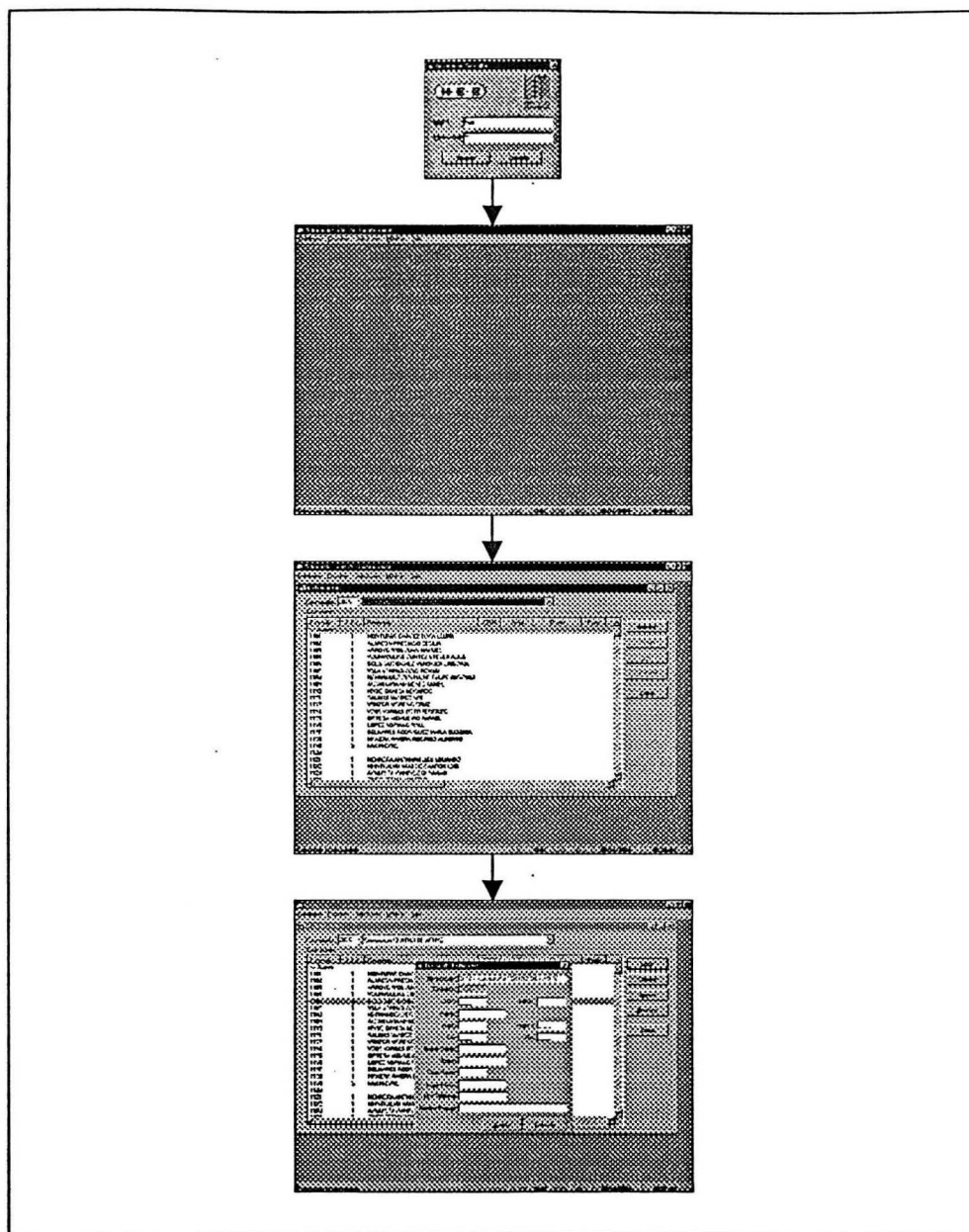


Figura 5.32. Jerarquía de formas en Visual Basic.

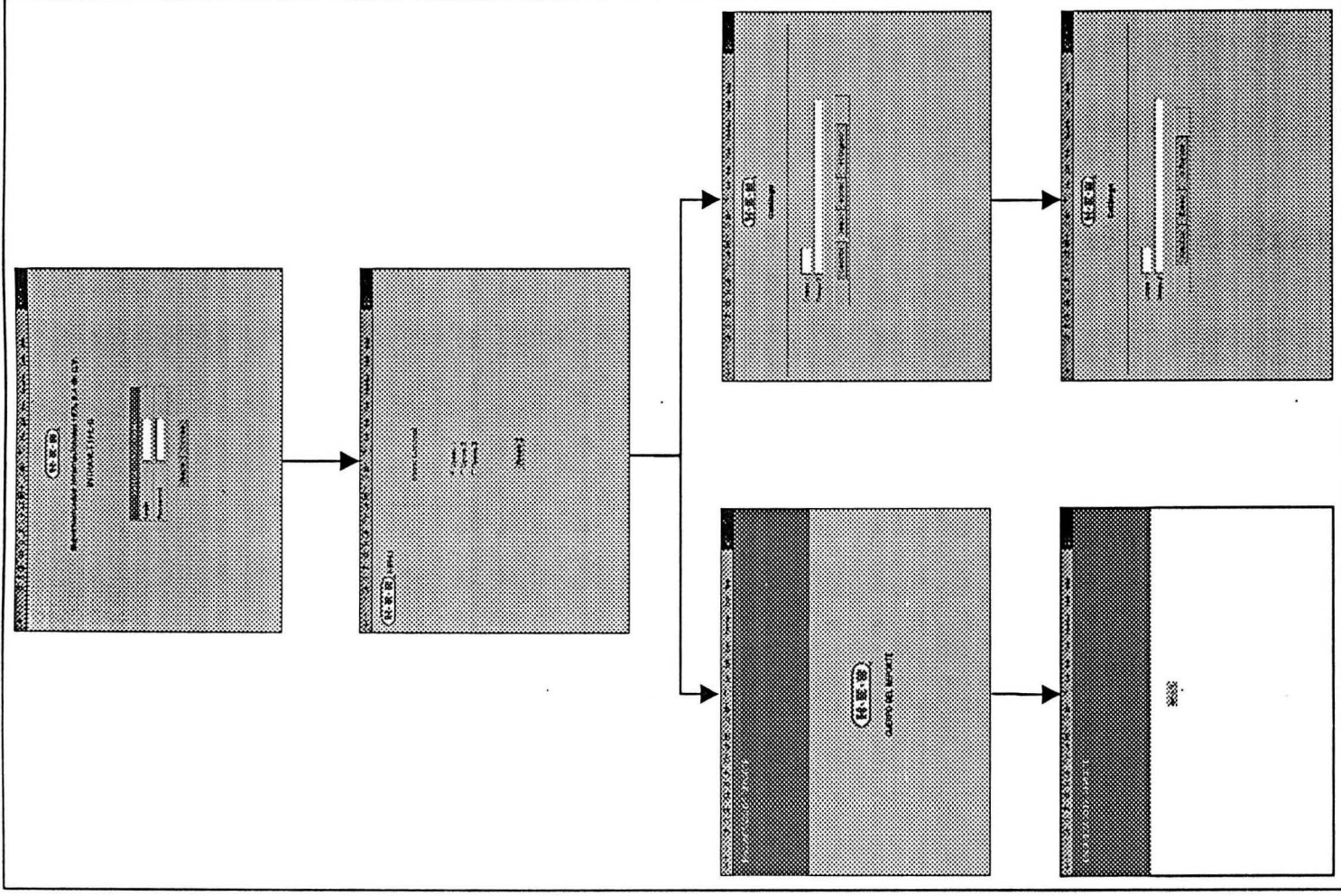


Figura 5.33. Jerarquía de Formas de Intranet.

5.6. Descripción de tablas.

Tabla: Conmutador					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ConmutadorId	Clave del conmutador	Smallint		✓	
Descripción	Descripción del conmutador	Varchar	50		
DireccionIP	Dirección IP para realizar conexión con el conmutador	Varchar	15		

Tabla: Tipo_Extensión					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
TipoExtensionId	Clave del Tipo de extensión	Smallint		✓	
Descripción	Descripción del tipo de extensión	Varchar	30		

Tabla: Extensión					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ConmutadorId	Clave del conmutador	Smallint		✓	✓
Extensión	Número de extensión	Smallint		✓	
TipoExtensionId	Clave del tipo de extensión	Smallint			✓
CTUA	CTUA de la extensión	Smallint			
Signal	Señal	Varchar	4		
Puerto	Puerto de la extensión	Varchar	12		
Rack	Rack de la extensión	Varchar	2		
Regleta	Regleta de la extensión	Smallint			
Fila	Fila de la regleta	Smallint			
Par	Par de la regleta	Smallint			
Raiser4		Varchar	8		
Raiser		Varchar	8		
CRaiser		Varchar	2		
LFisico	Lugar físico de la extensión	Varchar	7		
TipoTel	Tipo de aparato tel. de la ext.	Varchar	5		
NombreDisplay	Título de despliegue en T. Digital	Varchar	20		

Tabla: Ubicación					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
UbicacionId	Clave de la ubicación	Smallint		✓	
Descripción	Descripción de la ubicación	Varchar	50		
ConmutadorId	Clave del conmutador de la ubic.	Smallint			✓

Tabla: Unegocio					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
UnegocioId	Clave de la unidad de negocio	Varchar	5	✓	
Descripción	Descripción de la unidad de neg.	Varchar	50		
UbicacionId	Clave de la ubicación de la UN.	Smallint			✓

Tabla: Departamento					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Departamentoid	Clave del departamento	Varchar	10	✓	
Descripción	Descripción del departamento	Varchar	50		

Tabla: Socio					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Socioid	Clave del socio	Varchar	11	✓	
Unegocioid	Clave de la UN del socio	Varchar	5		✓
Departamentoid	Clave del depto. Del socio	Varchar	10		✓
Nombre	Nombre del socio	Varchar	25		
Apatemo	Apellido paterno del socio	Varchar	20		
Amatemo	Apellido materno del socio	Varchar	20		
Mail	Correo electrónico del socio	Varchar	50		
FechaNac	Fecha de nacimiento del socio	Datetime			

Tabla: AreaGenérica					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
AreaGenericaID	Clave del area genérica	Smallint		✓	
Descripción	Descripción de la ubicación	Varchar	50		
Unegocioid	Clave de la unidad de negocio	Varchar	5		✓
Departamentoid	Clave del departamento	Varchar	10		✓

Tabla: Extension_Departamento					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ConmutadorId	Clave del Conmutador	Smallint		✓	
ExtensiónId	Número de extensión	Smallint		✓	
Unegocioid	Clave de la unidad de negocio	Varchar	5		✓
Departamentoid	Clave del departamento	Varchar	10		✓

Tabla: Extension_Socio					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ConmutadorId	Clave del Conmutador	Smallint		✓	
ExtensiónId	Número de extensión	Smallint		✓	
Socioid	Clave del Socio	Varchar	11		✓

Tabla: Extension_AreaGenérica					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ConmutadorId	Clave del Conmutador	Smallint		✓	
ExtensiónId	Número de extensión	Smallint		✓	
AreaGenericaID	Clave del area genérica	Smallint			✓

Tabla: Tipo Llamada					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
TipoLlamadaId	Clave del tipo de llamada	Smallint		✓	
Prefijo	Prefijo de marcación para el servicio telefónico.	Varchar	10		
Descripción	Descripción del tipo de llamada	Varchar	30		
TipoExtraeLada	Identificador para el tipo de extracción de la clave de LD.	Tinyint			

Tabla: ClaveLD					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
ClaveLDId	Consecutivo de la clave de LD	Smallint		✓	
TipoLlamadaId	Clave del tipo de llamada al que pertenece la clave de LD.	Smallint		✓	✓
Clave	Clave de larga distancia	Carácter	5		
Descripción	Descripción de la clave de LD	Carácter	30		

Tabla: Llamada					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
LlamadaId	Consecutivo para identificar llamada	Int		✓	
Fecha	Fecha y hora en la que se realizó la llamada	Datetime			
ConmutadorId	Clave del conmutador de donde se realizó la llamada	Smallint			✓
ExtensiónId	Número de la extensión que realizó la llamada	Smallint			✓
Llamada_TelmexId	Clave de la llamada facturada	Smallint			✓
TipoLlamadaId	Clave del tipo de llamada realizada	Smallint			✓
ClaveLDId	Clave de LD de la llamada realizada	Smallint			✓
Duración	Duración en segundos de la llamada realizada	Int			
Codigo_Autorización	Código de autorización que se utilizó para realizar la llamada.	Smallint			
Teléfono	Número telefónico marcado	Entero			

Tabla: Llamada_Telmex					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Llamada_TelmexId	Clave de la llamada facturada.	Smallint		✓	
Fecha	Fecha y hora en la que se realizó la llamada según el proveedor	Datetime			
Origen	Número telefónico de origen	Varchar	20		
Destino	Número telefónico destino	Varchar	20		
Duración	Duración en segundos de la llamada	Int			
Importe	Costo de la llamada	Decimal	17		
Pob_Destino	Nombre de la población destino	Varchar	30		
Fecha_Factura	Fecha de la factura	Smalldatetime			

Tabla: Tipo_Solicitud					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Tipo_SolicitudId	Clave del tipo de solicitud	Smallint		✓	
Descripción	Descripción del tipo de solicitud	Varchar	30		

Tabla: Solicitud					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
SolicitudId	Clave de la solicitud	Int		✓	
Tipo_SolicitudId	Clave del tipo de solicitud	Smallint		✓	✓
ConmutadorId	Clave del conmutador de la extensión de la solicitud.	Smallint			✓
Extensión	Número de extensión de la solicitud.	Smallint			✓
SocioId	Clave del socio que realiza la solicitud	Varchar	11		✓
Num_Nodo	Número del nodo del lugar del socio	Varchar	15		
Centro_Costo	Centro de costos del socio	Varchar	30		
Motivo	Motivo de la solicitud	Varchar	255		
Fecha	Fecha en la que se realizó la solicitud.	Datetime			
Estatus	Estado de la solicitud	Smallint			

Tabla: Solicitud_Act					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
SolicitudId	Clave de la solicitud.	Smallint		✓	✓
Tipo_SolicitudId	Clave del tipo de solicitud	Smallint		✓	✓
TipoLinea	Tipo de línea solicitada	Smallint			
TipoNumero	Tipo de número telefónico solicitado	Smallint			
Display	Título de despliegue en teléfonos digitales.	Varchar	20		
CodigoAut	Código de autorización a configurar para la extensión.	Varchar	5		
Fecha_inicio	Fecha de inicio del socio	Smalldatetime			

Tabla: <i>Solicitud_Camb</i>					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
SolicitudId	Clave de la solicitud.	Smallint		✓	✓
Tipo_SolicitudId	Clave del tipo de solicitud	Smallint		✓	✓
CodigoAut	Código de autorización que se desea configurar con la extensión	Varchar	5		
CodigoAnterior	Código de autorización anterior	Varchar	5		
DiaplayNuevo	Nuevo título de despliegue para teléfonos digitales.	Varchar	20		
NuevoAparato	Solicitud de un nuevo aparato	Tinyint			
NodoNuevo	El nuevo nodo a donde se moverá la extensión.	Varchar	15		
NumDirectoAnt	Número directo que se tenía anteriormente.	Varchar	15		
Ext_Directo	Número de extensión anterior.	Varchar	5		
BloqueoLD	Código de bloqueo de LD.	Varchar	4		

Tabla: <i>Tel_Permitado</i>					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Teléfonold	Clave del teléfono permitido	Smallint		✓	
Teléfono	Número telefónico permitido.	Varchar	20		
Descripción	Descripción del número telefónico	Varchar	50		

Tabla: <i>Opc_Sistema</i>					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
OpcionId	Clave de la opción del sistema.	Int		✓	
Descripción	Descripción de la opción del sistema.	Varchar	50		
TipoOpcion	Clave del Tipo de Opción (Intranet, VB o acceso a datos)	Smallint			
Agrupación	Tipo de agrupación a la que pertenece la opción.	Smallint			
Programa	Programa que debe ejecutar la opción.	Smallint			

Tabla: <i>Perfil</i>					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Perfilld	Clave del perfil	Int		✓	
Descripción	Descripción del perfil	Varchar	50		

Tabla: <i>Usuario</i>					
Campo	Descripción	Tipo de dato	Longitud	Llave	
				PK	FK
Usuariold	Clave del usuario	Varchar	10	✓	
Sociold	Clave del socio al que pertenece al usuario				✓
Password	Contraseña del usuario	Varchar	20		
Perfilld	Clave del perfil del usuario	Int			✓

5.7. Diagramas de Acción.

Recopilación de llamadas

Recibe registro de llamada por puerto serial.

IF ConexionBD THEN

Procesar Llamadas de Archivo Local

Abrir Archivo Local

DO WHILE NOT EOF

Leer Línea del Archivo.

Desglosar registro.

CREATE LLAMADA

Cerrar Archivo Local

Desglosar registro recibido

CREATE LLAMADA

ELSE

Abrir Archivo Local

Escribir Registro Recibido

Cerrar Archivo Local

Actualización de Opciones del Sistema

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoOpcionId <> "" THEN

SELECT * FROM OPC_SISTEMA WHERE OpcionID = CampoOpcionID

Desplegar datos de la opción consultada.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE OPC_SISTEMA

IF Opcion2 = Eliminar THEN

DELETE OPC_SISTEMA

IF Opción = Insertar THEN

CREATE OPC_SISTEMA

Actualización de Perfiles

```
SELECT * FROM OPC_SISTEMA
```

Crear campos para selección de opciones.

Recibe datos en la forma.

```
IF Opción = Consultar THEN
```

```
IF CampoPerfilID <> "" THEN
```

```
SELECT * FROM PERFIL WHERE PerfilID = CampoPerfilID
```

```
SELECT * FROM PERFIL_OPCION WHERE PerfilID = CampoPerfilID
```

Desplegar datos del perfil consultado.

Recibe datos en la forma.

```
IF Opcion2 = Modificar THEN
```

```
UPDATE PERFIL
```

```
UPDATE PERFIL_OPCION
```

```
IF Opcion2 = Eliminar THEN
```

```
DELETE PERFIL
```

```
DELETE PERFIL_OPCION
```

```
IF Opción = Insertar THEN
```

```
CREATE PERFIL
```

```
CREATE PERFIL_OPCION
```

Actualización de Usuarios

SELECT * FROM **PERFIL**

Llenar combo de perfiles.

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoUsuariold <> "" THEN

SELECT * FROM **USUARIO** WHERE UsuarioID = CampoUsuarioID

Desplegar datos del usuario consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE **USUARIO**

IF Opcion2 = Eliminar THEN

DELETE **USUARIO**

IF Opción = Insertar THEN

CREATE **USUARIO**

Actualización de Catálogo de Conmutadores

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoConmutadorId <> "" THEN

SELECT * FROM **CONMUTADOR** WHERE ConmutadorID = CampoConmutadorID

Desplegar datos del conmutador consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE **CONMUTADOR**

IF Opcion2 = Eliminar THEN

DELETE **CONMUTADOR**

IF Opción = Insertar THEN

CREATE **CONMUTADOR**

Actualización de Catálogo de Tipo Extensión

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoTipoExtensionId <> "" THEN

SELECT * FROM TIPO_EXTENSION

WHERE TipoExtensionID = CampoTipoExtensionID

Desplegar datos del tipo de extensión consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE TIPO_EXTENSION

IF Opcion2 = Eliminar THEN

DELETE TIPO_EXTENSION

IF Opción = Insertar THEN

CREATE TIPO_EXTENSION

Actualización de Catálogo de Tipo Solicitud

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoTipoSolicitudId <> "" THEN

SELECT * FROM TIPO_SOLICITUD WHERE TipoSolicitudID = CampoTipoSolicitudID

Desplegar datos del tipo de solicitud consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE TIPO_SOLICITUD

IF Opcion2 = Eliminar THEN

DELETE TIPO_SOLICITUD

IF Opción = Insertar THEN

CREATE TIPO_SOLICITUD

Actualización de Teléfonos Permitidos

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoTelefonold <> "" THEN

SELECT * FROM TEL_PERMITIDO WHERE TelefonoID = CampoTelefonoID

Desplegar datos del teléfono consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE TEL_PERMITIDO

IF Opcion2 = Eliminar THEN

DELETE TEL_PERMITIDO

IF Opción = Insertar THEN

CREATE TEL_PERMITIDO

Actualización de Catálogo de Tipos de Llamada

Recibe datos en la forma.

IF Opción = Consultar THEN

IF TipoLlamadaId <> "" THEN

SELECT * FROM TIPO_LLAMADA WHERE TipoLlamadaID = CampoTipoLlamadaID

Desplegar datos del tipo de llamada consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE TIPO_LLAMADA

IF Opcion2 = Eliminar THEN

DELETE TIPO_LLAMADA

IF Opción = Insertar THEN

CREATE TIPO_LLAMADA

Actualización de Catálogo de Claves de LD

SELECT * FROM TIPO_LLAMADA

Llenar combo de tipo de llamadas.

Recibe datos en la forma.

IF Opción = Consultar THEN

IF TipoLlamadaID <> "" THEN

SELECT * FROM CLAVELD WHERE TipoLlamadaID = CampoTipoLlamadaID

AND Clave = CampoClave

Desplegar datos del tipo de llamada consultado.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE CLAVELD

IF Opcion2 = Eliminar THEN

DELETE CLAVELD

IF Opción = Insertar THEN

CREATE CLAVELD

Actualización de Catálogo de Áreas Genéricas

SELECT * FROM UNEGOCIO

Llenar combo de unidades de negocio.

SELECT * FROM DEPARTAMENTO

Llenar combo de departamentos.

Recibe datos en la forma.

IF Opción = Consultar THEN

IF AreaGenericaID <> "" THEN

SELECT * FROM AREAGENERICA WHERE AreaGenericaID = CampoAreaGenericaID

Desplegar datos del área genérica consultada.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE AREAGENERICA

IF Opcion2 = Eliminar THEN

DELETE AREAGENERICA

IF Opción = Insertar THEN

CREATE AREAGENERICA

Depuración de BD

Recibir datos en la forma.

IF Opcion = Depurar

DELETE FROM **LLAMADA** WHERE Fecha BETWEEN CampoFechaInicial AND CampoFecha Final

DELETE FROM **LLAMADA_TELMEX** WHERE Fecha BETWEEN CampoFechaInicial AND CampoFecha Final

AND NOT EXISTS (SELECT * FROM **LLAMADA** WHERE LLAMADA.Llamada_TelmexID = Llamada_TelmexID)

DELETE FROM **SOLICITUD_ACT** WHERE EXISTS (SELECT * FROM **SOLICITUD** WHERE SOLICITUD.Fecha BETWEEN CampoFechaInicial AND CampoFecha Final AND SOLICITUD.Solicitud_Id = Solicitud_Id AND SOLICITUD.Tipo_solicitudID = Tipo_solicitudID)

DELETE FROM **SOLICITUD_CAMB** WHERE EXISTS (SELECT * FROM **SOLICITUD** WHERE SOLICITUD.Fecha BETWEEN CampoFechaInicial AND CampoFecha Final AND SOLICITUD.Solicitud_Id = Solicitud_Id AND SOLICITUD.Tipo_solicitudID = Tipo_solicitudID)

DELETE FROM **SOLICITUD** WHERE Fecha BETWEEN CampoFechaInicial AND CampoFecha Final

Acceso

Recibe Usuario y Contraseña en la forma.

SELECT * FROM **USUARIO** WHERE UsuarioID = CampoUsuario

IF Password = CampoContraseña THEN

SELECT *
FROM **PERFIL**, **OPCION_PERFIL**, **OPC_SISTEMA**
WHERE PERFIL.PERFILID = USUARIO.PERFIL
AND PERFIL.PERFILID = OPCION_PERFIL.PERFILID
AND OPCION_PERFIL.OPCIONID = OPC_SISTEMA.OPCIONID

Crear menú con las opciones obtenidas.

Generación de Solicitudes de Activación y Cambios en Extensiones

SELECT * FROM **SOCIO**

Llenar combo de socios.

SELECT * FROM **CONMUTADOR**

Llenar combo de conmutadores.

Recibe datos en la forma.

IF Opción = Consultar THEN

IF CampoSolicitudID <> "" THEN

SELECT * FROM **SOLICITUD** WHERE Solicitud_Id = CampoSolicitudID AND Estatus <= "Atendida"

IF CampoTipoSolicitud = "Act" THEN

SELECT * FROM **SOLICITUD_ACT** WHERE Solicitud_Id = CampoSolicitudID AND TipoSolicitudID = 1

SELECT * FROM **SOLICITUD_CAMB** WHERE Solicitud_Id = CampoSolicitudID AND TipoSolicitudID = 1

Desplegar datos de la Solicitud consultada.

Recibe datos en la forma.

IF Opcion2 = Modificar THEN

UPDATE **SOLICITUD**

IF CampoTipoSolicitud = "Act" THEN

UPDATE **SOLICITUD_ACT**

UPDATE **SOLICITUD_CAMB**

IF Opcion2 = Eliminar THEN

DELETE **SOLICITUD**

IF CampoTipoSolicitud = "Act" THEN

DELETE **SOLICITUD_ACT**

DELETE **SOLICITUD_CAMB**

IF Opción = Insertar THEN

CREATE **CLAVELD**

IF CampoTipoSolicitud = "Act" THEN

CREATE **SOLICITUD_ACT**

CREATE **SOLICITUD_CAMB**

Reporte de Llamadas por Ubicación

Recibe Clave de la ubicación y rango de fechas en la forma.

```
SELECT UbicacionID, Descripcion, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UBICACION, CONMUTADOR, EXTENSION, TIPO_LLAMADA, LLAMADA,
LLAMADA_TELMEX
WHERE UbicacionID = CampoUbicacion
AND UBICACION.ConmutadorID = CONMUTADOR.ConmutadorID
AND CONMUTADOR.ConmutadorID = EXTENSION.CONMUTADORID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY UbicacionID, Descripción, TipoLlamadaID, Descripción
```

Desplegar Resultados.

Reporte de Llamadas por Unidad de Negocio

Recibe Clave de la unidad de negocio y rango de fechas en la forma.

```
SELECT UnegocioID, Descripcion, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UNEGOCIO, DEPARTAMENTO, SOCIO, AREAGENERICA, EXTENSION,
TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE UNegocioID = CampoUnidadNegocio
AND UNEGOCIO.UnegocioID = DEPARTAMENTO*.UnegocioID
AND DEPARTAMENTO*.UnegocioID = SOCIO.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = SOCIO. DepartamentoID
AND DEPARTAMENTO*.UnegocioID = AREAGENERICA.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = AREAGENERICA.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = EXTENSION*.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = EXTENSION*. DepartamentoID
AND SOCIO.SocioID = EXTENSION*.SocioID
AND AREAGENERICA.AreaGenericaID = EXTENSION*. AreaGenericaID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY UnegocioID, Descripción, TipoLlamadaID, Descripción
```

Desplegar Resultados.

Reporte de Llamadas por Departamento

Recibe Clave de la unidad de negocio y departamento y rango de fechas en la forma.

```
SELECT DepartamentoID, Descripcion, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UNEGOCIO, DEPARTAMENTO, SOCIO, AREAGENERICA, EXTENSION,
TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE UNegocioID = CampoUnidadNegocio
AND UNEGOCIO.UnegocioID = DEPARTAMENTO*.UnegocioID
AND DEPARTAMENTO.DepartamentoID = CampoDepartamento
AND DEPARTAMENTO*.UnegocioID = SOCIO.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = SOCIO.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = AREAGENERICA.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = AREAGENERICA.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = EXTENSION*.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = EXTENSION*.DepartamentoID
AND SOCIO.SocioID = EXTENSION*.SocioID
AND AREAGENERICA.AreaGenericaID = EXTENSION*.AreaGenericaID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY DepartamentoID, Descripción, TipoLlamadaID, Descripción
```

Desplegar Resultados.

Reporte de Llamadas por Socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amaterno, Nombre, SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSION, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID = EXTENSION*.SocioID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY SocioID, Apaterno, Amaterno, Nombre, TipoLlamadaID, Descripción
```

Desplegar Resultados.

Reporte de Teléfonos más marcados por socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amaterno, Nombre, Teléfono SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSION, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID= EXTENSION*.SocioID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY SocioID, Apaterno, Amaterno, Nombre, TipoLlamadaID, Descripción, Teléfono
ORDER BY Cantidad DESC, Duración DESC, Importe DESC
```

Desplegar Resultados.

Reporte de Teléfonos no involucrados con la empresa por socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amaterno, Nombre, Teléfono SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSION, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID= EXTENSION*.SocioID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
AND NOT EXISTS (SELECT * FROM TEL_PERMITIDO WHERE LLAMADA.Telefono =
TEL_PERMITIDO.Telefono)
GROUP BY SocioID, Apaterno, Amaterno, Nombre, TipoLlamadaID, Descripción, Teléfono
```

Desplegar Resultados.

Reporte de Costos

Recibe rango de fechas en la forma.

```
SELECT UnegocioID, Descripcion, SUM(Importe)
FROM [UNEGOCIO], [DEPARTAMENTO], [SOCIO], [AREAGENERICA], [EXTENSION],
[TIPO_LLAMADA], [LLAMADA], [LLAMADA_TELMEX]
WHERE UNEGOCIO.UnegocioID = DEPARTAMENTO*.UnegocioID
AND DEPARTAMENTO*.UnegocioID = SOCIO.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = SOCIO. DepartamentoID
AND DEPARTAMENTO*.UnegocioID = AREAGENERICA.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = AREAGENERICA.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = EXTENSION*.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = EXTENSION*. DepartamentoID
AND SOCIO.SocioID= EXTENSION*.SocioID
AND AREAGENERICA.AreaGenericaID = EXTENSION*. AreaGenericaID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extension = LLAMADA.Extension
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY UnegocioID, Descripción, TipoLlamadaID, Descripción
```

Desplegar Resultados.

Directorio de Extensiones

Recibe nombre en la forma.

```
SELECT APaterno, AMaterno, Nombre, DEPARTAMENTO.Descripción,
AREAGENERICA.Descripción, ConmutadorID, Extension
FROM [DEPARTAMENTO], [SOCIO], [AREAGENERICA], [EXTENSION]
WHERE SOCIO.Apaterno = %Campo%
OR SOCIO.AMaterno = %Campo%
OR SOCIO.Nombre = %Campo%
OR DEPARTAMENTO.Descripción = %Campo%
OR AREAGENERICA.Descripción = %Campo%
AND SOCIO.SocioID = EXTENSION*.SocioID
AND DEPARTAMENTO.DepartamentoID = EXTENSION*.DepartamentoID
AND AREAGENERICA.AreaGenericaID = EXTENSION*.AreaGenericaID
```

Desplegar Resultados.

Reporte Gráfico de Llamadas por Ubicación

Recibe Clave de la ubicación y rango de fechas en la forma.

```
SELECT UbicacionID, Descripcion, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UBICACION, CONMUTADOR, EXTENSION, TIPO_LLAMADA, LLAMADA,
LLAMADA_TELMEX
WHERE UbicacionID = CampoUbicacion
AND UBICACION.ConmutadorID = CONMUTADOR.ConmutadorID
AND CONMUTADOR.ConmutadorID = EXTENSION.CONMUTADORID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY UbicacionID, Descripción, TipoLlamadaID, Descripción
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Reporte Gráfico de Llamadas por Unidad de Negocio

Recibe Clave de la unidad de negocio y rango de fechas en la forma.

```
SELECT UnegocioID, Descripcion, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UNEGOCIO, DEPARTAMENTO, SOCIO, AREAGENERICA, EXTENSION,
TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE UnegocioID = CampoUnidadNegocio
AND UNEGOCIO.UnegocioID = DEPARTAMENTO*.UnegocioID
AND DEPARTAMENTO*.UnegocioID = SOCIO.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = SOCIO.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = AREAGENERICA.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = AREAGENERICA.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = EXTENSION*.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = EXTENSION*.DepartamentoID
AND SOCIO.SocioID = EXTENSION*.SocioID
AND AREAGENERICA.AreaGenericaID = EXTENSION*.AreaGenericaID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSION.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSION.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY UnegocioID, Descripción, TipoLlamadaID, Descripción
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Reporte Gráfico de Llamadas por Departamento

Recibe Clave de la unidad de negocio y departamento y rango de fechas en la forma.

```
SELECT DepartamentoID, Descripción, SUM(Cantidad), SUM(Duración), SUM(Importe)
FROM UNEGOCIO, DEPARTAMENTO, SOCIO, AREAGENERICA, EXTENSIÓN,
TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE UNegocioID = CampoUnidadNegocio
AND UNEGOCIO.UnegocioID = DEPARTAMENTO*.UnegocioID
AND DEPARTAMENTO.DepartamentoID = CampoDepartamento
AND DEPARTAMENTO*.UnegocioID = SOCIO.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = SOCIO.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = AREAGENERICA.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = AREAGENERICA.DepartamentoID
AND DEPARTAMENTO*.UnegocioID = EXTENSIÓN*.UnegocioID
AND DEPARTAMENTO*.DepartamentoID = EXTENSIÓN*.DepartamentoID
AND SOCIO.SocioID = EXTENSIÓN*.SocioID
AND AREAGENERICA.AreaGenericaID = EXTENSIÓN*.AreaGenericaID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSIÓN.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSIÓN.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY DepartamentoID, Descripción, TipoLlamadaID, Descripción
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Reporte Gráfico de Llamadas por Socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amatemo, Nombre, SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSIÓN, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID = EXTENSIÓN*.SocioID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSIÓN.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSIÓN.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY SocioID, Apaterno, Amatemo, Nombre, TipoLlamadaID, Descripción
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Reporte Gráfico de Teléfonos más marcados por socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amatemo, Nombre, Teléfono SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSION, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID= EXTENSIÓN*.SocioID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSIÓN.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSIÓN.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
GROUP BY SocioID, Apaterno, Amatemo, Nombre, TipoLlamadaID, Descripción, Teléfono
ORDER BY Cantidad DESC, Duración DESC, Importe DESC
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Reporte Gráfico de Teléfonos no involucrados con la empresa por socio

Recibe Clave del socio y rango de fechas en la forma.

```
SELECT SocioID, Apaterno, Amatemo, Nombre, Teléfono SUM(Cantidad), SUM(Duración),
SUM(Importe)
FROM SOCIO, EXTENSION, TIPO_LLAMADA, LLAMADA, LLAMADA_TELMEX
WHERE SOCIO.SocioID = CampoSocio
AND SOCIO.SocioID= EXTENSIÓN*.SocioID
AND TIPO_LLAMADA.TipoLlamadaID = LLAMADA.TipoLlamadaID
AND EXTENSIÓN.ConmutadorID = LLAMADA.ConmutadorID
AND EXTENSIÓN.Extensión = LLAMADA.Extensión
AND LLAMADA.Fecha >= CampoFechaInicial
AND LLAMADA.Fecha <= CampoFechaFinal
AND LLAMADA.Llamada_TelmexID = LLAMADA_TELMEX.Llamada_TelmexID
AND NOT EXISTS (SELECT * FROM TEL_PERMITIDO WHERE LLAMADA.Telefono =
TEL_PERMITIDO.Telefono)
GROUP BY SocioID, Apaterno, Amatemo, Nombre, TipoLlamadaID, Descripción, Teléfono
```

Construir parámetros para Netcharts.

Enviar parámetros.

Desplegar Gráfica.

Actualización de Extensiones

SELECT * FROM **CONMUTADOR**

Llenar combo de conmutadores.

Permitir seleccionar un conmutador

SELECT * FROM **EXTENSION**

WHERE ConmutadorID = campoConmutadorID

Llenar listview de extensiones.

Permitir seleccionar extensiones o nuevo.

IF Opción = "Editar" THEN

Presentar pantalla de edición

SELECT * FROM **EXTENSION**

WHERE ConmutadorID = campoConmutadorID AND Extension = campoExtension

Llenar campos con información.

Permitir capturar en los campos.

UPDATE **EXTENSION**

IF Opción = "Agregar" THEN

Presentar pantalla de edición.

Llenar campos con información.

Permitir capturar en los campos.

INSERT **EXTENSION**

IF Opción = "Eliminar" THEN

Confirmar Eliminación

IF Respuesta = "S" THEN

DELETE **EXTENSION**

IF Opción = "Asignar" THEN

Presentar pantalla de asignación.

SELECT * FROM **TIPO_EXTENSION**

Llenar combo de tipo de extensión.

Permitir seleccionar tipo de extensión.

IF campoTipoExtensionID = 1 THEN

SELECT * FROM **SOCIO**, **EXTENSION_SOCIO**

WHERE SocioID = SocioID AND TipoExtensionID = 1

Llenar combo de socios.

Permitir seleccionar socio.

CREATE **EXTENSION_SOCIO**

IF campoTipoExtensionID = 2 THEN

SELECT * FROM **UNEGOCIO**, **DEPARTAMENTO**, **UNEG DEPARTAMENTO**,
EXTENSION DEPARTAMENTO

WHERE UNegocioID = UNegocioID AND DepartamentoID = DepartamentoID AND
TipoExtensionID = 2

Llenar combo de Unidad de Negocio y de Departamentos.

Permitir seleccionar Unidad de Negocio y Departamento.

CREATE **EXTENSION DEPARTAMENTO**

IF campoTipoExtensionID = 3 THEN

```
SELECT * FROM AREA_GENERICA, EXTENSION_AREAGENERICA
WHERE AreaGenericaID = AreaGenericaID AND TipoExtensionID = 3
Llenar combo de Areas Genericas.
Permitir seleccionar socio.
CREATE EXTENSION_AREAGENERICA
```

IF Opción = "Permisos" THEN

```
Presentar pantalla de permisos
SELECT * FROM TIPO_LLAMADA
Llenar combo de servicios disponible
Permitir seleccionar permisos.
CREATE EXTENSION_TIPOLL
```

Conciliación

```
Cargar archivo a tabla de paso.
Desglosar registro y CREATE LLAMADA_TELMEX
UPDATE LLAMADA
SET LlamadaTelmexID = LlamadaTelmexID
FROM LLAMADA, LLAMADA_TELMEX
WHERE ISNULL(LLAMADA.Llamada_TelmexID,-1) = -1
AND LLAMADA.Telefono = B.Destino
AND LLAMADA.Fecha >= DATEADD(mi,-1,DATEADD(ss, LLAMADA.Duracion * -1,B.Fecha))
AND LLAMADA.Fecha <= DATEADD(mi,1,DATEADD(ss, LLAMADA.Duracion * -1,B.Fecha))
AND (LLAMADA.Duracion / 60) >= B.Duracion - 1
AND (LLAMADA.Duracion / 60) <= B.Duracion + 1
Revisar si existen registros por conciliar.
Conciliar manualmente.
```

Administración de Solicitudes

```
SELECT * FROM SOLICITUD
Llenar lista y permitir seleccionar solicitud.
```

IF Opción = "Detalle" THEN

```
Presentar pantalla de detalle de solicitud.
Mostrar detalle de solicitud.
```

IF Opción = "Autorizar" THEN

```
UPDATE SOLICITUD
SET Estatus = 2
Enviar mail.
```

IF Opción = "Cancelar" THEN

```
UPDATE SOLICITUD
SET Estatus = 3
Enviar mail.
```

IF Opción = "Procesar" THEN

Conectarse al conmutador.
Ejecutar comandos para cumplir con el detalle de la solicitud.
Mostrar reporte de procesamiento.
Enviar mail.
Permitir asignar la extensión a un propietario.

Actualización de Información en el Conmutador.

Mostrar lista de conmutadores.
Conectarse al conmutador seleccionado.
Ejecutar comando para extraer extensiones disponibles.
Llenar combo de extensiones disponibles.

IF Opción = "Consultar" THEN

Ejecutar comando para consulta de información de extensión.
Llenar campos con información obtenida.

IF Opción = "EQU's Libres" THEN

Ejecutar comando para consulta de EQU's libres.
Llenar lista.

IF Opción = "Ext. Libres" THEN

Ejecutar comando para consulta de Extensiones libres.
Llenar lista.

IF Opción = "Nombre" THEN

Ejecutar comando para consulta de Nombre de la extensión.
Llenar campos.
Ejecutar comando para actualizar nombre.

IF Opción = "Código" THEN

Ejecutar comando para consulta de código de la extensión.
Llenar campos.
Ejecutar comando para actualizar código.

IF Opción = "Activar" THEN

Ejecutar comando para consulta de EQU's libres.
Llenar lista.
Ejecutar comando para consulta de Extensiones libres.
Llenar lista.
Permitir seleccionar campos.
Ejecutar comando para activar la extensión.

IF Opción = "Desactivar" THEN

Confirmar desactivación.
Ejecutar comando para desactivar la extensión.

Desconectarse del conmutador.

5.8. Prototipos.

5.8.1. Recopilación de Llamadas.

5.8.1.1. Ambiente de Pruebas.

El primer prototipo fue probado en el siguiente ambiente.

HP NetServer LH3

Windows NT 4.0

Cliente de SQL Server

Conmutador Ericsson MD-110 del Centro de Apoyo de HEB México.

5.8.1.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Almacenar la información de las llamadas externas.
- Enviar un correo electrónico al administrador.
- Depurar los archivos de almacenamiento local.

5.8.1.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información y a socios del mismo departamento. Las observaciones realizadas fueron las siguientes.

- El prototipo cumple con las expectativas y funciones que se tenían contempladas.
- El prototipo necesita ser implementado y continuar durante un tiempo en estado de pruebas para verificar su estabilidad y confiabilidad.

5.8.2. Perfiles y Usuarios.

5.8.2.1. Ambiente de Pruebas.

El segundo prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM

Windows 95

Microsoft Internet Explorer 5.0

5.8.2.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Actualizar las opciones del sistema.
- Actualizar la información de los usuarios del sistema y sus privilegios.

5.8.2.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información y a socios del mismo departamento. Las observaciones realizadas fueron las siguientes:

- El prototipo cumple con las expectativas y funciones que se tenían contempladas.
- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Validar la integridad referencial.

5.8.3. Utilerías.

5.8.3.1. Ambiente de Pruebas.

El tercer prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.3.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Actualizar la información de conmutadores, tipos de extensión, tipos de llamada, claves de larga distancia, tipos de solicitudes, áreas genéricas y teléfonos permitidos.

5.8.3.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información y a socios del mismo departamento. Las observaciones realizadas fueron las siguientes:

- El prototipo cumple con las expectativas y funciones que se tenían contempladas.
- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Sin embargo sería conveniente agregar a los actualizadores la forma de consultar por un listado con los datos almacenados en las tablas.

5.8.4. Depuración de Base de Datos.

5.8.4.1. Ambiente de Pruebas.

El cuarto prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.4.2. Funciones a Evaluar.

Este prototipo incluye las siguiente función.

- Depurar base de datos.

5.8.4.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información y a socios del mismo departamento. Las observaciones realizadas fueron las siguientes:

- El prototipo cumple con las expectativas y funciones que se tenían contempladas.
- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Seria importante validar bien las fechas y los rangos de las mismas.

5.8.5. Generación de Solicitudes de Servicio.

5.8.5.1. Ambiente de Pruebas.

El quinto prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.5.2. Funciones a Evaluar.

Este prototipo incluye la siguiente función.

- Actualizar solicitudes de activación y cambios de extensiones.
- Enviar un correo electrónico al socio.

5.8.5.3. Evaluación.

El prototipo fue presentado al jefe de comunicaciones y mantenimiento y al analista de comunicaciones. Las observaciones realizadas fueron las siguientes:

- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Realizar pruebas para que la validación de la información este comprobada.

5.8.6. Reportes.

5.8.6.1. Ambiente de Pruebas.

El sexto prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.6.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Consultar las llamadas realizadas por las unidades de negocio.
- Graficar la cantidad de llamadas realizadas por las unidades de negocio.
- Consultar las cantidades, tiempos y costos de las llamadas realizadas por los departamentos.
- Graficar la cantidad de llamadas realizadas por los departamentos.
- Consultar las cantidades, tiempos y costos de las llamadas realizadas por los socios.
- Graficar la cantidad de llamadas realizadas por los socios.
- Consultar los teléfonos más marcados por los socios, indicando los teléfonos que no se involucran con la empresa.
- Graficar las cantidades y tiempos de las llamadas a los teléfonos más marcados por los socios.
- Graficar las cantidades y tiempos de los teléfonos marcados por los socios y que no se involucran con la empresa.
- Consultar el directorio de extensiones de la empresa.

5.8.6.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información, al subdirector de contraloría, al supervisor de cuentas por pagar y a socios del departamento de sistemas de información. Las observaciones realizadas fueron las siguientes:

- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Las fechas que aparecen por omisión en los campos que no dependan de la configuración de la computadora cliente.
- Crear un reporte que permita identificar los máximos por unidades de negocio.
- Poder visualizar los reportes tanto en gráficos de barras como en de pastel.
- Contar con un desglose del reporte de llamadas por ubicación.

5.8.7. Conciliación.

5.8.7.1. Ambiente de Pruebas.

El séptimo prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.7.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Actualizar los costos de las llamadas mediante conciliación de información.
- Consultar la forma en que se conciliaron las llamadas.

5.8.7.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información, al subdirector de contraloría, al supervisor de cuentas por pagar y a socios del departamento de sistemas de información. Las observaciones realizadas fueron las siguientes:

- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- El prototipo cumple con las funciones a necesarias.
- Mostrar la cantidad registros pendientes por conciliar.
- Filtrar los registros pendientes por conciliar mediante fechas.

5.8.8. Administración del Conmutador.

5.8.8.1. Ambiente de Pruebas.

El octavo prototipo fue probado en el siguiente ambiente:

PC Pentium con 46MB RAM
Windows 95
Microsoft Internet Explorer 5.0

5.8.8.2. Funciones a Evaluar.

Este prototipo incluye las siguientes funciones.

- Actualizar la relación socio y extensión
- Actualizar la relación entre las extensiones y los servicios de telefonía.
- Enviar un correo electrónico al socio.
- Activar extensiones análogas.
- Activar extensiones digitales.
- Desactivar extensiones digitales.
- Desactivar extensiones análogas.

5.8.8.3. Evaluación.

El prototipo fue presentado al gerente de sistemas de información, al jefe del departamento de comunicaciones y al analista de comunicaciones. Las observaciones realizadas fueron las siguientes:

- El prototipo cubre los estándares de desarrollo de sistemas de la empresa.
- Permitir imprimir el reporte de procesamiento de solicitudes.
- Verificar si es posible agilizar la carga de las extensiones disponibles.

Con los resultados obtenidos de la evaluación de los prototipos expuestos anteriormente, se da por concluida la fase de diseño o JAD ya que se cuenta con la suficiente información para continuar con la etapa de construcción.

6. CONSTRUCCIÓN

En esta fase del desarrollo rápido de aplicaciones (RAD) se desarrollan físicamente los resultados del diseño o JAD. A continuación se exponen los estandares utilizados para construir la aplicación, los procesos de mayor complejidad, algunos problemas y soluciones y un resumen estadístico sobre la construcción de la aplicación.

6.1. Estándares.

A continuación se explicarán los estándares de programación utilizados por el departamento de sistemas de NEP México para la programación en Intranet, Visual Basic y SQL Server.

6.1.1. Estándares de Programación en Intranet.

6.1.1.1. Convención para los nombres de archivos.

Los archivos físicos o páginas que se depositen sobre el servidor de Intranet podrán llevar las siguientes extensiones:

.htm o *.html*: Conviene definir desde el principio del proyecto el tipo de extensión que se estará manejando, ya sea *.htm* o *.html* para páginas cuyo contenido es texto o código de html.

.asp: Esta extensión se utilizará para las Active Server Pages, cuyo contenido puede ser código de HTML, VBScript, o VBScript de ASP's.

.gif o *.jpg*: Para los archivos que se inserten como imágenes en alguna de las páginas.

Para las páginas que contienen Frames o que dividen a una página en 2 o más Frames, se ha convenido utilizar el prefijo *fra* antes del nombre del Archivo. Por ejemplo:

Una página que contiene a 2 frames y que genera un reporte, lleva como primer frame un encabezado y en el otro el cuerpo del reporte en donde aparecerán los resultados, y suponiendo que el reporte es de un listado de artículos, pues el nombre conveniente sería *Fra_VentasDiarias.htm*.

Para los encabezados del reporte, se utilizará el prefijo *enca*. Por Ejemplo: *Enca_ventasdiarias.asp*.

El cuerpo del reporte podrá ser un archivo *.asp* o *.htm*, el cual podrá contener el logotipo de HEB al centro y el nombre de la aplicación, antes de que se dispare el reporte sobre este archivo.

Los reportes deberán llevar el mismo nombre del encabezado, a diferencia del prefijo, el cual será: *Rpt_nombre del reporte.asp*.
Ejemplo: *Rpt_VentasDiarias.asp*.

6.1.1.2. Encabezados

Todas las páginas con extensión *.asp* deberán llevar el siguiente encabezado en la parte superior de la página, antes de cualquier otra documentación.

```
<%@ Language = "VBScript" %>
<% Response.Buffer = true %>
```

Nota: Si se utilizara la sentencia *include file* deberá ir inmediatamente después de estas 2 líneas.

6.1.1.3. Conexión a una Base de Datos

Sintaxis:

```
<%
    usuario = "usuario"
    password = "password"
```



```

Set cnn = server.createobject("ADODB.connection")
Strcn = "DSN=NombredelDSN; UID=" & usuario & "; PWD=" & Password &
"DATABASE=BASEDATOS"
Cnn.open strcn
%>

```

Donde:

Usuario: Es el que se define previamente en SQL Server con los permisos necesarios para tener acceso a tablas y stored procedures que se encuentren en la base de datos.

Password: Es el password que se le asigna al usuario.

NombredelDSN: El Nombre del DSN deberá ser especificado en el Servidor de Intranet. Éi comúnmente usado para las aplicaciones desarrolladas en la Intranet, recibe el nombre de: *Intranet*. El cual se encuentra dado de alta en el servidor de Intranet.

6.1.1.4. Nombre y configuración del Microsoft SQL Server ODBC Driver. (DSN)

Data Source Name: Intranet.

Data Source Description: Se conecta al Servidor de Data Warehouse.

Server: DB_DWH

Database: Corporativa

Language: (Default)

Translation Option: Automatic

Log Long Running Queries: No

Log Driver Statistics: No

Use Integrated Security: No

Use Regional Settings: No

Prepared Statements Option: Drop temporary procedures on disconnect

Use Failover Server: No

Use ANSI Quoted Identifiers: Yes

Use ANSI Null, Paddings and Warnings: Yes.

6.1.1.5. Manejo de fechas con APS's.

Las fechas que se capturen en los reportes como período, deberán ser ingresadas por el usuario con el formato: dd/mm/aaaa. Validando que estas sean capturadas en el orden correcto, y que no acepte fechas inválidas o inexistentes.

Las Fechas de los reportes deberán ser desplegadas con el siguiente formato:

Del: dd/mm/aaaa. Al: dd/mm/aaaa.

6.1.1.6. Ejemplo del código de una página que se divide en 2 frames.

```
<%@ LANGUAGE="VBScript" %>
<% response.buffer = True %>

<!-- #include file = "secure.asp" -->
<!-- #include file = "dsn.asp" -->
<!-- #include file = "adovbs.inc" -->

<HTML>
<HEAD>
<TITLE>[Sistema para Pago Variable de Panadería]</TITLE>
</HEAD>

<frameset rows="35%,75%" border="0">
  <frame src="Enca_ProduccionMerma.asp" name="encabezado" noresize
scrolling="no">
  <frame src="Cuerpo.asp" name="cuerpo" noresize scrolling="auto">
</frameset>

</HTML>
```

Visual Basic Script.

Las subrutinas de Visual Basic Script deberán estar definidas entre los siguientes tags, y de preferencia al final del código de Html y de Asp's:

```
<Script Language = "VBScript">
<!--
    Sub NombreSubrutina()

        End Sub
-->
</Script>
```

6.1.1.7. Comentarios y Encabezados.

Todos los programas deberán llevar la siguiente documentación con el fin de tener identificados a los responsables de su construcción, así como las modificaciones que puedan afectar a la aplicación. Los comentarios de ASP's y HTML se delimitan con los siguientes tags:

```
<!-- - Comentarios - -->
```

Los comentarios de VBScript se delimitan con un apostrofe (') al inicio de este. A continuación se muestra un ejemplo del encabezado para la documentación de programas.

```
'***** Nombre del Autor: Autor.          *****
'***** Nombre del Programa: Programa.      *****
'***** Fecha Creación: Fecha.              *****
'***** Modifico: Nombre.                   *****
'***** Fecha Modificación: Fecha.          *****
'***** Descripción: Descripción.          *****
'***** Dependencia: Programa Padre        *****
```

6.1.2. Estándares de Programación en Visual Basic.

Nomenclatura de variables.

Los nombres de las variables pueden estar en español o en inglés. La estructura de los nombres de las variables es la siguiente:

{Prefijo}{Cuerpo}[Calificadores][Sufijo]

Donde prefijo describe el uso y la visibilidad o "scope" de la variable y está constituido por una o más letras minúsculas:

El cuerpo es el nombre de la variable, es decir, lo que la identifica y la hace diferente a las demás. Debe ser lo suficientemente descriptivo para que el programador tenga una idea de la función que la variable desempeña. La primer letra del cuerpo debe ser mayúscula y las demás minúsculas.

Los calificadores denotan derivaciones estándar de una variable base. Por ejemplo, si se tiene una variable sName y se quiere declarar algunas variables relacionadas con esta pero que almacenen algunos de los nombres más significativos de una serie, se podrían utilizar sNameFirst, sNameLast, sNameCur, sNamePrev, sNameNext, etc.

La siguiente tabla muestra los prefijos estándar a utilizar en el desarrollo.

<i>Prefijo</i>	<i>Descripción</i>
i	Integer
Fn	Function
b	Boolean
by	Byte
l	Long
f	Float
d	Double
Cur	Currency
s	String
Vnt	Variant

Tabla 6.1. Prefijos para variables en VB.

Los prefijos especiales definen alguna propiedad extra de la variable a la cual están asignados. Es decir, no identifican el tipo sino la visibilidad, el dimensionamiento u alguna otra particularidad de dicha variable. Estos prefijos se deben de emplear antes del prefijo de tipo. La siguiente tabla muestra los prefijos especiales.

<i>Prefijo Especial</i>	<i>Descripción</i>
a	Array
g	Global
m	Variable local a un módulo o forma
r	Variable pasada por referencia
t	Tipo definido por el usuario
p	Pointer

Tabla 6.2. Prefijos especiales para variables en VB.

La siguiente tabla muestra los calificadores estándar.

<i>Calificador</i>		<i>Descripción</i>
First	Pri	Primer elemento de un conjunto
Last	Ult	Último elemento de un conjunto
Next	Sig	Siguiente elemento de un conjunto
Prev	Ant	Elemento siguiente de un conjunto
Cur	Act	Elemento Actual de un conjunto
Mm	Mm	Valor mínimo de un conjunto
Max	Max	Valor máximo de un conjunto
Tmp	Tmp	Variable cuyo uso está restringido a un área mínima de código
Src	Fte	Fuente. Usado en comparaciones y en operaciones de transferencia
Dst	Dst	Destino. Usado en comparaciones y operaciones de transferencia

Tabla 6.3. Calificadores para variables en VB.

Nomenclatura de controles.

La nomenclatura utilizada en los controles tiene la misma estructura de las variables a excepción de uso de sufijos, los cuales no son válidos en los nombres de los controles. La siguiente tabla muestra los diversos controles y sus prefijos.

Prefijo	Control
Ani	Animated Button
Bed	Pen Bedit
Cbo	ComboBox
Chk	CheckBox
Clp	Picture Clip
Cmd	Command Button
Com	Communications
Ctr	Control
Dir	Dir ListBox
Dlg	Common Dialog
Drv	Drive ListBox
Ds	ODBC Dynaset
Fil	File ListBox
Fol	Folder Control
Frm	Form
Fra	Frame
Gau	Gauge
Grd	Grid
Hed	Pen Hedit
Hsb	Horizontal ScrollBar
Img	Image
Ink	Pen Ink
Key	Keyboard Key Status
Lbl	Label
Lin	Line
Lst	ListBox
Lvw	ListView
Mpm	MAPI Message
Mps	MAPI Session
Mci	MCI
Mnu	Menu
Opt	Option Button
Ole	Ole Client
Pic	Picture
Pnl	3D Panel
Shp	Shape
Spn	Spin Control

<i>Prefijo</i>	<i>Control</i>
Tab	Tabs Control
Txt	TextBox
Tmr	Timer
Vsb	Vertical ScrollBar

Tabla 6.4. Prefijos para controles de VB.

Encabezados.

Todos los módulos y formas deben de contar con un encabezado en el que se describa de una manera breve y clara la funcionabilidad del mismo. El formato es el siguiente:

'Titulo: *Titulo*
 'Modulo: *Archivo*
 'Versión: *Versión*
 'Fecha: *Fecha*
 'Autor: *Autor*

Los procedimientos y funciones más importantes de la aplicación deben tener el siguiente encabezado:

'Función: *Función*
 'Descripción: *Descripción*
 'Usar: *Forma de uso*

6.1.3. Estándares de Programación en SQL Server.

6.1.3.1. Identificadores, nombres de objetos, bases de datos y variables.

Dispositivos

Al crear un dispositivo es necesario especificar el nombre lógico y físico del mismo. Para el nombre lógico usar el siguiente formato:

<nombre_bd>Dat (Dispositivo para la base de datos)

<nombre_bd>Log (Dispositivo para el Log)

En donde *<nombre_bd>* es el nombre de la base de datos que va a contener este dispositivo. Cuando se necesita tener más de un dispositivo para la base de datos (o Log) se especifica el número de dispositivo que es; se empiezan a numerar a partir del segundo.

Para el nombre físico usar el siguiente formato:

<nombre_bd>Dat.Dat (Dispositivo para la base de datos)

<nombre_bd>Log.Log (Dispositivo para el Log)

En donde <nombre_bd> es el nombre de la base de datos. Cuando se necesita tener más de un dispositivo para la base de datos (o Log) se especifica el número de dispositivo que es; se empiezan a numerar a partir del segundo. El nombre de la base de datos más el número de dispositivos deben ocupar máximo 8 caracteres.

Segmentos

Utilizar el siguiente formato:

Sg<nombre_segmento>

Ejemplo:

SgNomina

Bases de datos

Nombrar las bases de datos usando identificadores representativos, sin incluir prefijos ni sufijos y siempre en singular.

Ejemplo:

Nomina

Cheque

Tesoreria

Tablas

Usar identificadores representativos y en forma singular. Evitar abreviaciones en los nombres. La longitud del nombre para las tablas debe ser menor o igual a 30 caracteres y en el caso de tablas temporales menor o igual a 12.

Cuando se tengan tablas de paso, es decir, aquellas que no son temporales pero que se usan para realizar algún proceso y después de cierto tiempo se eliminan, su nombre se define utilizando el símbolo _ (guión bajo) como primer carácter. Las tablas temporales llevan como primer carácter el símbolo #.

Columnas

Los nombres de las columnas de una tabla deben seguir las reglas de los identificadores. Estos deben ser lo más representativos posibles, en forma singular y evitando el uso de abreviaciones.

Stored Procedures

Utilizar el siguiente formato:

sp<nombre_procedimiento>

Los nombres a utilizar para stored procedures básicos son los siguientes:

spIns<nombre_tabla> (Inserta un registro en la tabla)

spDel<nombre_tabla> (Elimina un registro de la tabla)

spUpd<nombre_tabla> (Actualiza un registro de la tabla)

spSel<nombre_tabla> (Selección no completa de una tabla. No incluye columnas y/o lleva un *where*)

spAll<nombre_tabla> (Selección completa de una tabla)

Triggers

Utilizar el siguiente formato:

tr<operación(es)><nombre_tabla>

Variables y parámetros

Utilizar nombres representativos siguiendo las reglas de los identificadores. Si la variable representa una columna de una tabla, utilizar ese mismo nombre.

Ejemplo:

@Clienteld

@PersonaDireccion

Identificadores

Los identificadores son utilizados para nombrar los objetos de una base de datos, tablas, vistas, stored procedures, triggers, rules, defaults, etc.

Sintaxis de los identificadores usados en SQL Server.

En SQL Server, los identificadores tienen una longitud de 1 a 30 caracteres. El primer carácter debe ser una letra o el símbolo @ ó _. Después de este se pueden incluir letras, dígitos, o los símbolos #, \$, _.

Los nombres de las tablas temporales inician con el símbolo #. El nombre no debe ser mayor a 13 caracteres incluyendo el #, ya que SQL Server les agrega un sufijo numérico para manejo interno. Ningún otro nombre de objeto debe llevar este símbolo como primer carácter.

Los nombres de las variables locales inician con el símbolo @.

6.1.3.2. Estándares de desarrollo.

Se deben seguir las mismas reglas descritas anteriormente para los identificadores usados en SQL Server.

Deberá hacerse uso de mayúsculas y minúsculas para denotar los nombres de los objetos. No se deberán usar acentos ni ñ's en los identificadores.

Encabezados y Comentarios

A continuación se muestra un ejemplo de encabezado para stored procedure y trigger:

```

/*****/
/*  Nombre físico: Nombre del Archivo          */
/*  Autor: Autor                                */
/*  Fecha: Fecha                                */
/*  Proyecto: Nombre Proyecto                  */
/*  Descripción: Descripción                    */
/*  Fecha Modificación: Fecha                  */
/*  Modificación: Descripción                  */
/*  Autor Modificación: Autor                  */
/*****/

```

El encabezado debe ir después de la palabra reservada AS. El log de modificaciones en el encabezado contendrá las tres últimas modificaciones.

Para los stored procedures generales (spAll, spDel, spIns, spSel, spUpd) se utilizará el siguiente encabezado.

```

/*****/
/*  Nombre físico: Nombre del Archivo          */
/*  Autor: Autor                                */
/*  Fecha: Fecha                                */
/*  Fecha Modificación: Fecha                  */
/*  Proyecto: Nombre Proyecto                  */
/*****/

```

En cuanto a comentarios se refiere, no se cuenta con ningún estándar, sin embargo se debe intentar introducir comentarios lo mas que se pueda, sobre todo si el proceso es complejo.

Consideraciones al programar.

- Las instrucciones o estatutos de Transact SQL deberán escribirse todos con letras mayúsculas. Excepto los tipos de datos de SQL Server.
- Hacer uso de indentación apropiada en cada línea de código.

6.2. Construcción del backend.

A continuación se incluyen las rutinas de SQL Server para la creación de las tablas que componen el modelo de datos del sistema.

```

/***** Object: Table dbo.AreaGenerica    Script Date:
05/05/2000 7:11:33 PM *****/
CREATE TABLE dbo.AreaGenerica (
    AreaGenericaID smallint IDENTITY (1, 1)
NOT NULL,
    UNegocioid varchar (5) NULL,
    Departamentoid varchar (10) NULL,
    Descripcion varchar (30) NULL
)
GO

```

```

/***** Object: Table dbo.ClaveLD        Script Date:
05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.ClaveLD (
    ClaveLDId int IDENTITY (1, 1) NOT NULL,
    TipoLlamadaid smallint NOT NULL,
    Clave varchar (5) NULL,
    Descripcion varchar (30) NULL
)
GO

```

```

/***** Object: Table dbo.Conmutador    Script Date:
05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Conmutador (
    ConmutadorId smallint NOT NULL,
    Descripcion varchar (50) NULL,
    DireccionIP varchar (15) NULL
)
GO

```

```

/***** Object: Table dbo.Extension    Script Date:
05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Extension (
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    TipoExtensionId smallint NOT NULL,
    CTUA smallint NULL,
    Signal varchar (4) NULL,
    Puerto varchar (12) NULL,
    Rack varchar (2) NULL,
    Regleta smallint NULL,
    Fila smallint NULL,
    Par smallint NULL,
    Raiser4 varchar (8) NULL,
    Raiser varchar (8) NULL,
    CRaiser varchar (2) NULL,
    LFisico varchar (7) NULL,
    TipoTel varchar (5) NULL,
    NombreDisplay varchar (20) NULL
)
GO

```

```

/***** Object: Table dbo.Extension_AreaGenerica
Script Date: 05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Extension_AreaGenerica (
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    AreaGenericaID smallint NULL
)
GO

```

```

/***** Object: Table dbo.Extension_Departamento
Script Date: 05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Extension_Departamento (
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    Departamentoid varchar (10) NULL,
    UNegocioid varchar (5) NULL
)
GO

```

```

/***** Object: Table dbo.Extension_Socio    Script
Date: 05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Extension_Socio (
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    Socioid varchar (11) NULL
)
GO

```

```

/***** Object: Table dbo.Extension_TipoLL    Script
Date: 05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Extension_TipoLL (
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    TipoLlamadaid smallint NOT NULL
)
GO

```

```

/***** Object: Table dbo.Llamada    Script Date:
05/05/2000 7:11:34 PM *****/
CREATE TABLE dbo.Llamada (
    Fecha datetime NOT NULL,
    Llamadaid int IDENTITY (1, 1) NOT NULL,
    ConmutadorId smallint NOT NULL,
    Extension smallint NOT NULL,
    Llamada_TelmexId int NULL,
    TipoLlamadaid smallint NULL,
    ClaveLDId int NULL,
    Duracion decimal(10, 2) NULL,
    Codigo_Autorizacion smallint NULL,
    Telefono varchar (20) NULL
)
GO

```

```

/***** Object: Table dbo.Llamada_Telmex    Script
Date: 05/05/2000 7:11:35 PM *****/
CREATE TABLE dbo.Llamada_Telmex (
    Llamada_TelmexId int IDENTITY (1, 1) NOT
NULL,
    Fecha datetime NULL,
    Origen varchar (20) NULL,
    Destino varchar (20) NULL,
    Duracion decimal(10, 2) NULL,
    Importe decimal(15, 2) NULL,
    Pob_Destino varchar (30) NULL,
    Fecha_Factura smalldatetime NULL
)
GO

```

```

/***** Object: Table dbo.Opc_sistema    Script Date:
05/05/2000 7:11:35 PM *****/

```

```

CREATE TABLE dbo.Opc_sistema (
    OpcionID int NOT NULL ,
    Descripcion varchar (30) NULL ,
    TipoOpcion smallint NULL ,
    Agrupacion smallint NULL ,
    Programa varchar (50) NULL
)
GO

/***** Object: Table dbo.Perfil      Script Date:
05/05/2000 7:11:35 PM *****/
CREATE TABLE dbo.Perfil (
    PerfilID int NOT NULL ,
    Descripcion varchar (30) NULL
)
GO

/***** Object: Table dbo.Perfil_Opcion  Script Date:
05/05/2000 7:11:35 PM *****/
CREATE TABLE dbo.Perfil_Opcion (
    PerfilID int NOT NULL ,
    OpcionID int NOT NULL
)
GO

/***** Object: Table dbo.Solicitud      Script Date:
05/05/2000 7:11:36 PM *****/
CREATE TABLE dbo.Solicitud (
    Solicitud_Id int IDENTITY (1, 1) NOT NULL ,
    Tipo_SolicitudID smallint NOT NULL ,
    ConmutadorID smallint NULL ,
    Extension smallint NULL ,
    Socioid varchar (11) NULL ,
    Num_Nodo varchar (15) NULL ,
    Centro_Costo varchar (30) NULL ,
    Motivo varchar (255) NULL ,
    Fecha datetime NULL ,
    Estatus smallint NULL
)
GO

/***** Object: Table dbo.Solicitud_Act  Script Date:
05/05/2000 7:11:36 PM *****/
CREATE TABLE dbo.Solicitud_Act (
    Solicitud_Id int NOT NULL ,
    Tipo_SolicitudID smallint NOT NULL ,
    TipoLinea smallint NULL ,
    TipoNumero smallint NULL ,
    Display varchar (20) NULL ,
    CodigoAut numeric(5, 0) NULL ,
    Fecha_Inicio datetime NULL
)
GO

/***** Object: Table dbo.Solicitud_Camb  Script Date:
05/05/2000 7:11:36 PM *****/
CREATE TABLE dbo.Solicitud_Camb (
    Solicitud_Id int NOT NULL ,

```

```

    Tipo_SolicitudID smallint NOT NULL ,
    CodigoAut numeric(5, 0) NULL ,
    CodigoAnterior numeric(5, 0) NULL ,
    DisplayNuevo varchar (20) NULL ,
    NuevoAparato tinyint NULL ,
    NuevoNodo varchar (15) NULL ,
    NumDirectoAnt numeric(15, 0) NULL ,
    Ext_Directo numeric(15, 0) NULL ,
    BloqueoLD numeric(4, 0) NULL
)
GO

/***** Object: Table dbo.Tel_Permitido  Script Date:
05/05/2000 7:11:36 PM *****/
CREATE TABLE dbo.Tel_Permitido (
    Telefonoid int IDENTITY (1, 1) NOT NULL ,
    Telefono varchar (20) NULL ,
    Descripcion varchar (50) NULL
)
GO

/***** Object: Table dbo.TipoExtension  Script Date:
05/05/2000 7:11:36 PM *****/
CREATE TABLE dbo.TipoExtension (
    TipoExtensionID smallint NOT NULL ,
    Descripcion varchar (30) NULL
)
GO

/***** Object: Table dbo.Tipo_Llamada  Script Date:
05/05/2000 7:11:37 PM *****/
CREATE TABLE dbo.Tipo_Llamada (
    TipoLlamadaID smallint NOT NULL ,
    Prefijo varchar (10) NULL ,
    Descripcion varchar (30) NULL ,
    TipoExtraeLada tinyint NOT NULL
)
GO

/***** Object: Table dbo.Tipo_Solicitud  Script Date:
05/05/2000 7:11:37 PM *****/
CREATE TABLE dbo.Tipo_Solicitud (
    Tipo_SolicitudID smallint NOT NULL ,
    Descripcion varchar (30) NULL
)
GO

/***** Object: Table dbo.Usuario      Script Date:
05/05/2000 7:11:37 PM *****/
CREATE TABLE dbo.Usuario (
    UsuarioID varchar (10) NOT NULL ,
    PerfilID int NULL ,
    Password varchar (20) NULL ,
    Socioid varchar (11) NULL
)
GO

```

6.3. Procesos de mayor complejidad.

6.3.1. Actualización de Información en el Conmutador.

El proceso de actualización de información se compone de varias formas, sin embargo, no es necesario mostrar todas ellas para observar las operaciones básicas del proceso.

Módulo vt100

Este módulo cuenta con funciones que permiten identificar y procesar la información que es enviada por el conmutador.

Option Explicit

```
Private Declare Function IsCharAlpha Lib "user32"
Alias "IsCharAlphaA" (ByVal cChar As Byte) As Long
```

```
Private Const GO_IAC1 = 6
Private bInEscape As Boolean
Private sEscString As String
Public sRecibido As String
```

```
Public Function FnProcess_Char(CH As Byte)
```

```
    If (bInEscape) Then
```

```
        Call EscapeProcess(CH)
```

```
    Else
```

```
        Select Case CH
```

```
            Case 7
```

```
                Beep
```

```
            Case 8
```

```
                sRecibido = Mid(sRecibido, 1, Len(sRecibido) -
```

```
1)
```

```
                Case 9
```

```
                    sRecibido = sRecibido + " "
```

```
                Case 10, 11, 12
```

```
                    sRecibido = sRecibido + Chr(10)
```

```
                Case 13
```

```
                Case 27
```

```
                    bInEscape = True
```

```
                    sEscString = ""
```

```
                Case 255
```

```
                    FnProcess_Char = GO_IAC1
```

```
                Case Else
```

```
                    sRecibido = sRecibido & Chr$(CH)
```

```
                End Select
```

```
            End If
```

```
        End Function
```

```
        Private Sub EscapeProcess(CH As Byte)
```

```
            Dim c As String
```

```
            Dim yDiff As Integer
```

```
            Dim xDiff As Integer
```

```
            c = Chr$(CH)
```

```
            If sEscString = "" Then
```

```
                Select Case c
```

```
                    Case Chr$(8)
```

```
                        sRecibido = Mid(sRecibido, 1, Len(sRecibido))
```

```
                        bInEscape = False
```

```
                    Case "7"
```

```
                        bInEscape = False
```

```
                    Case "8"
```

```
                        bInEscape = False
```

```
                    Case "D"
```

```
                        sRecibido = sRecibido & Chr(10)
```

```
                        bInEscape = False
```

```
                    Case "E"
```

```
                        sRecibido = sRecibido & Chr(10)
```

```
                        bInEscape = False
```

```
                    Case "H"
```

```
                        sRecibido = sRecibido & " "
```

```
                        bInEscape = False
```

```
                    Case "I"
```

```
                        bInEscape = False
```

```
                    Case "Z"
```

```
                        bInEscape = False
```

```
                    Case Else
```

```
                        bInEscape = False
```

```
                        Exit Sub
```

```
                End Select
```

```
            End If
```

```
            sEscString = sEscString & c
```

```
            If IsCharAlpha(CH) = 0 Then
```

```
                If Len(sEscString) > 15 Then
```

```
                    bInEscape = False
```

```
                End If
```

```
            Exit Sub
```

```
            End If
```

```
        Select Case c
```

```
            Case "A"
```

```
                sEscString = Mid$(sEscString, 2)
```

```
            Case "B"
```

```
                sEscString = Mid$(sEscString, 2)
```

```
            Case "C"
```

```
                sEscString = Mid$(sEscString, 2)
```

```
            Case "D"
```

```
                sEscString = Mid$(sEscString, 2)
```

```
            Case "H"
```

```
                sEscString = Mid$(sEscString, 2)
```

```

Case "r"
    sEscString = Mid$(sEscString, 2)
Case "m"
    sEscString = Mid$(sEscString, 2)
Case "r"
    sEscString = Mid$(sEscString, 2)

End Select

binEscape = False
sEscString = ""

End Sub

```

Forma frmTelnet

La siguiente forma realiza las funciones de un cliente Telnet, realiza la conexión con el Host, envía y recibe información y se desconecta del Host. En el caso del presente proyecto el Host es el conmutador por lo que las rutinas solamente pueden realizar conexión y desconexión con un conmutador Ericsson MD-110.

```

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"
Object = "{3B7C8863-D78F-101B-B9B5-04021C009402}#1.1#0"; "RICHTX32.OCX"
Begin VB.Form frmTelnet
    BorderStyle = 3 'Fixed Dialog
    Caption = "Telnet Client"
    ClientHeight = 6120
    ClientLeft = 1095
    ClientTop = 570
    ClientWidth = 10500
    FillColor = &H00800000&
    BeginProperty Font
        Name = "Fixedsys"
        Size = 9
        Charset = 0
        Weight = 400
        Underline = 0 'False
        Italic = 0 'False
        Strikethrough = 0 'False
    EndProperty
    ForeColor = &H00000000&
    KeyPreview = -1 'True
    LinkTopic = "Form1"
    PaletteMode = 1 'UseZOrder
    ScaleHeight = 6120
    ScaleWidth = 10500
    Visible = 0 'False
    Begin RichTextLib.RichTextBox txtPantalla
        Height = 3495
        Left = 0
        TabIndex = 0
        Top = 600
        Width = 10455
        _ExtentX = 18441
        _ExtentY = 6165
        _Version = 327681
        HideSelection = 0 'False
        ScrollBars = 3
        TextRTF = "$"telnet.frx":0000
    End
    Begin MSWinsockLib.Winsock WinsockClient
        Left = 6120
        Top = 600
        _ExtentX = 741
        _ExtentY = 741
        _Version = 327681
    End
    Begin VB.Timer tmrTimer
        Left = 6600
        Top = 600
    End
End
Attribute VB_Name = "frmTelnet"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Const GO_NORM = 0

Const GO_ESC1 = 1
Const GO_ESC2 = 2
Const GO_ESC3 = 3
Const GO_ESC4 = 4
Const GO_ESC5 = 5

Const GO_IAC1 = 6
Const GO_IAC2 = 7
Const GO_IAC3 = 8
Const GO_IAC4 = 9
Const GO_IAC5 = 10
Const GO_IAC6 = 11

Const SUSP = 237
Const ABORT = 238
Const SE = 240
Const NOP = 241
Const DM = 242
Const BREAK = 243
Const IP = 244
Const AO = 245
Const AYT = 246
Const EC = 247
Const EL = 248
Const GOAHEAD = 249
Const SB = 250
Const WILLTEL = 251
Const WONTTEL = 252
Const DOTEL = 253

Const DONTTEL = 254
Const IAC = 255

Const BINARY = 0
Const ECHO = 1
Const RECONNECT = 2
Const SGA = 3
Const AMSN = 4
Const STATUS = 5

```

```

Const TIMING = 6
Const RCTAN = 7
Const OLW = 8
Const OPS = 9
Const OCRD = 10
Const OHTS = 11
Const OHTD = 12
Const OFFD = 13
Const OVTS = 14
Const OVTD = 15
Const OLFD = 16
Const XASCII = 17
Const LOGOUT = 18
Const BYTEM = 19
Const DET = 20
Const SUPDUP = 21
Const SUPDUPOUT = 22
Const SENDLOC = 23
Const TERMTYPE = 24
Const EOR = 25
Const TACACSUID = 26
Const OUTPUTMARK = 27
Const TERMLOCNUM = 28
Const REGIME3270 = 29
Const X3PAD = 30
Const NAWS = 31
Const TERMSPEED = 32
Const TFLOWCNTRL = 33
Const LINEMODE = 34
Const DISPLOC = 35
Const ENVIRON = 36
Const AUTHENTICATION = 37
Const UNKNOWN39 = 39
Const EXTENDED_OPTIONS_LIST = 255
Const RANDOM_LOSE = 256

Public bReceiving As Boolean

Private iParsedata(10) As Integer
Private iPpno As Integer

Private bControl_on As Boolean

Private bSw_ugoahead As Boolean
Private bSw_igoahead As Boolean
Private bSw_echo As Boolean
Private bSw_linemode As Boolean
Private bSw_termsent As Boolean
Private bSubstate As Boolean

Dim bBanEnviaS As Boolean
Dim iCuentaS As Integer

Public Function FnConecta(sDireccion) As Boolean
Dim iCuenta As Long
On Error Resume Next
With WinsockClient
If .State <> 0 Then
.Close

.RemotePort = 0
.LocalPortPort = 0
Do
Loop Until .State = 0
bConectado = False
End If

```

```

.RemoteHost = sDireccion
.RemotePort = "2001"
.Connect

DoEvents
iCuenta = 0
While Right(txtPantalla.Text, 10) <> "Password: "
And iCuenta < 10000
iCuenta = iCuenta + 1
DoEvents
Wend
If Right(txtPantalla.Text, 10) = "Password: "
Then
FnEjecutaComando "redes"
DoEvents
iCuenta = 0
While Right(txtPantalla.Text, 12) <> "Password OK" & Chr(10) And iCuenta < 10000
iCuenta = iCuenta + 1
DoEvents
Wend
If Right(txtPantalla.Text, 12) = "Password OK" & Chr(10) Then
EnviaSS
DoEvents
iCuenta = 0
While Right(txtPantalla.Text, 2) <> "<" & Chr(28) And iCuenta < 10000
iCuenta = iCuenta + 1
DoEvents
Wend
If Right(txtPantalla.Text, 2) = "<" & Chr(28)
Then
FnEjecutaComando "HELP"
DoEvents
iCuenta = 0
While Right(txtPantalla.Text, 2) <> "<" & Chr(28) And iCuenta < 10000 Or frmTelnet.txtPantalla.Find("(C) COPYRIGHT ERICSSON") = -1
iCuenta = iCuenta + 1
DoEvents
Wend
If Right(txtPantalla.Text, 2) = "<" & Chr(28) And frmTelnet.txtPantalla.Find("(C) COPYRIGHT ERICSSON") <> -1 Then
FnConecta = True
bConectado = True
End If
End If
End If

End If
If iCuenta >= 10000 Then
FnConecta = False
End If
End With

End Function
Public Function FnLogOff() As Boolean
Dim iCuenta As Long

If WinsockClient.State <> sckClosed Then
WinsockClient.SendData Chr(24)
DoEvents
iCuenta = 0

```



```

While Right(txtPantalla.Text, 2) <> "<" & Chr(28)
And iCuenta < 10000
    iCuenta = iCuenta + 1
    DoEvents
Wend
If Right(txtPantalla.Text, 2) = "<" & Chr(28) Then
    FnEjecutaComando "O;"
    DoEvents
    iCuenta = 0
    While Right(txtPantalla.Text, 11) <> "LOGGED
OFF" & Chr(250) And iCuenta < 10000
        iCuenta = iCuenta + 1
        DoEvents
    Wend
    WinsockClient_Close
    DoEvents
    sRecibido = ""
    txtPantalla.Text = ""
End If
If iCuenta >= 10000 Then
    FnLogOff = False
Else
    FnLogOff = True
End If
End If
End Function
Public Function FnEjecutaComando(sComando As
String) As Boolean
    If WinsockClient.State <> sckClosed Then
        WinsockClient.SendData sComando & Chr(13) &
Chr(10)
        sRecibido = ""
        FnEjecutaComando = True
    Else
        FnEjecutaComando = False
    End If
End Function
Public Sub EnviaSS()
    Dim iCounter As Integer
    If WinsockClient.State <> sckClosed Then
        bBanEnviaS = True
        iCuentaS = 0
        tmrTimer.Interval = 5
    End If
End Sub
Private Sub tmrTimer_Timer()
    If Right(txtPantalla.Text, 3) <> "SSS" Then
        WinsockClient.SendData "S"
    Else
        tmrTimer.Interval = 0
        WinsockClient.SendData ";" & Chr(13) & Chr(10)
    End If
End Sub
Private Sub Form_QueryUnload(Cancel As Integer,
UnloadMode As Integer)
    WinsockClient_Close
End Sub
Private Sub Form_Unload(Cancel As Integer)
    Unload Me
End Sub
Private Sub WinsockClient_Close()
    With WinsockClient
        .Close
        .RemotePort = 0
        .LocalPort = 0
    End With

```

```

End With
bConectado = False
End Sub
Private Sub WinsockClient_Connect()
    Dim sConnectionString As String
    bSw_ugoahead = True
    bSw_igoahead = False
    bSw_echo = True
    bSw_linemode = False
    bSw_termsent = False
    bSubstate = False
    sConnectionString = Chr$(IAC) & Chr$(DOTE) &
Chr$(ECHO) _
        & Chr$(IAC) & Chr$(DOTE) &
Chr$(SGA) _
        & Chr$(IAC) & Chr$(WILLTEL) &
Chr$(NAWS) _
        & Chr$(IAC) & Chr$(WILLTEL) &
Chr$(TERMTYPE) _
        & Chr$(IAC) & Chr$(WILLTEL) &
Chr$(TERMSPEED)
    WinsockClient.SendData sConnectionString
End Sub
Private Sub WinsockClient_DataArrival(ByVal
bytesTotal As Long)
    Dim byCH() As Byte
    Dim iTest() As Integer
    Dim iCounter As Integer
    Static iCmd As Byte
    If Not bReceiving Then
        bReceiving = True
    Else
        Exit Sub
    End If
    If (bytesTotal > 0) Then
        WinsockClient.GetData byCH, vbByte + vbArray,
bytesTotal
        For iCounter = 0 To bytesTotal - 1
            Select Case iCmd
                Case GO_NORM
                    iCmd = FnProcess_Char(byCH(iCounter))
                Case GO_IAC1
                    iCmd = fnIAC1(byCH(iCounter))
                Case GO_IAC2
                    iCmd = fnIAC2(byCH(iCounter))
                Case GO_IAC3
                    iCmd = fnIAC3(byCH(iCounter))
                Case GO_IAC4
                    iCmd = fnIAC4(byCH(iCounter))
                Case GO_IAC5
                    iCmd = fnIAC5(byCH(iCounter))
                Case GO_IAC6
                    iCmd = fnIAC6(byCH(iCounter))
            End Select
        Next iCounter
    End If

```

```

End If
txtPantalla.Text = sRecibido
bReceiving = False
End Sub

```

Private Function fnIAC1(CH As Byte) As Integer

```

fnIAC1 = GO_NORM

Select Case CH
Case DOTEL
    fnIAC1 = GO_IAC2
Case DONTTEL
    fnIAC1 = GO_IAC6
Case WILLTEL
    fnIAC1 = GO_IAC3
Case WONTTEL
    fnIAC1 = GO_IAC4
Case SB
    fnIAC1 = GO_IAC5
    iPpno = 0
Case SE
    Select Case iParsedata(0)
    Case TERMTYPE
        If iParsedata(1) = 1 Then
            WinsockClient.SendData Chr$(IAC) &
            Chr$(SB) & Chr$(TERMTYPE) & "DEC-VT100" &
            Chr$(0) & Chr$(IAC) & Chr$(SE)
        End If
        Case TERMSPEED
            If iParsedata(1) = 1 Then
                WinsockClient.SendData Chr$(IAC) &
                Chr$(WILLTEL) & Chr$(CH)
                WinsockClient.SendData Chr$(IAC) &
                Chr$(SB) & Chr$(TERMSPEED) & Chr$(0) &
                Chr$(0) & Chr$(IAC) & Chr$(SE)
            End If
        End Select
    End Select
End Select

```

End Function

Private Function fnIAC2(CH As Byte) As Integer

```

fnIAC2 = GO_NORM

Select Case CH
Case BINARY
    WinsockClient.SendData Chr$(IAC) &
    Chr$(WONTTEL) & Chr$(BINARY)
Case ECHO
    WinsockClient.SendData Chr$(IAC) &
    Chr$(WONTTEL) & Chr$(ECHO)
Case NAWS
    WinsockClient.SendData Chr$(IAC) & Chr$(SB) &
    Chr$(NAWS) & Chr$(0) & Chr$(80) & Chr$(0) &
    Chr$(24) & Chr$(IAC) & Chr$(SE)
Case SGA
    If Not bSw_igoahead Then
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WILLTEL) & Chr$(SGA)
        bSw_igoahead = True
    End If
    Case TERMTYPE
        If Not bSw_termsent Then

```

```

        bSw_termsent = True
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WILLTEL) & Chr$(TERMTYPE)
        WinsockClient.SendData Chr$(IAC) &
        Chr$(SB) & Chr$(TERMTYPE) & _
        Chr$(0) & "VT100" & Chr$(IAC) & Chr$(SE)
    End If

```

```

    Case TERMSPEED
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WILLTEL) & Chr$(TERMSPEED)

        WinsockClient.SendData Chr$(IAC) & Chr$(SB) &
        Chr$(TERMSPEED) & Chr$(0)
        WinsockClient.SendData "57600,57600"
        WinsockClient.SendData Chr$(IAC) & Chr$(SE)

```

```

    Case TFLOWCNTRL
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case LINEMODE
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case STATUS
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case TIMING
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case DISPLOC
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case ENVIRON
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case UNKNOWN39
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)

```

```

    Case AUTHENTICATION
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WILLTEL) & Chr$(CH)

```

```

        WinsockClient.SendData Chr$(IAC) & _
        Chr$(SB) & _
        Chr$(AUTHENTICATION) & _
        Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0)
    & _
        Chr$(IAC) & _
        Chr$(SE)

```

```

    Case Else
        WinsockClient.SendData Chr$(IAC) &
        Chr$(WONTTEL) & Chr$(CH)
    End Select

```

End Function

Private Function fnIAC3(CH As Byte) As Integer

```

fnIAC3 = GO_NORM

```

```

Select Case CH
Case ECHO

```

```

    If Not bSw_echo Then
        bSw_echo = True
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(ECHO)
    End If
    Case SGA
        If Not bSw_ugoahead Then
            bSw_ugoahead = True
            WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(SGA)
        End If

    Case TERMSPEED
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case TFLOWCNTRL
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case LINEMODE
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case STATUS
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case TIMING
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case DISPLOC
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case ENVIRON
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case UNKNOWN39
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case Else
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)
    End Select

End Function

Private Function fnIAC4(CH As Byte) As Integer

    fnIAC4 = GO_NORM

    Select Case CH

        Case ECHO
            If bSw_echo = True Then
                WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(ECHO)
                bSw_echo = False
            End If

        Case SGA
            WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(SGA)
            bSw_ugoahead = False

```

```

    Case TERMSPEED
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case TFLOWCNTRL
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case LINEMODE
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case STATUS
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case TIMING
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case DISPLOC
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case ENVIRON
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case UNKNOWN39
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)

    Case Else
        WinsockClient.SendData Chr$(IAC) &
Chr$(DONTTEL) & Chr$(CH)
    End Select

End Function

Private Function fnIAC5(CH As Byte) As Integer

    Dim ich As Integer

    ich = CH
    If ich = IAC Then
        fnIAC5 = GO_IAC1
        Exit Function
    End If

    iParsedata(iPpno) = ich
    iPpno = iPpno + 1

    fnIAC5 = GO_IAC5

End Function

Private Function fnIAC6(CH As Byte) As Integer

    fnIAC6 = GO_NORM

    Select Case CH
        Case SE

        Case ECHO
            If Not bSw_echo Then
                bSw_echo = True

```

```

WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(ECHO)
End If
Case SGA
If Not bSw_ugoahead Then
bSw_ugoahead = True
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(SGA)
End If

Case TERMSPEED
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case TFLOWCNTRL
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case LINEMODE
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case STATUS
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case TIMING

```

```

WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case DISPLOC
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case ENVIRON
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case UNKNOWN39
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)

Case Else
WinsockClient.SendData Chr$(IAC) &
Chr$(WONTTEL) & Chr$(CH)
End Select

End Function
Public Sub LimpiaSeleccionPantalla()
bxtPantalla.SelStart = 0
bxtPantalla.SelLength = 0
End Sub

```

Forma frmActualizaInf

Esta forma es la forma principal de la actualización de información en el conmutador. Mediante esta se realizan consultas y modificaciones en la información de las extensiones interactuando con el cliente Telnet e interpretando la información recibida del conmutador.

```

Begin VB.Form frmActualizaInf
Caption = "Actualización de Información en
Conmutador"
ClientHeight = 4980
ClientLeft = 60
ClientTop = 345
ClientWidth = 6285
LinkTopic = "Form1"
MaxButton = 0 'False
MDIChild = -1 'True
MinButton = 0 'False
ScaleHeight = 4980
ScaleWidth = 6285
Begin VB.CommandButton btnCodigo
Caption = "Codi&go"
Enabled = 0 'False
Height = 360
Left = 4920
TabIndex = 32
Top = 2520
Width = 1230
End
Begin VB.CommandButton btnNombre
Caption = "&Nombre"
Enabled = 0 'False
Height = 360
Left = 4920
TabIndex = 31
Top = 2040
Width = 1230
End

```

```

Begin VB.CommandButton btnExtLibres
Caption = "E&xt. Libres"
Height = 360
Left = 4920
TabIndex = 30
Top = 600
Width = 1230
End
Begin VB.CommandButton btnEQUlibre
Caption = "&EQU's Libres"
Height = 360
Left = 4920
TabIndex = 29
Top = 120
Width = 1230
End
Begin VB.CommandButton btnDesactivar
Caption = "&Desactivar"
Enabled = 0 'False
Height = 360
Left = 4920
TabIndex = 28
Top = 1560
Width = 1230
End
Begin VB.CommandButton btnActivar
Caption = "&Activar"
Height = 360
Left = 4920
TabIndex = 27
Top = 1080
End

```

```

Width      = 1230
End
Begin VB.Frame Frame2
Height     = 3495
Left      = 120
TabIndex  = 4
Top       = 1440
Width     = 4695
Begin VB.TextBox txtIDN
Enabled    = 0 'False
Height     = 285
Left      = 2760
TabIndex  = 22
Top       = 3120
Width     = 855
End
Begin VB.TextBox txtGDND
Enabled    = 0 'False
Height     = 285
Left      = 2760
TabIndex  = 21
Top       = 2760
Width     = 855
End
Begin VB.TextBox txtGH
Enabled    = 0 'False
Height     = 285
Left      = 2760
TabIndex  = 20
Top       = 2400
Width     = 855
End
Begin VB.TextBox txtGCP
Enabled    = 0 'False
Height     = 285
Left      = 2760
TabIndex  = 19
Top       = 2040
Width     = 855
End
Begin VB.TextBox txtBell
Enabled    = 0 'False
Height     = 285
Left      = 2760
TabIndex  = 18
Top       = 1680
Width     = 855
End
Begin VB.TextBox txtCategoria
Enabled    = 0 'False
Height     = 285
Left      = 1320
TabIndex  = 17
Top       = 1200
Width     = 2295
End
Begin VB.TextBox txtEQU
Enabled    = 0 'False
Height     = 285
Left      = 1320
TabIndex  = 16
Top       = 870
Width     = 2295
End
Begin VB.TextBox txtNombre
Enabled    = 0 'False
Height     = 285
Left      = 1320
TabIndex  = 15
Top       = 525

```

```

Width      = 2295
End
Begin VB.TextBox txtTipoExt
Enabled    = 0 'False
Height     = 285
Left      = 1320
TabIndex  = 14
Top       = 195
Width     = 1335
End
Begin VB.Label lblIDN
AutoSize   = -1 'True
Caption    = "Número individual de diversión:"
Height     = 195
Index      = 4
Left      = 480
TabIndex  = 13
Top       = 3165
Width     = 2205
End
Begin VB.Label lblGDND
AutoSize   = -1 'True
Caption    = "Grupo de no molestar:"
Height     = 195
Index      = 3
Left      = 1125
TabIndex  = 12
Top       = 2805
Width     = 1560
End
Begin VB.Label lblGH
AutoSize   = -1 'True
Caption    = "Grupo de búsqueda:"
Height     = 195
Index      = 2
Left      = 1230
TabIndex  = 11
Top       = 2445
Width     = 1455
End
Begin VB.Label lblGCP
AutoSize   = -1 'True
Caption    = "Grupo de contestación de
llamadas:"
Height     = 195
Index      = 1
Left      = 135
TabIndex  = 10
Top       = 2085
Width     = 2550
End
Begin VB.Label lblbell
AutoSize   = -1 'True
Caption    = "Timbre común:"
Height     = 195
Left      = 1635
TabIndex  = 9
Top       = 1725
Width     = 1050
End
Begin VB.Label lblCategoria
AutoSize   = -1 'True
Caption    = "Categoría:"
Height     = 195
Index      = 0
Left      = 465
TabIndex  = 8
Top       = 1245
Width     = 750
End

```

```

Begin VB.Label lblTipoExt
    AutoSize = -1 'True
    Caption = "Tipo Extension:"
    Height = 195
    Left = 120
    TabIndex = 7
    Top = 240
    Width = 1095
End
Begin VB.Label lblEQU
    AutoSize = -1 'True
    Caption = "EQU:"
    Height = 195
    Left = 825
    TabIndex = 6
    Top = 915
    Width = 390
End
Begin VB.Label lblNombre
    AutoSize = -1 'True
    Caption = "Nombre:"
    Height = 195
    Left = 615
    TabIndex = 5
    Top = 570
    Width = 600
End
End
Begin VB.CommandButton btnCerrar
    Caption = "&Cerrar"
    Height = 360
    Left = 4920
    TabIndex = 0
    Top = 3000
    Width = 1230
End
Begin VB.Frame Frame1
    Height = 1455
    Left = 120
    TabIndex = 1
    Top = 0
    Width = 4695
    Begin VB.TextBox txtDireccion
        Enabled = 0 'False
        Height = 285
        Left = 1080
        TabIndex = 26
        Top = 600
        Width = 3495
    End
    Begin VB.TextBox txtConmutador
        Enabled = 0 'False
        Height = 285
        Left = 1080
        TabIndex = 25
        Top = 240
        Width = 3495
    End
    Begin VB.ComboBox cboExtension
        Height = 315
        Left = 1080
        Sorted = -1 'True
        Style = 2 'Dropdown List
        TabIndex = 2
        Top = 960
        Width = 1215
    End
    End
    Begin VB.Label Label2
        AutoSize = -1 'True
        Caption = "Dirección:"
        Height = 195
        Left = 300
        TabIndex = 24
        Top = 600
        Width = 720
    End
    Begin VB.Label Label1
        AutoSize = -1 'True
        Caption = "Conmutador:"
        Height = 195
        Left = 120
        TabIndex = 23
        Top = 240
        Width = 900
    End
    End
    Begin VB.Label lblExtension
        AutoSize = -1 'True
        Caption = "Extensión:"
        Height = 195
        Left = 285
        TabIndex = 3
        Top = 1020
        Width = 735
    End
    End
Attribute VB_Name = "frmActualizaInf"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Public iConmutador As Integer
Public sDireccion As String
Public sNombre1 As String
Public sNombre2 As String
Public bAceptar As Boolean
Public iCategoria As Integer
Public sCodigo As String
Public bBorrar As Boolean

Private Sub btnActivar_Click()
    MDIPrincipal.MousePointer = 11
    frmActivarExt.Show vbModal

    MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnCerrar_Click()
    Unload Me
End Sub

Private Sub btnCodigo_Click()
    Dim sExtension As String

    frmCodigoAut.Show vbModal

    If Not bAceptar Then Exit Sub

    MDIPrincipal.MousePointer = 11

    DoEvents
    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) And frmTelnet.txtPantalla.Find("PASSWORD")
<> -1
        DoEvents
    Wend

```

```

    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
    Then
        sExtension =
        cboExtension.List(cboExtension.ListIndex)
        If Not
        frmTelnet.FnEjecutaComando("AUCOE:DIR=" &
        sExtension & ";") Then
            MsgBox "No es posible ejecutar comandos",
            vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<"
            & Chr(28) Or frmTelnet.txtPantalla.Find("AUCOE:DIR="
            & sExtension & ";") = -1
                DoEvents
            Wend
            If Not bBorrar And
            Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
            frmTelnet.txtPantalla.Find("AUCOE:DIR=" & sExtension
            & ";") <> -1 Then
                If Not
                frmTelnet.FnEjecutaComando("AUCOI:DIR=" &
                sExtension & ",CILCOD=" & sExtension & ",AUTH=" &
                sCodigo & ",CAT=" & iCategoria & ";") Then
                    MsgBox "No es posible ejecutar
                    comandos", vbCritical, "Error"
                    Exit Sub
                Else
                    DoEvents

                    While Right(frmTelnet.txtPantalla.Text, 2)
                    <> "<" & Chr(28) Or
                    frmTelnet.txtPantalla.Find("AUCOI:DIR=" & sExtension
                    & ",CILCOD=" & sExtension & ",AUTH=" & sCodigo &
                    ",CAT=" & iCategoria & ";") = -1
                        DoEvents
                    Wend
                    If Right(frmTelnet.txtPantalla.Text, 2) = "<"
                    & Chr(28) And
                    frmTelnet.txtPantalla.Find("AUCOI:DIR=" & sExtension
                    & ",CILCOD=" & sExtension & ",AUTH=" & sCodigo &
                    ",CAT=" & iCategoria & ";") <> -1 Then
                        cboExtension_Click
                    End If
                End If
            Else
                cboExtension_Click
            End If
        End If
        MDIPrincipal.MousePointer = 3

    End Sub

    Private Sub btnDesactivar_Click()
    Dim iRespuesta As VbMsgBoxResult
    Dim sExtension As String
    Dim sComando As String

    iRespuesta = MsgBox("¿ Desea desactivar la
    extensión ?", vbExclamation + vbYesNo, "Precaución")
    If iRespuesta = vbNo Then Exit Sub

    MDIPrincipal.MousePointer = 11

    DoEvents

```

```

    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" &
    Chr(28) And frmTelnet.txtPantalla.Find("PASSWORD")
    <> -1
        DoEvents
    Wend
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
    Then
        sExtension =
        cboExtension.List(cboExtension.ListIndex)
        If Not
        frmTelnet.FnEjecutaComando("AUCOE:DIR=" &
        sExtension & ";") Then
            MsgBox "No es posible ejecutar comandos",
            vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<"
            & Chr(28) Or frmTelnet.txtPantalla.Find("AUCOE:DIR="
            & sExtension & ";") = -1
                DoEvents
            Wend
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" &
            Chr(28) And frmTelnet.txtPantalla.Find("AUCOE:DIR="
            & sExtension & ";") <> -1 Then
                sComando = If(bTipoExt.Text = "ATS",
                "EXTEE", "KSEXE")
                If Not
                frmTelnet.FnEjecutaComando(sComando & ":DIR=" &
                sExtension & ";") Then
                    MsgBox "No es posible ejecutar
                    comandos", vbCritical, "Error"
                    Exit Sub
                Else
                    DoEvents
                    While Right(frmTelnet.txtPantalla.Text, 2)
                    <> "<" & Chr(28) Or
                    frmTelnet.txtPantalla.Find(sComando & ":DIR=" &
                    sExtension & ";") = -1
                        DoEvents
                    Wend
                    If Right(frmTelnet.txtPantalla.Text, 2) = "<"
                    & Chr(28) And frmTelnet.txtPantalla.Find(sComando &
                    ":DIR=" & sExtension & ";") <> -1 Then
                        If Not
                        frmTelnet.FnEjecutaComando("YES") Then
                            MsgBox "No es posible ejecutar
                            comandos", vbCritical, "Error"
                            Exit Sub
                        Else
                            DoEvents
                            While
                            Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
                            frmTelnet.txtPantalla.Find("YES") = -1
                                DoEvents
                            Wend
                            If Right(frmTelnet.txtPantalla.Text, 2)
                            = "<" & Chr(28) And frmTelnet.txtPantalla.Find("YES")
                            <> -1 Then
                                MsgBox "Se elimino la extension "
                                & sExtension, vbInformation, "Información"
                                cboExtension.RemoveItem
                                cboExtension.ListIndex
                                cboExtension_Click
                            End If
                        End If
                    End If
                End If
            End If
        End If
    End If

```



```

End If
MDIPrincipal.MousePointer = 0

End Sub

Private Sub btnEQULibre_Click()
    MDIPrincipal.MousePointer = 11

    frmEQULibre.Show vbModal

    MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnExtLibres_Click()
    MDIPrincipal.MousePointer = 11

    frmExtLibres.Show vbModal

    MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnNombre_Click()
    Dim sRespuesta As String
    Dim sExtension As String
    Dim sComando As String

    frmNombre.Show vbModal

    If Not bAceptar Then Exit Sub

    MDIPrincipal.MousePointer = 11

    DoEvents
    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) And frmTelnet.txtPantalla.Find("PASSWORD")
<> -1
    DoEvents
    Wend
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
Then
        sExtension =
cboExtension.List(cboExtension.ListIndex)
        sNombre1 = If(sNombre1 = "", "", sNombre1)
        sNombre2 = If(sNombre2 = "", "", sNombre2)

        If Not frmTelnet.FnEjecutaComando("NIINI:DIR="
& sExtension & ",NAME1=" & Chr(34) & sNombre1 &
Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34)
& ",PRES=1;") Then
            MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or frmTelnet.txtPantalla.Find("NIINI:DIR=" &
sExtension & ",NAME1=" & Chr(34) & sNombre1 &
Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34)
& ",PRES=1;") = -1
                DoEvents
            Wend
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" &
Chr(28) And frmTelnet.txtPantalla.Find("NIINI:DIR=" &
sExtension & ",NAME1=" & Chr(34) & sNombre1 &
Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34)
& ",PRES=1;") <> -1 Then
                cboExtension_Click
            End If
        End If
    End If
End Sub

```

```

End If
MDIPrincipal.MousePointer = 0

End Sub

Private Sub cboExtension_Click()
    Dim iCounter As Integer
    Dim sExtension As String

    bdTipoExt.Text = ""
    bdBell.Text = ""
    bdCategoria.Text = ""
    bdEQU.Text = ""
    bdGCP.Text = ""
    bdGDND.Text = ""
    bdGH.Text = ""
    bdIDN.Text = ""
    bdNombre.Text = ""
    btnDesactivar.Enabled = False
    btnNombre.Enabled = False
    btnCodigo.Enabled = False

    If cboExtension.ListIndex = -1 Then Exit Sub

    MDIPrincipal.MousePointer = 11
    btnDesactivar.Enabled = True
    btnNombre.Enabled = True
    btnCodigo.Enabled = True

    DoEvents
    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) And frmTelnet.txtPantalla.Find("PASSWORD")
<> -1
        DoEvents
    Wend
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
Then
        sExtension =
cboExtension.List(cboExtension.ListIndex)
        If Not
frmTelnet.FnEjecutaComando("SUDIP:DIR=" &
sExtension & ";") Then
            MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or frmTelnet.txtPantalla.Find("SUDIP:DIR="
& sExtension & ";") = -1
                DoEvents
            Wend
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" &
Chr(28) And frmTelnet.txtPantalla.Find("SUDIP:DIR=" &
sExtension & ";") <> -1 Then

                frmTelnet.LimpiaSeleccionPantalla
                iCounter = frmTelnet.txtPantalla.Find("TYPE
OF DIRECTORY NUMBER :") + 28
                While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 27
                    bdTipoExt.Text = bdTipoExt.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
                    iCounter = iCounter + 1
                Wend
            End If
        End If
    End If
End Sub

```

```

frmTelnet.LimpiaSeleccionPantalla
iCounter = frmTelnet.txtPantalla.Find("NAME
:") + 28
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 27
    txtNombre.Text = txtNombre.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

frmTelnet.LimpiaSeleccionPantalla
iCounter = frmTelnet.txtPantalla.Find("EQU
TERMD ADN") + 23
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 22
    iCounter = iCounter + 1
Wend
iCounter = iCounter + 2
If iCounter > 24 Then txtEQU.Text =
Trim(Mid(frmTelnet.txtPantalla.Text, iCounter, 12))

frmTelnet.LimpiaSeleccionPantalla
iCounter = frmTelnet.txtPantalla.Find("CAT
TRAF SERV CDIV ROC TRM ADC
CUST") + 67
If iCounter > 66 Then txtCategoria.Text =
Trim(Mid(frmTelnet.txtPantalla.Text, iCounter, 4))

frmTelnet.LimpiaSeleccionPantalla
iCounter =
frmTelnet.txtPantalla.Find("COMMON BELL: ") + 13
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 12
    txtBell.Text = txtBell.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

frmTelnet.LimpiaSeleccionPantalla
iCounter =
frmTelnet.txtPantalla.Find("GROUP CALL PICKUP: ") +
19
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 18
    txtGCP.Text = txtGCP.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

frmTelnet.LimpiaSeleccionPantalla
iCounter =
frmTelnet.txtPantalla.Find("GROUP HUNTING: ") + 15
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 14

```

```

    txtGH.Text = txtGH.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

frmTelnet.LimpiaSeleccionPantalla
iCounter =
frmTelnet.txtPantalla.Find("GROUP DO NOT
DISTURB: ") + 22
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 21
    txtGDND.Text = txtGDND.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

frmTelnet.LimpiaSeleccionPantalla
iCounter =
frmTelnet.txtPantalla.Find("INDIVIDUAL DIVERSION
NUMBER: ") + 29
While Mid(frmTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) <> Chr(10)
And iCounter < Len(frmTelnet.txtPantalla.Text) And
iCounter > 28
    txtIDN.Text = txtIDN.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
    iCounter = iCounter + 1
Wend

End If
End If
End If
MDIPrincipal.MousePointer = 0
End Sub

Private Sub Form_Load()
Dim VntArreglo As Variant
Dim iCounter As Integer
Dim rsCon As Recordset

frmActualizaInf.Height = 5385
frmActualizaInf.Width = 6405

If bConectado Then
    MsgBox "Se esta utilizando la conexión al
conmutador." & Chr(13) & "Cierre las ventanas que la
utilizan", vbCritical, "ErroE"
    Exit Sub
End If

frmConmutador.Show vbModal

If iConmutador <> 0 And sDireccion <> "" Then
    MDIPrincipal.MousePointer = 11

    EjecutaQuery rsCon, "SpSelConmutador " &
iConmutador
    If Not rsCon.EOF Then
        txtConmutador.Text = iConmutador & " - " &
FnEsNulo(rsCon("Descripcion"), "")
    End If
    txtDireccion.Text = sDireccion

```

```
Global sCONFIGURACION As String
Global sCENTRO As String
Global sCENTROID As String
```

```
Global bBanderaConexion As Boolean
```

```
Declare Function GetPrivateProfileString Lib "kernel32"
Alias "GetPrivateProfileStringA" (ByVal
lpApplicationName As String, ByVal lpKeyName As
Any, ByVal lpDefault As String, ByVal lpReturnedString
As String, ByVal nSize As Long, ByVal lpFileName As
String) As Long
Public Sub Main()
Dim x As Date
```

```
On Error GoTo etiq_error
```

```
    AsignaValoresIniciales
    Set gConn = CreateObject("ADODB.Connection")
    frm_principal.Show
```

```
Exit Sub
etiq_error:
    Envia_mail "Main", Err.Number, Err.Description
    Resume Next
End Sub
```

```
Public Sub Descompone_Registro(Vnt_registro As
Variant, _
    VntR_Fecha As Variant, _
    VntR_Conmutador As Variant, _
    VntR_Extension As Variant, _
    VntR_Duracion As Variant, _
    VntR_Codigo As Variant, _
    VntR_Telefono As Variant)
Dim Vnt_Auxiliar As Variant
On Error GoTo etiq_error
```

```
frm_principal.Mensaje "Descomponiendo Registro"
```

```
VntR_Fecha = Mid$(Vnt_registro, 4, 8)
```

```
If sLONGITUD = "4" Then
    '—para 4 dígitos en ID conmutador
    If sCENTRO = "CON_ID" Then
        VntR_Conmutador = Mid$(Vnt_registro, 59, 4)
        VntR_Extension = Mid$(Vnt_registro, 63, 4)
    Else
        Vnt_Auxiliar = Mid$(Vnt_registro, 59, 8)
        VntR_Conmutador = If(Len(Trim(Vnt_Auxiliar)) >
4, Mid$(Vnt_Auxiliar, 1, 4), sCENTROID)
        VntR_Extension = If(Len(Trim(Vnt_Auxiliar)) > 4,
Mid$(Vnt_Auxiliar, 5, 4), Vnt_Auxiliar)
    End If
Else
    '— para 2 dígitos en ID conmutador
    If sCENTRO = "CON_ID" Then
        VntR_Conmutador = Mid$(Vnt_registro, 59, 2)
        VntR_Extension = Mid$(Vnt_registro, 61, 4)
    Else
        Vnt_Auxiliar = Mid$(Vnt_registro, 59, 6)
        VntR_Conmutador = If(Len(Trim(Vnt_Auxiliar)) >
4, Mid$(Vnt_Auxiliar, 1, 2), sCENTROID)
        VntR_Extension = If(Len(Trim(Vnt_Auxiliar)) > 4,
Mid$(Vnt_Auxiliar, 3, 4), Vnt_Auxiliar)
    End If
End If
```

```
VntR_Duracion = Mid$(Vnt_registro, 13, 4)
VntR_Codigo = Mid$(Vnt_registro, 86, 6)
VntR_Telefono = Mid$(Vnt_registro, 38, 20)
```

```
frm_principal.Mensaje ""
Exit Sub
etiq_error:
    Envia_mail "Descompone_Registro", Err.Number,
Err.Description
    frm_principal.Mensaje "ERROR descomponiendo el
registro"
    Resume Next
```

```
End Sub
```

```
Public Sub Envia_mail(Vnt_rutina As Variant, _
    Vnt_num_error As Variant, _
    Vnt_descripcion As Variant)
```

```
Dim MpsSesion As Object
Dim Mpmmensaje As Object
Dim ObjRecipiente As Object
```

```
On Error GoTo etiq_error
frm_principal.Mensaje "Enviando Mail"
```

```
    Set MpsSesion = CreateObject("mapi.session")
    MpsSesion.Logon profileName:=sPROFILE
    Set Mpmmensaje =
MpsSesion.Outbox.Messages.Add
    Mpmmensaje.subject = "Error en Tarificador"
    Mpmmensaje.Text = Format$(Now, "dd/MMM/yyyy
hh:mm:ss") & " Error en rutina: " & Vnt_rutina & " (" &
Vnt_num_error & "- " & Vnt_descripcion & ")")
    Set ObjRecipiente = Mpmmensaje.Recipients.Add
```

```
    ObjRecipiente.Name = sEMAIL
    ObjRecipiente.Resolve
    Mpmmensaje.Send showDialog:=0
    MpsSesion.Logoff
    frm_principal.Mensaje ""
```

```
Exit Sub
etiq_error:
```

```
    Escribe_log Vnt_rutina, Vnt_num_error,
Vnt_descripcion
    frm_principal.Mensaje "ERROR Enviando Mail"
    Resume Next
```

```
End Sub
```

```
Public Sub Escribe_log(Vnt_rutina As Variant, _
    Vnt_num_error As Variant, _
    Vnt_descripcion As Variant)
```

```
On Error GoTo etiq_error
frm_principal.Mensaje "Escribiendo en Log"
```

```
Open sPATH_APLICACION & "tarif.log" For Append
Access Read Write Shared As #1
Write #1, Format$(Now, "dd/MMM/yyyy hh:mm:ss") & "
Error en rutina: " & Vnt_rutina & " (" & Vnt_num_error &
"- " & Vnt_descripcion & ")")
Close #1
```

```
frm_principal.Mensaje ""
```

```

Exit Sub
etiq_error:
    Resume Next
    frm_principal.Mensaje "ERROR Escribiendo en
LOG"

End Sub
Public Sub AsignaValoresIniciales()
Dim sNoexiste As String
Dim Lerror As Long
On Error GoTo etiq_error

sPATHINI = App.Path & "\TARIFICA.INI"

sPATH_APLICACION = FnBuscaValorINI(sPATHINI,
"PATHS", "APLICACION")
sEMAIL = FnBuscaValorINI(sPATHINI, "MAIL",
"DIRECCION")
sPROFILE = FnBuscaValorINI(sPATHINI, "MAIL",
"PROFILE")
sCONMUTADORID = FnBuscaValorINI(sPATHINI,
"CONMUTADOR", "ID")
sLONGITUD = FnBuscaValorINI(sPATHINI,
"CONMUTADOR", "LONGITUD")
sUSUARIO = FnBuscaValorINI(sPATHINI,
"CONECTIVIDAD", "USUARIO")
sPWD = FnBuscaValorINI(sPATHINI,
"CONECTIVIDAD", "PASSWORD")
sPWD = If(sPWD <> "NULL", sPWD, "")
sDSN = FnBuscaValorINI(sPATHINI,
"CONECTIVIDAD", "DSN")
sPUERTO = FnBuscaValorINI(sPATHINI, "SERIAL",
"PUERTO")
sCONFIGURACION = FnBuscaValorINI(sPATHINI,
"SERIAL", "CONFIGURACION")
sCENTRO = FnBuscaValorINI(sPATHINI,
"CONMUTADOR", "CENTROAPOYO")
sCENTROID = FnBuscaValorINI(sPATHINI,
"CONMUTADOR", "CENTROAPOYOID")

Exit Sub
etiq_error:
    Envia_mail "AsignaValoresIniciales", Err.Number,
Err.Description
    Resume Next

End Sub

Function FnBuscaValorINI(sArchivoINI As String,
sSeccion As String, sEntrada As String) As String
Dim sNoexiste As String
Dim sCadena As String * 80
Dim iTamaño As Integer
Dim iResultado As Integer
Dim sResultado As String
On Error GoTo etiq_error
    sCadena = ""
    iTamaño = 80
    If Dir(sArchivoINI, 0) <> "" Then
        iResultado = GetPrivateProfileString(sSeccion,
sEntrada, sNoexiste, sCadena, iTamaño, sArchivoINI)
        If (Not (IsNull(sCadena))) And (sCadena <> "")
Then
            sResultado = Trim$(Mid$(sCadena, 1,
iResultado))
            If sResultado = sNoexiste Then
                MsgBox "No se encontró la sección [" +
sSeccion + "]" y entrada " + sEntrada + " = en el archivo "
+ sArchivoINI + ", verifique."

```

```

Shell "notepad.exe " & sArchivoINI,
vbNormalFocus
End
End If
End If
Else
    MsgBox "No se encontró el archivo " + sArchivoINI
+ ", verifique."
End
End If
Rem ---
FnBuscaValorINI = sResultado
Exit Function
etiq_error:
    Envia_mail "FnBuscaValorINI", Err.Number,
Err.Description
    FnBuscaValorINI = ""
    Resume Next
End Function

```

```

Public Function FnConectaBD() As Boolean
On Error GoTo etiq_error
frm_principal.Mensaje "Conectando a BD"
FnConectaBD = True

If gConn.State = 0 Then
    gConn.Open "UID=" & sUSUARIO & ";pwd=" &
sPWD & ";dsn=" & sDSN
    If gConn.State = 1 Then
        FnConectaBD = True
        bBanderaConexion = False
    Else
        FnConectaBD = False
    End If
End If

```

```

frm_principal.Mensaje ""
Exit Function
etiq_error:
    FnConectaBD = False
    If bBanderaConexion = False Then
        Envia_mail "Se perdio conexion con la BD",
Err.Number, Err.Description
        bBanderaConexion = True
    End If
    frm_principal.Mensaje "ERROR Conectando BD"
    Resume Next
End Function

```

```

Public Function FnDesconectaBD() As Boolean
On Error GoTo etiq_error
frm_principal.Mensaje "Desconectando BD"

```

```

If IsObject(gConn) Then
    If gConn.State <> 0 Then
        gConn.Close
        FnDesconectaBD = True
    End If
End If
frm_principal.Mensaje ""

Exit Function
etiq_error:
    FnDesconectaBD = False
    Envia_mail "FnDesconectaBD", Err.Number,
Err.Description
    frm_principal.Mensaje "ERROR Desconectando BD"
    Resume Next
End Function

```

```

Public Function FnHora(VntFecha As Variant) As Variant
Dim VntHora As Variant
On Error GoTo etiq_error

VntHora = Mid(VntFecha, 5, 2)

If IsNumeric(VntHora) Then
    FnHora = VntHora
Else
    FnHora = 0
End If
Exit Function
etiq_error:
    FnHora = "0"
    Envia_mail "FnHora", Err.Number, Err.Description
    frm_principal.Mensaje "ERROR Calculando Hora"
    Resume Next

End Function
Public Function FnMin(VntFecha As Variant) As Variant
Dim VntMin As Variant
On Error GoTo etiq_error

VntMin = Mid(VntFecha, 7, 2)

If IsNumeric(VntMin) Then
    FnMin = VntMin
Else
    FnMin = 0
End If
Exit Function
etiq_error:
    FnMin = 0
    Envia_mail "FnFecha", Err.Number, Err.Description
    frm_principal.Mensaje "ERROR Calculando Minutos"
    Resume Next

End Function

Public Function FnFecha(VntFecha As Variant) As Variant
Dim VntMes As Variant
Dim VntDia As Variant
Dim VntHora As Variant
Dim VntMin As Variant
Dim VntAnio As Variant
On Error GoTo etiq_error

VntMes = Mid(VntFecha, 1, 2)
VntDia = Mid(VntFecha, 3, 2)
VntAnio = Year(Now)

If IsNumeric(VntMes) And IsNumeric(VntDia) And IsNumeric(VntAnio) Then

    If VntMes = 12 And VntDia = 31 Then
        If Month(Now) = 1 And Day(Now) = 1 Then
            VntAnio = Year(Now) - 1
        End If
    End If

    FnFecha = Str(VntAnio) & Format(Str(VntMes), "00") & Format(Str(VntDia), "00")
Else
    FnFecha = "19000101"
End If

```

```

Exit Function
etiq_error:
    FnFecha = "19000101"
    Envia_mail "FnFecha", Err.Number, Err.Description
    frm_principal.Mensaje "ERROR Calculando Fecha"
    Resume Next

End Function
Public Function FnDuracionSeg(VntDuracion As Variant) As Variant
Dim VntHoras As Variant
Dim VntMin As Variant
Dim VntSeg As Variant

On Error GoTo etiq_error
If Len(VntDuracion) >= 4 Then
    VntHoras = Mid(VntDuracion, 1, 1)
    VntMin = Mid(VntDuracion, 2, 2)
    VntSeg = Mid(VntDuracion, 4, 1)
    If IsNumeric(VntHoras) And IsNumeric(VntMin) And IsNumeric(VntSeg) Then
        FnDuracionSeg = VntHoras * 3600 + VntMin * 60 + VntSeg * 6
    End If
Else
    FnDuracionSeg = 0
End If

Exit Function
etiq_error:
    FnDuracionSeg = 0
    Envia_mail "FnDuracionSeg", Err.Number, Err.Description
    frm_principal.Mensaje "ERROR Calculando duraci3n"
    Resume Next

End Function

Public Function FnTipoLlamada(VntTelefono As Variant) As Variant
Dim RsResultado As Recordset

On Error GoTo etiq_error
frm_principal.Mensaje "Clasificando Tipo Llamada"

If gConn.State = 1 Then
    Set RsResultado = gConn.Execute("spClasificaTipoLlamada " & VntTelefono & "")
    FnTipoLlamada = RsResultado.Fields("TipoLlamada")
Else
    FnTipoLlamada = "NULL"
End If
frm_principal.Mensaje ""

Exit Function
etiq_error:
    FnTipoLlamada = "NULL"
    Envia_mail "FnTipoLlamada", Err.Number, Err.Description
    frm_principal.Mensaje "ERROR Clasificando Tipo de Llamada"
    Resume Next

End Function

```

```

Public Function FnClaveLD(VntTelefono As Variant,
VntTipoLlamadaId As Variant) As Variant
Dim RsResultado As Recordset

On Error GoTo etiq_error
frm_principal.Mensaje "Buscando Clave LD"

If gConn.State = 1 Then
Set RsResultado =
gConn.Execute("spBuscaClaveLD " & VntTelefono &
" ," & VntTipoLlamadaId)
FnClaveLD = RsResultado.Fields("ClaveLDID")
Else
FnClaveLD = "NULL"
End If
frm_principal.Mensaje ""

Exit Function
etiq_error:
FnClaveLD = "NULL"
Envia_mail "FnClaveLD", Err.Number,
Err.Description
frm_principal.Mensaje "ERROR en Búsqueda de
Clave LD"

Resume Next
End Function

Public Sub InsertaLlamada(VntFecha As Variant,
VntHora As Variant, VntMin As Variant, VntConmutador
As Variant, VntExtension As Variant, VntTipoLlamada
As Variant, VntClaveLD As Variant, VntDuracion As
Variant, VntCodigoA As Variant, VntTelefono As
Variant)
Dim RsResultado As Recordset

On Error GoTo etiq_error
frm_principal.Mensaje "Insertando Llamada"

If gConn.State = 1 Then
Set RsResultado = gConn.Execute("spInsLlamada " &
VntFecha & " ," & VntHora & " ," & VntMin & " ," &
Trim(VntConmutador) & " ," & Trim(VntExtension) & " ," &
If(Trim(VntTipoLlamada) <> "" ,
Trim(VntTipoLlamada), "NULL") & " ," &
If(Trim(VntClaveLD) <> "" , Trim(VntClaveLD), "NULL")
& " ," & Trim(VntDuracion) & " ," & If(Trim(VntCodigoA)
<> "" , Trim(VntCodigoA), "NULL") & " ," &
Trim(VntTelefono) & """)
End If
frm_principal.Mensaje ""

Exit Sub
etiq_error:
Envia_mail "InsertaLlamada", Err.Number,
Err.Description
frm_principal.Mensaje "ERROR Insertando Llamada"
Resume Next

End Sub

Public Sub AlmacenaLlamadaBD(llamada As Variant)
Dim VntFecha As Variant
Dim VntFecha1 As Variant
Dim VntHora As Variant
Dim VntMin As Variant
Dim VntConn As Variant
Dim VntExt As Variant
Dim VntDur As Variant
Dim VntCodigo As Variant

```

```

Dim VntTel As Variant
Dim VntTipo As Variant
Dim VntClave As Variant
On Error GoTo etiq_error

frm_principal.Mensaje "Almacenando en BD"

Descompone_Registro llamada, VntFecha1, VntConn,
VntExt, VntDur, VntCodigo, VntTel
VntFecha = FnFecha(VntFecha1)
VntHora = FnHora(VntFecha1)
VntMin = FnMin(VntFecha1)
VntTipo = FnTipoLlamada(VntTel)
VntClave = FnClaveLD(VntTel, VntTipo)
VntDur = FnDuracionSeg(VntDur)
InsertaLlamada VntFecha, VntHora, VntMin, VntConn,
VntExt, VntTipo, VntClave, VntDur, VntCodigo, VntTel

frm_principal.Mensaje ""

Exit Sub
etiq_error:
Envia_mail "AlmacenaLlamadaBD", Err.Number,
Err.Description
frm_principal.Mensaje "ERROR Almacenando en
BD"
Resume Next

End Sub

Public Sub AlmacenaLlamadaArchivo(VntLlamada As
Variant)
On Error GoTo etiq_error
frm_principal.Mensaje "Almacenando Localmente"

Open sPATH_APLICACION & "TarifAlm.TXT" For
Append Access Read Write Shared As #1
Print #1, VntLlamada
Close #1

frm_principal.Mensaje ""

Exit Sub
etiq_error:
Envia_mail "AlmacenaLlamadaArchivo", Err.Number,
Err.Description
frm_principal.Mensaje "ERROR Almacenando
localmente"
Resume Next

End Sub

Public Sub ProcesaLlamadasArchivo()
Dim VntLlamada As Variant
On Error GoTo etiq_error

frm_principal.Mensaje "Procesando Llamadas del
Almacenamiento Local"

If Dir(sPATH_APLICACION & "TarifAlm.TXT") <> ""
Then
Open sPATH_APLICACION & "TarifAlm.TXT" For
Input Access Read Shared As #1

While Not EOF(1)
Line Input #1, VntLlamada
AlmacenaLlamadaBD (VntLlamada)
Wend
Close #1
Kill sPATH_APLICACION & "TarifAlm.TXT"
End If

```



```

frm_principal.Mensaje ""

Exit Sub
eti_q_error:
  Close #1
  Envia_mail "ProcesaLlamadasArchivo", Err.Number,
  Err.Description
  frm_principal.Mensaje "ERROR Procesando
  Llamadas de Almacenamiento Local"
  Resume Next
End Sub

Public Sub ProcesaLlamada(VntLlamada As Variant)
On Error GoTo eti_q_error
frm_principal.Mensaje "Procesando Llamada"

If FnConectaBD Then
  ProcesaLlamadasArchivo

```

```

  AlmacenaLlamadaBD VntLlamada
  'FnDesconectaBD
Else
  AlmacenaLlamadaArchivo VntLlamada
End If

DoEvents
frm_principal.Mensaje ""

Exit Sub
eti_q_error:
  Envia_mail "ProcesaLlamada", Err.Number,
  Err.Description
  frm_principal.Mensaje "Error Procesando Llamada"
  Resume Next

End Sub

```

Forma frmTarifica

Mediante esta forma se recibe la información de las llamadas del conmutador y se procesan utilizando las rutinas generales del módulo principal. He aquí que el proceso principal es a través del evento OnComm, que permite establecer una comunicación con el conmutador.

```

Object = "{648A5603-2C6E-101B-82B6-
000000000014}#1.1#0"; "MSCOMM32.OCX"
Object = "{831FDD16-0C5C-11D2-A9FC-
0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
Begin VB.Form frm_principal
  BorderStyle = 4 'Fixed ToolWindow
  Caption = "Tarificador"
  ClientHeight = 930
  ClientLeft = 645
  ClientTop = 2910
  ClientWidth = 7920
  LinkTopic = "Form1"
  MaxButton = 0 'False
  MinButton = 0 'False
  ScaleHeight = 930
  ScaleWidth = 7920
  ShowInTaskbar = 0 'False
  StartUpPosition = 2 'CenterScreen
  Begin MSComctlLib.StatusBar SbrMensajes
    Align = 2 'Align Bottom
    Height = 300
    Left = 0
    TabIndex = 2
    Top = 630
    Width = 7920
    _ExtentX = 13970
    _ExtentY = 529
    _Version = 393216
  End
  BeginProperty Panels {8E3867A5-8586-11D1-
B16A-00C0F0283628}
    NumPanels = 1
    BeginProperty Panel1 {8E3867AB-8586-11D1-
B16A-00C0F0283628}
      Object.Width = 12700
      MinWidth = 12700
    EndProperty
  EndProperty
End
Begin MSCommLib.MSComm ComPuerto
  Left = 1560

```

```

  Top = 480
  _ExtentX = 1005
  _ExtentY = 1005
  _Version = 327681
  DTREnable = -1 'True
  InputLen = 1
  RThreshold = 1
  RTSEnable = -1 'True
  BaudRate = 2400
End
Begin VB.CommandButton Command1
  Caption = "Procesar"
  Height = 375
  Left = 120
  TabIndex = 1
  Top = 480
  Visible = 0 'False
  Width = 1335
End
Begin VB.TextBox TxtLlamada
  Height = 285
  Left = 120
  TabIndex = 0
  Top = 120
  Width = 7695
End
Attribute VB_Name = "frm_principal"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Private Sub Command1_Click()

On Error GoTo eti_q_error

Mensaje "Procesando Llamada"
ProcesaLlamada TxtLlamada.Text
Mensaje ""

```



```

Exit Sub
etiq_error:
    Envia_mail "Command1_click", Err.Number,
    Err.Description
    Resume Next

End Sub

Private Sub ComPuerto_OnComm()
    Dim VntCuenta As Variant
    Dim VntBuffer As Variant

    On Error GoTo etiq_error
        'Mensaje ComPuerto.Input

        Do While ComPuerto.InBufferCount > 0
            VntCuenta = VntCuenta + 1
            If ComPuerto.InBufferCount >= 113 Then
                VntBuffer = ComPuerto.Input
                TxtLlamada.Text = VntBuffer
                If Len(VntBuffer) >= 113 Then
                    ProcesaLlamada VntBuffer
                End If
            ElseIf VntCuenta > 30000 Then
                VntBuffer = ComPuerto.Input
                Exit Do
            End If
        DoEvents
    Loop
Exit Sub
etiq_error:
    Envia_mail "ComPuerto_OnComm", Err.Number,
    Err.Description
    Resume Next

End Sub

Private Sub Form_DblClick()
    'Command1_Click
End Sub

Private Sub Form_Load()

    Dim sDummy As String
    On Error GoTo etiq_error
        Mensaje "Abriendo Puerto"
        ComPuerto.InputLen = 113
        ComPuerto.RThreshold = 5
        ComPuerto.CommPort = sPUERTO
        ComPuerto.Settings = sCCNFIGURACION
        ComPuerto.PortOpen = True

        sDummy = ComPuerto.Input
        Mensaje ""

Exit Sub
etiq_error:
    Envia_mail "Form_Load", Err.Number,
    Err.Description
    Mensaje "ERROR Abriendo Puerto"
    Resume Next

End Sub

Private Sub Form_QueryUnload(Cancel As Integer,
    UnloadMode As Integer)
    Dim VntOpcion As Variant
    On Error GoTo etiq_error

```

```

        Mensaje "Saliendo"
        VntOpcion = MsgBox("Seguro que desea salir del
        Tarificador", vbYesNo + vbExclamation, "Precaución")
        If VntOpcion = vbYes Then
            ComPuerto.PortOpen = False
            End
        Else
            Cancel = True
        End If
        Mensaje ""

Exit Sub
etiq_error:
    Envia_mail "Form_QueryUnLoad", Err.Number,
    Err.Description
    Resume Next

End Sub

Public Sub Mensaje(VntMensaje As Variant)
    On Error GoTo etiq_error
        SbrMensajes.Panels(1).Text = VntMensaje
Exit Sub
etiq_error:
    Resume Next
End Sub

```

6.4. Problemas y soluciones.

Conexión al conmutador.

Para poder conectarse al conmutador e iniciar la ejecución de comandos es necesario enviar una cadena de letras "S" con las cuales el conmutador determina la velocidad de la terminal que realiza la conexión. Inicialmente se intentó enviar una cadena de 10 y 15 letras "S", pero el conmutador nunca respondió. Después se creó un ciclo que enviara 10 y 15 letras "S" pero de forma individual, pero tampoco se obtuvo respuesta del conmutador. Finalmente se agregó un Timer a la forma de Visual Basic y se programó para que cada segundo enviara una letra "S", no se obtuvo respuesta del conmutador hasta que se enviaron las letras "S" con un intervalo de tiempo de cinco milésimas de segundo.

Manejo de errores en Visual Basic.

Para que el proceso de recolección de llamadas fuera confiable fue necesario codificar una rutina de manejo de errores para cada uno de los procedimientos y funciones del proceso permitiendo detectar en que rutina se produjo el error y tomar una acción para el error producido.

Creación de menú dinámico en Visual Basic.

La aplicación necesita de la creación de un menú dinámicamente dependiendo de las opciones que tenga habilitado el perfil del usuario. La forma que debe ejecutar cada opción está almacenada en la tabla de Opciones de la base de datos de la aplicación. Sin embargo es imposible instanciar una forma a partir de la variable string donde se almacena el nombre de la forma que debe ejecutar la opción. Para poder ejecutar la forma a partir del string fue necesario crear un Case para cada nombre de las formas y en cada caso ejecutar el estatuto forma.show.

6.5. Resumen.

6.5.1. Elementos utilizados en Visual Basic.

Módulo-Forma	Procesos	Funciones	Controles
Recopilación de llamadas			
frmprincipal.frm	5	0	3
modPrincipal.bas	10	9	1
Acceso			
frmLogin.frm	2	0	8
Actualización de Inf. Conmutador			
frmActivarExt.frm	9	0	16
frmActualizaInf.frm	10	0	33
frmCodigoAut.frm	5	0	7
frmConmutador.frm	4	0	7
frmEQULibre.frm	2	0	5
frmExtLibres.frm	2	0	3
frmNombre.frm	3	0	6
Administración de Solicitudes			
frmProcActivarExt.frm	9	0	16
frmProcAsignExtension.frm	13	0	18
frmProcCodigoAut.frm	4	0	6
frmProcesa.frm	4	0	58
frmProcNombre.frm	2	0	6
frmProcReporte.frm	2	0	2
frmSolDetalle.frm	2	0	57
frmSolicitudes.frm	11	0	14
Actualización de Inf. Extensiones			
frmExtension.frm	11	0	10
frmAsignExtension.frm	13	0	18
frmDatExtension.frm	4	0	32
frmPermisosExt.frm	8	0	12
Rutinas Generales			
MDIPrincipal.frm	9	0	2
frmtnet.frm	9	9	4
ModGeneral.bas	5	5	2
Vt100.bas	1	1	0
Total	24	15	8

Tabla 6.5. Resumen de construcción en VB.

6.5.2. Elementos utilizados en ASP's.

Página	Procesos	Funciones	Controles	Frames
Módulo de Acceso				
Default.asp	1	0	0	0
Menu.asp	8	0	11	0
Procesa_Opcion.asp	1	0	0	0
Módulo de Configuración				
<i>Actualizador de Áreas Genéricas</i>				
Fra_AreasGen.asp	1	0	0	4
Abc_AreasGen1.asp	3	0	1	0
Abc_AreasGen2.asp	4	0	1	0
Abc_AreasGen3.asp	7	0	9	0
Graba_AreasGen.asp	4	0	10	0
Actualiza_areasgen.asp	1	0	0	0
<i>Actualizador de Claves de LD</i>				
Fra_ClavesLD.html	1	0	0	2
Abc_ClaveLD1.asp	3	0	1	0
Abc_ClaveLD2.asp	8	0	9	0
Graba_ClaveLD.asp	4	0	8	0
Actualiza_ClaveLD.asp	1	0	0	0
<i>Actualizador de Conmutadores</i>				
Abc_con.asp	8	0	10	0
Graba_Con.asp	4	0	7	0
Actualiza_Con.asp	1	0	0	0
<i>Actualizador de Opciones</i>				
Abc_opcion.asp	8	0	10	0
Graba_opcion.asp	4	0	10	0
Actualiza_Opcion.asp	1	0	0	0
<i>Actualizador de Perfiles</i>				
Abc_perfil.asp	8	0	8	0
Graba_perfil.asp	4	0	7	0
Actualiza_Perfil.asp	1	0	0	0
<i>Actualizador de Tel. Permitidos</i>				
Abc_TelPer.asp	8	0	8	0
Graba_TelPer.asp	4	0	6	0
Actualiza_TelPer.asp	1	0	0	0
<i>Actualizador de Tipos de Ext.</i>				
Abc_tipoext.asp	8	0	7	0
Graba_TipoExt.asp	4	0	6	0
Actualiza_TipoExt.asp	1	0	0	0
<i>Actualizador de Tipos de Llam.,</i>				
Abc_tipoLlam.asp	8	0	9	0
Graba_TipoLlam.asp	4	0	8	0

Página	Procesos	Funciones	Controles	Frames
Actualiza_TipoLlam.asp	1	0	0	0
<i>Actualizador de Tipos de Sol.</i>				
Abc_tipoSol.asp	8	0	8	0
Actualiza_TipoSol.asp	1	0	0	0
Graba_TipoSol.asp	4	0	6	0
<i>Actualizador de Usuarios</i>				
Abc_usuario.asp	8	0	11	0
Graba_usuario.asp	4	0	10	0
Actualiza_Usuario.asp	1	0	0	0
Módulo de Conciliación				
Conciliacion_carga.asp	5	0	5	0
Conciliacion_carga2.asp	5	0	0	0
Conciliacion_reporte.asp	5	4	5	0
Conciliacion_manual.asp	5	4	25	0
Módulo de Depuración				
DepuraBD.asp	5	2	5	0
DepuraBD2.asp	1	0	1	0
Módulo de Reportes				
Cuerpo.asp	1	0	0	0
<i>Reporte de Costos</i>				
Fra_Costos.asp	1	0	0	2
Enc_Costos.asp	5	11	18	0
Rpt_Costos.asp	1	3	1	0
<i>Directorio de extensiones</i>				
Fra_Directorío.asp	1	0	0	2
Enc_Directorío.asp	5	0	4	0
Rpt_Directorío.asp	1	0	0	0
<i>Reporte gráfico por depto.</i>				
Fra_GrafxDepto.asp	1	0	0	2
Enc_GrafxDepto.asp	5	11	22	0
Rpt_GrafxDepto.asp	1	6	1	0
<i>Reporte gráfico por socio</i>				
Fra_GrafxSocio.asp	1	0	0	4
Enc_GrafxSocio1.asp	3	0	1	0
Enc_GrafxSocio2.asp	2	0	2	0
Enc_GrafxSocio3.asp	5	11	23	0
Rpt_GrafxSocio.asp	1	7	1	0
<i>Reporte gráfico por ubicación</i>				
Fra_GrafxUbi.asp	1	0	0	2
Enc_GrafxUbi.asp	5	11	21	0
Rpt_GrafxUbi.asp	1	7	1	0
<i>Reporte gráfico por U.N.</i>				
Fra_GrafxUN.asp	1	0	0	2
Enc_GrafxUN.asp	5	11	21	0

Página	Procesos	Funciones	Controles	Frames
Rpt_GrafxUN.asp	1	7	1	0
<i>Reporte de Llamadas por Depto.</i>				
Fra_LlamadasxDpto.asp	1	0	0	4
Enc_LlamadasxDpto.asp	1	0	0	0
Enc_LlamadasxDpto1.asp	3	0	1	0
Enc_LlamadasxDpto2.asp	3	0	2	0
Enc_LlamadasxDpto3.asp	5	11	17	0
Rpt_LlamadasxDepto.asp	1	3	1	0
Rpt_LlamadasxDeptoDetalle2.asp	1	1	1	0
Rpt_LlamadasxTExtDetalle2.asp	1	1	1	0
Rpt_LlamadasxTelefonoDetalle2.asp	1	1	1	0
<i>Reporte de Llamadas por Socio</i>				
Fra_LlamadasxSocio.asp	1	0	0	2
Enc_LlamadasxSocio.asp	5	11	16	0
Rpt_LlamadasxSocio.asp	1	3	1	0
Rpt_LlamadasxSocioDetalle.asp	1	1	1	0
Rpt_LlamadasxSocioTelefono.asp	1	1	1	0
<i>Reporte de Llamadas por Ubicación</i>				
Fra_LlamadasxUbi.asp	1	0	0	2
Enc_LlamadasxUbi.asp	5	11	16	0
Rpt_LlamadasxUbi.asp	1	3	1	0
Rpt_LlamadasxUbiDetalle.asp	1	1	1	0
Rpt_LlamadasxUNDetalle2.asp	1	1	1	0
Rpt_LlamadasxDeptoDetalle3.asp	1	1	1	0
Rpt_LlamadasxTExtDetalle3.asp	1	1	1	0
Rpt_LlamadasxTelefonoDetalle3.asp	1	1	1	0
<i>Reporte de Llamadas por U.N.</i>				
Fra_LlamadasxUN.asp	1	0	0	2
Enc_LlamadasxUN.asp	5	11	16	0
Rpt_LlamadasxUN.asp	1	3	1	0
Rpt_LlamadasxUNDetalle.asp	1	1	1	0
Rpt_LlamadasxDeptoDetalle.asp	1	3	1	0
Rpt_LlamadasxTExtDetalle.asp	1	1	1	0
Rpt_LlamadasxTelefonoDetalle.asp	1	1	1	0
<i>Reporte de Tel. No Involucrados</i>				
Fra_TelNlxSocio.asp	1	0	0	2
Enc_TelNlxSocio.asp	5	11	16	0
Rpt_TelNlxSocio.asp	1	3	1	0
Rpt_TelNlxSocioDetalle.asp	1	1	1	0
<i>Reporte gráfico de Tel. No Invol.</i>				
Fra_TelNlxSocioGraf.asp	1	0	0	2
Enc_TelNlxSocioGraf.asp	6	11	20	0
Rpt_TelNlxSocioGraf.asp	1	5	1	0
<i>Reporte de Máximos por Socio</i>				

Página	Procesos	Funciones	Controles	Frames
Fra_TopxSocio.asp	1	0	0	2
Enc_TopxSocio.asp	6	11	20	0
Rpt_TopxSocio.asp	1	3	1	0
Rpt_TopxSocioDetalle.asp	1	1	1	0
Rpt_TopxTExtDetalle.asp	1	1	1	0
Rpt_TopxTelefonoDetalle.asp	1	1	1	0
Rpt_TopxSocioTelefono.asp	1	1	1	0
<i>Reporte grafico de Máximos por Soc.</i>				
Fra_TopxSocioGraf.asp	1	0	0	2
Enc_TopxSocioGraf.asp	6	11	19	0
Rpt_TopxSocioGraf.asp	1	5	1	0
<i>Reporte de Máximos por U.N.</i>				
Fra_TopxUN.asp	1	0	0	2
Enc_TopxUN.asp	6	11	20	0
Rpt_TopxUN.asp	1	3	1	0
Rpt_TopxUNDetalle.asp	1	1	1	0
Rpt_TopxDeptoDetalle.asp	1	1	1	0
Módulo de Solicitudes				
Solicitud.asp	14	1	40	0
Graba_Sol.asp	4	0	48	0
Actualiza_Sol.asp	1	0	0	0
Módulo de Seguridad				
Seguridad.asp	1	0	0	0
Usuario_invalido.asp	1	0	0	0
Rutinas Generales				
AgregaTelefono.asp	1	0	3	0
Fechas.asp	0	10	0	0
Fechas2.asp	0	15	0	0
Llena_combo.asp	1	0	1	0
Llena_combo2.asp	1	0	1	0
Utilerias.asp	0	5	0	0
Valida_usuario.asp	1	0	0	0
Total	4	30	5	80

Tabla 6.6. Resumen de construcción en ASP's.

7. LIBERACIÓN

De forma que se pueda asegurar el buen funcionamiento del sistema, tanto modularmente como integralmente, se llevaron a cabo una serie de pruebas las cuales se describen a continuación.

7.1. Pruebas en ambiente Intranet.

Las siguientes pruebas fueron realizadas en una PC Pentium con 46MB RAM, Windows 95 y Microsoft Internet Explorer 5.0.

Pruebas de actualización y consulta de conmutadores.

Datos de Entrada

Campo	Dato
ConmutadorID	1610
Descripcion	Centro de Distribución Estrella
DireccionIP	255.255.255.255

Resultados Esperados

- Almacenar en la tabla Conmutador el registro ingresado en caso de una alta.
- Ingresar la clave de un registro y obtener el resto de sus datos en caso de una consulta. Si no existe en la tabla desplegar un mensaje de error.

- Almacenar en la tabla Conmutador las modificaciones realizadas a un registro en caso de una actualización.
- Eliminar el registro en la tabla Conmutador cuando se desee dar de baja. Se debe presentar una pregunta de confirmación de la baja.

Resultados Obtenidos

- Se almacenó el registro en la tabla Conmutador cuando se realizó el alta.
- Los datos restantes del registro fueron desplegados cuando se solicitó la consulta y se desplegó un mensaje de error cuando se ingresó una clave que no existía.
- Se almacenaron los cambios en la tabla cuando se realizó la actualización de los datos del registro.
- El registro fue eliminado cuando se solicitó la baja y se respondió a la pregunta de confirmación.

Pruebas de actualización y consulta de áreas genéricas.

Datos de Entrada

Campo	Dato
UnegocioID	TSIST
Departamento	01221
Descripción	FAX SISTEMAS

Resultados Esperados

- En caso de un alta almacenar los datos ingresados en la tabla de áreas genéricas.
- Para el caso de una consulta permitir la captura de una clave y mostrar el resto de los datos del registro. Si no existe un registro asociado a la clave ingresada desplegar un mensaje de error.
- Cuando se trate de una actualización almacenar los cambios realizados al registro en la tabla de áreas genéricas.
- En el caso de una baja eliminar el registro de la tabla de áreas genéricas. Mostrar una pregunta de confirmación antes de realizar la baja.

Resultados Obtenidos

- El alta se realizó correctamente.
- La consulta mostró el resto de los datos del registro.
- Los cambios realizados a los datos del registro fueron almacenados en la tabla correctamente.
- Se eliminó el registro que se deseaba dar de baja después de responder a la pregunta de confirmación.

*Pruebas de actualización y consulta de claves de larga distancia.**Datos de Entrada*

Campo	Dato
TipoLlamadaID	1
ClaveLD	806
Descripción	Amarillo, Texas, USA

Resultados Esperados

- Almacenar en la tabla de Claves de Larga Distancia el registro ingresado en caso de una alta.
- Ingresar la clave de un registro y obtener el resto de sus datos en caso de una consulta. En caso de que el registro no exista en la tabla desplegar un mensaje de error.
- Almacenar en la tabla de Claves de Larga Distancia las modificaciones realizadas a un registro en caso de una actualización.
- Eliminar el registro en la tabla de Claves de Larga Distancia cuando se desee dar de baja. Se debe presentar una pregunta de confirmación de la baja.

Resultados Obtenidos

- Se almacenó el registro en la tabla de Claves de Larga Distancia cuando se realizó el alta.
- Los datos restantes del registro fueron desplegados cuando se solicitó la consulta y se desplegó un mensaje de error cuando se ingresó una clave que no existía.
- Se almacenaron los cambios en la tabla cuando se realizó la actualización de los datos del registro.
- El registro fue eliminado cuando se solicitó la baja y se respondió a la pregunta de confirmación.

Pruebas de actualización y consulta de opciones.

Datos de Entrada

Campo	Dato
OpcionID	60
Descripción	Reporte de Prueba
TipoOpcion	1
Agrupación	2
Programa	Reporte_Prueba.asp

Resultados Esperados

- En caso de un alta almacenar los datos ingresados en la tabla de opciones.
- En relación a la consulta permitir la captura de una clave y mostrar el resto de los datos del registro. Si no existe un registro asociado a la clave ingresada desplegar un mensaje de error.
- Cuando se trate de una actualización almacenar los cambios realizados al registro en la tabla de opciones.
- En el caso de una baja eliminar el registro de la tabla de opciones. Mostrar una pregunta de confirmación antes de realizar la baja.

Resultados Obtenidos

- El alta se realizó correctamente.
- La consulta mostró el resto de los datos del registro.
- Los cambios realizados a los datos del registro fueron almacenados en la tabla correctamente.
- Se eliminó el registro que se deseaba dar de baja después de responder a la pregunta de confirmación.

Pruebas de actualización y consulta de perfiles.

Datos de Entrada

Campo	Dato
PerfilID	10
Descripción	Perfil de Prueba
OpcionID	1,2,3,4

Resultados Esperados

- Almacenar en la tabla de Perfiles el registro ingresado en caso de una alta. Almacenar también las opciones seleccionadas en la tabla de Perfil_Opcion para tener relacionadas las opciones con su perfil.

- Ingresar la clave de un registro y obtener el resto de sus datos en caso de una consulta. En caso de que el registro no exista en la tabla desplegar un mensaje de error.
- Almacenar en la tabla de Perfiles y en la tabla de Perfil_Opcion las modificaciones realizadas a un perfil en caso de una actualización.
- Eliminar el registro en la tabla de Perfiles y todos los registros que coincidan con el perfil de la tabla de Perfil_Opcion cuando se desee dar de baja. Se debe presentar una pregunta de confirmación de la baja.

Resultados Obtenidos

- Se almacenó el registro en la tabla de Perfiles cuando se realizó el alta y las opciones seleccionadas en la tabla de Perfil_Opcion
- Los datos restantes del registro fueron desplegados cuando se solicitó la consulta y se desplegó un mensaje de error cuando se ingresó una clave que no existía.
- Se almacenaron los cambios en las tablas cuando se realizó la actualización de los datos del registro.
- Los registros fueron eliminados cuando se solicitó la baja y se respondió a la pregunta de confirmación.

Pruebas de actualización y consulta de teléfonos permitidos.

Datos de Entrada

Campo	Dato
Telefono	12345678
Descripción	Teléfono de Prueba

Resultados Esperados

- En caso de un alta almacenar los datos ingresados en la tabla de teléfonos permitidos.
- Para el caso de una consulta permitir la captura de una clave y mostrar el resto de los datos del registro. Si no existe un registro asociado a la clave ingresada desplegar un mensaje de error.
- Cuando se trate de una actualización almacenar los cambios realizados al registro en la tabla de teléfonos permitidos.
- En el caso de una baja eliminar el registro de la tabla de teléfonos permitidos. Mostrar una pregunta de confirmación antes de realizar la baja.

Resultados Obtenidos

- El alta se realizó correctamente.
- La consulta mostró el resto de los datos del registro.

- Los cambios realizados a los datos del registro fueron almacenados en la tabla correctamente.
- Se eliminó el registro que se deseaba dar de baja después de responder a la pregunta de confirmación.

Pruebas de actualización y consulta de tipos de extensión.

Datos de Entrada

Campo	Dato
TipoExtensionID	1
Descripción	Extensión de Socio

Resultados Esperados

- Almacenar en la tabla de Tipos de Extensión el registro ingresado en caso de una alta.
- Ingresar la clave de un registro y obtener el resto de sus datos en caso de una consulta. En caso de que el registro no exista en la tabla desplegar un mensaje de error.
- Almacenar en la tabla de Tipos de Extensión las modificaciones realizadas a un registro en caso de una actualización.
- Eliminar el registro en la tabla de Tipos de Extensión cuando se desee dar de baja. Se debe presentar una pregunta de confirmación de la baja.

Resultados Obtenidos

- Se almacenó el registro en la tabla de Teléfonos Permitidos cuando se realizó el alta.
- Los datos restantes del registro fueron desplegados cuando se solicitó la consulta y se desplegó un mensaje de error cuando se ingresó una clave que no existía.
- Se almacenaron los cambios en la tabla cuando se realizó la actualización de los datos del registro.
- El registro fue eliminado cuando se solicitó la baja y se respondió a la pregunta de confirmación.

Pruebas de actualización y consulta de tipos de llamada.

Datos de Entrada

Campo	Dato
TipoLlamadaID	1
Descripción	LARGA DISTANCIA EU Y CANADA
Prefijo	001
TipoExtraeLada	2

Resultados Esperados

- En caso de un alta almacenar los datos ingresados en la tabla de tipos de llamada.
- Para el caso de una consulta permitir la captura de una clave y mostrar el resto de los datos del registro. Si no existe un registro asociado a la clave ingresada desplegar un mensaje de error.
- Cuando se trate de una actualización almacenar los cambios realizados al registro en la tabla de tipos de llamada.
- En el caso de una baja eliminar el registro de la tabla de tipos de llamada. Mostrar una pregunta de confirmación antes de realizar la baja.

Resultados Obtenidos

- El alta se realizó correctamente.
- La consulta mostró el resto de los datos del registro.
- Los cambios realizados a los datos del registro fueron almacenados en la tabla correctamente.
- Se eliminó el registro que se deseaba dar de baja después de responder a la pregunta de confirmación.

Pruebas de actualización y consulta de tipos de solicitud.

Datos de Entrada

Campo	Dato
Tipo_SoliciutdID	1
Descripción	Activación

Resultados Esperados

- Almacenar en la tabla de Tipos de Solicitud el registro ingresado en caso de una alta.

- Ingresar la clave de un registro y obtener el resto de sus datos en caso de una consulta. En caso de que el registro no exista en la tabla desplegar un mensaje de error.
- Almacenar en la tabla de Tipos de Solicitud las modificaciones realizadas a un registro en caso de una actualización.
- Eliminar el registro en la tabla de Tipos de Solicitud cuando se desee dar de baja. Se debe presentar una pregunta de confirmación de la baja.

Resultados Obtenidos

- Se almacenó el registro en la tabla de Tipos de Solicitud cuando se realizó el alta.
- Los datos restantes del registro fueron desplegados cuando se solicitó la consulta y se desplegó un mensaje de error cuando se ingresó una clave que no existía.
- Se almacenaron los cambios en la tabla cuando se realizó la actualización de los datos del registro.
- El registro fue eliminado cuando se solicitó la baja y se respondió a la pregunta de confirmación.

Pruebas de actualización y consulta de usuarios.

Datos de Entrada

Campo	Dato
UsuarioID	Prueba
Password	Contraseña
SocioID	0565
PerfilID	4

Resultados Esperados

- En caso de un alta almacenar los datos ingresados en la tabla de usuarios.
- Para el caso de una consulta permitir la captura de una clave y mostrar el resto de los datos del registro. Si no existe un registro asociado a la clave ingresada desplegar un mensaje de error.
- Cuando se trate de una actualización almacenar los cambios realizados al registro en la tabla de usuarios.
- En el caso de una baja eliminar el registro de la tabla de usuarios. Mostrar una pregunta de confirmación antes de realizar la baja.

Resultados Obtenidos

- El alta se realizó correctamente.

- La consulta mostró el resto de los datos del registro.
- Los cambios realizados a los datos del registro fueron almacenados en la tabla correctamente.
- Se eliminó el registro que se deseaba dar de baja después de responder a la pregunta de confirmación.

Pruebas de conciliación.

Datos de Entrada

Datos
Archivo de Telmex.
Llamadas recopiladas.

Resultados Esperados

- Relacionar las llamadas recopiladas por el sistema y las del detalle de llamadas de Telmex, intentando relacionar el mayor número de registros posibles por similitud.
- Los registros que no sean relacionados automáticamente mostrarlos para relacionarlos manualmente.

Resultados Obtenidos

- Se relacionó el mayor número de registros similares.
- El resto de los registros fue relacionado manualmente.

Pruebas de depuración de la base de datos.

Datos de Entrada

Campo	Dato
FechaInicial	01/01/1999
FechaFinal	01/02/1999

Resultados Esperados

- Eliminar los registros de las tablas operativas del sistema que se encuentren entre los rangos de fecha proporcionados por el usuario.

Resultados Obtenidos

- Se eliminaron los registros de las tablas operativas que contaban con una fecha entre el rango de fechas proporcionado.

Pruebas de reporte de costos.

Datos de Entrada

Campo	Dato
UnegociolD	-1 (Todas)
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Global

Resultados Esperados

- Obtener el costo de las llamadas realizadas por las unidades de negocio seleccionadas y sus departamentos, en caso de que el reporte sea desglosado.
- Obtener los totales de los costos por cada tipo de agrupación.

Resultados Obtenidos

- Se obtuvieron los costos correctamente.
- Se obtuvieron totales por unidad de negocio y por departamento cuando el reporte fue detallado.

Pruebas de reporte de directorio de extensiones.

Datos de Entrada

Campo	Dato
Búsqueda	GARZA

Resultados Esperados

- Desplegar todos los nombres de socios, áreas genéricas o departamentos y sus extensiones de modo que coincidan con el criterio de búsqueda.

Resultados Obtenidos

- Se desplegaron todos los nombres relacionados con el criterio de búsqueda.

Pruebas de reporte gráfico por departamento.

Datos de Entrada

Campo	Dato
UnegociolD	TSIST
TipoLlamadaID	-1 (Todos)
TipoReporte	Pesos
FechaInicial	01/04/2000
FechaFinal	30/04/2000

Resultados Esperados

- Obtener un reporte gráfico, de barras o de pastel, que representen las llamadas realizadas por los departamentos de la unidad de negocio seleccionada, detallando los tipos de servicios seleccionados por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- Se obtuvo el gráfico seleccionado con la información esperada.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte gráfico por socio.

Datos de Entrada

Campo	Dato
UnegociolD	TSIST
Departamento	01221
TipoLlamadaID	-1 (Todas)
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Tiempos

Resultados Esperados

- Desplegar un gráfico de barras o de pastel que represente las llamadas realizadas por los socios del departamento seleccionado, detallando los tipos de servicios seleccionados por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- Se obtuvo el gráfico seleccionado con la información esperada.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte gráfico por ubicación.

Datos de Entrada

Campo	Dato
UbacionID	1
TipoLlamadaID	-1 (Todas)
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Tiempos

Resultados Esperados

- Obtener un reporte gráfico, de barras o de pastel, que representen las llamadas realizadas por la ubicación seleccionada, detallando los tipos de servicios seleccionados por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- Se obtuvo el gráfico seleccionado con la información esperada.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte gráfico por unidad de negocio.

Datos de Entrada

Campo	Dato
TipoLlamadaID	-1 (Todas)
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Tiempos

Resultados Esperados

- Desplegar un gráfico de barras o de pastel que represente las llamadas realizadas por las unidades de negocio seleccionadas, detallando los tipos de servicios seleccionados por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- Se obtuvo el gráfico seleccionado con la información esperada.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte de llamadas por departamento.

Datos de Entrada

Campo	Dato
UnegocioID	TSIST
Departamento	01221
FechaInicial	01/04/2000
FechaFinal	30/04/2000

Resultados Esperados

- Mostrar la información de las llamadas realizadas por el departamento seleccionado en el rango de fechas ingresado por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- La información de las llamadas del departamento seleccionado fue desplegada correctamente.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte de llamadas por socio.

Datos de Entrada

Campo	Dato
SocioID	0005
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Desplegar los datos de las llamadas realizadas por el socio seleccionado para las fechas ingresadas por el usuario.
- Desplegar un mensaje de error cuando las fechas hayan sido mal capturadas o los rangos estén equivocados.

Resultados Obtenidos

- Se obtuvo la información de las llamadas realizadas por el socio en el rango de fechas ingresado.
- Se mostró un mensaje de error cuando se produjo un error en las fechas.

Pruebas de reporte de llamadas por ubicación.

Datos de Entrada

Campo	Dato
UbicacionID	1
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Pesos

Resultados Esperados

- Mostrar la información de las llamadas realizadas por el departamento seleccionado en el rango de fechas ingresado por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- La información de las llamadas del departamento seleccionado fue desplegada correctamente.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte de llamadas por unidad de negocio.

Datos de Entrada

Campo	Dato
UnegociolD	-1 (Todas)
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Desplegar los datos de las llamadas realizadas por el socio seleccionado para las fechas ingresadas por el usuario.
- Desplegar un mensaje de error cuando las fechas hayan sido mal capturadas o los rangos estén equivocados.

Resultados Obtenidos

- Se obtuvo la información de las llamadas realizadas por el socio en el rango de fechas ingresado.
- Se mostró un mensaje de error cuando se produjo un error en las fechas.

Pruebas de reporte de teléfonos no involucrados.

Datos de Entrada

Campo	Dato
SociolD	0005
FechaInicial	01/04/2000
FechaFinal	30/04/2000

Resultados Esperados

- Mostrar la información de las llamadas realizadas por el departamento seleccionado en el rango de fechas ingresado por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- La información de las llamadas del departamento seleccionado fue desplegada correctamente.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte gráfico de teléfonos no involucrados.

Datos de Entrada

Campo	Dato
SociolD	0005
TipoLlamadaID	1
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Desplegar los datos de las llamadas realizadas por el socio seleccionado para las fechas ingresadas por el usuario.
- Desplegar un mensaje de error cuando las fechas hayan sido mal capturadas o los rangos estén equivocados.

Resultados Obtenidos

- Se obtuvo la información de las llamadas realizadas por el socio en el rango de fechas ingresado.
- Se mostró un mensaje de error cuando se produjo un error en las fechas.

Pruebas de reporte de máximos por socio.

Datos de Entrada

Campo	Dato
SocioID	-1 (Todos)
TipoLlamadaID	1
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Mostrar la información de las llamadas realizadas por el departamento seleccionado en el rango de fechas ingresado por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- La información de las llamadas del departamento seleccionado fue desplegada correctamente.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de reporte gráfico de máximos por socio.

Datos de Entrada

Campo	Dato
SocioID	0005
TipoLlamadaID	1
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Desplegar los datos de las llamadas realizadas por el socio seleccionado para las fechas ingresadas por el usuario.
- Desplegar un mensaje de error cuando las fechas hayan sido mal capturadas o los rangos estén equivocados.

Resultados Obtenidos

- Se obtuvo la información de las llamadas realizadas por el socio en el rango de fechas ingresado.
- Se mostró un mensaje de error cuando se produjo un error en las fechas.

Pruebas de reporte de máximos por unidad de negocio.

Datos de Entrada

Campo	Dato
UnegocioID	-1 (Todas)
TipoLlamadaID	1
FechaInicial	01/04/2000
FechaFinal	30/04/2000
TipoReporte	Llamadas

Resultados Esperados

- Mostrar la información de las llamadas realizadas por el departamento seleccionado en el rango de fechas ingresado por el usuario.
- Si existe un error en las fechas o en los rangos se debe desplegar un error.

Resultados Obtenidos

- La información de las llamadas del departamento seleccionado fue desplegada correctamente.
- Se desplegó el error cuando las fechas fueron mal capturadas.

Pruebas de generación de solicitudes de servicio.

Datos de Entrada

Campo	Dato
SolicitudID	30
SocioID	0005
Extensión	
Nodo	V123E456
Motivo	Prueba
TipoSolicitudID	1
TipoLinea	1
TipoTel	1
Display	Prueba

Resultados Esperados

- Almacenar la solicitud en las tablas de Solicitudes, solicitudes de activación y de cambios según sea el caso.
- Generar automáticamente el identificador de la solicitud en caso de un alta.
- Validar que los datos necesarios para surtir la solicitud estén capturados.
- Mostrar los datos de la solicitud consultada en caso de que esta no haya sido autorizada.
- Actualizar las modificaciones al registro consultado en la tabla de solicitudes.
- Eliminar de la tabla de solicitudes la solicitud consultada en caso de que se desee realizar una baja. Mostrar una pregunta de confirmación.

Resultados Obtenidos

- Se realizó correctamente el alta de la solicitud.
- Se mostró la información de la solicitud cuando esta podía ser consultada.
- La información del registro consultado fue actualizada en sus tablas correspondientes.
- Se eliminó la solicitud de sus respectivas tablas.
- Los datos fueron validados correctamente.

Pruebas de acceso y seguridad.Datos de Entrada

Campo	Dato
Usuario	Prueba
Contraseña	Contraseña

Resultados Esperados

- Crear el menú con las opciones del sistema que el perfil del usuario tiene habilitadas.
- No permitir el acceso a las opciones ni a la información que no tiene acceso el usuario.

Resultados Obtenidos

- Se creó el menú con las opciones que el perfil del usuario tenía habilitadas.
- No fue posible acceder a opciones que el usuario no tenía habilitadas.

7.2. Pruebas en ambiente Visual Basic.

Pruebas de recopilación de llamadas.

Datos de Entrada

Campo	Dato
Llamada	04301136 0021 00 0152550470 101201 1511 00 010125

Resultados Esperados

- Desglosar el registro recibido desde el conmutador y almacenarlo en la tabla de llamadas.
- Si no existe conexión con la base de datos almacenarlo localmente.
- Si existen registros en el almacenamiento local y hay conexión con la BD desglosar cada uno de los registros y almacenarlos en la tabla de llamadas.

Resultados Obtenidos

- Se desglosó correctamente el registro y se almacenó en la tabla de llamadas.
- Se almacenó el registro localmente cuando no existía conexión con la base de datos.
- Cuando existieron registros en el almacenamiento local y había conexión con la base de datos se procesaron los registros y se almacenaron en la tabla de llamadas.

Pruebas de acceso.

Datos de Entrada

Campo	Dato
Usuario	Prueba
Contraseña	Contraseña

Resultados Esperados

- Crear el menú con las opciones del sistema que el perfil del usuario tiene habilitadas.

Resultados Obtenidos

- Se creó el menú con las opciones que el perfil del usuario tenía habilitadas.

Pruebas de actualización de información en el conmutador.

Datos de Entrada

Campo	Dato
ConmutadorID	2010
DireccionIP	110.209.150.254
Extensión	1675

Resultados Esperados

- Mostrar la información de la extensión consultada.
- Desplegar un listado de los EQU's libres del conmutador.
- Mostrar un listado de las extensiones libres del conmutador.
- Activar una extensión.
- Desactivar una extensión.
- Asignar un nombre a una extensión.
- Asignar un código para acceso a servicios.

Resultados Obtenidos

- Se obtuvo correctamente la información de la extensión consultada.
- Fue desplegado un listado de los EQU's libres.
- Se obtuvo un listado de las extensiones libres en el conmutador.
- La extensión fue activada.
- Fue posible desactivar la extensión.
- Se asignó correctamente el nombre a la extensión.
- El código de acceso a servicios fue asignado correctamente.

Pruebas de administración de solicitudes.

Datos de Entrada

Campo	Dato
SolicitudID	30
SociID	0005
Extensión	
Nodo	V123E456
Motivo	Prueba
TipoSolicitudID	1
TipoLinea	1
TipoTel	1
Display	Prueba

Resultados Esperados

- Mostrar los detalles de la solicitud cuando se desea consultar.
- Autorizar la solicitud.
- Cancelar la solicitud.
- Procesar la solicitud automáticamente.

Resultados Obtenidos

- La información de la solicitud fue desplegada correctamente cuando se realizó la consulta.
- La solicitud fue autorizada correctamente.
- La solicitud fue cancelada correctamente.
- Se procesó la solicitud haciendo los cambios correspondientes en el conmutador.

Pruebas de actualización y consulta de información de extensiones.

Datos de Entrada

Campo	Dato
ConmutadorId	2010
Extension	1002
TipoExtensionId	1
CTUA	1234
Signal	PR
Puerto	E1234V456
Rack	E1245
Regleta	10
Fila	1
Par	12
Raiser4	E547
Raiser	V5
CRaiser	E
LFisico	Piso 2
TipoTel	Ericsson 360
NombreDisplay	Prueba

Resultados Esperados

- Almacenar en la tabla de extensión los datos capturados en caso de una alta.
- Mostrar la información de una extensión si se desea consultar.
- Almacenar las modificaciones realizadas a una extensión consultada.
- Borrar de la tabla de extensiones el registro seleccionado cuando se trate de una baja.

- Crear el registro en la tabla correspondiente de modo que se realice una relación entre la extensión y su propietario.
- Crear los registros correspondientes de modo que se relacione la extensión con los tipos de servicio que pueda acceder.

Resultados Obtenidos

- Se almacenaron los datos de la extensión en la tabla de extensiones.
- Los datos de la extensión consultada fueron presentados correctamente.
- Las modificaciones realizadas a la información de la extensión consultada fue actualizada.
- Se eliminó el registro de la tabla de extensiones cuando se solicitó una baja.
- La extensión fue asignada correctamente.
- Los permisos fueron relacionados correctamente.

7.3. Sinopsis de pruebas.

La siguiente tabla muestra datos estadísticos sobre las pruebas realizadas a cada uno de los módulos del sistema.

Módulo	Pruebas Realizadas	Pruebas Satisfactorias
Intranet		
Acceso Intranet	10	6
Solicitudes	28	14
Conciliación	7	4
Reportes	46	32
Depuración	4	2
Seguridad	6	4
Utilerías	28	20
Visual Basic		
Recopilación de Llamadas	32	17
Almacenamiento	5	3
Depuración de Archivos Locales	3	2
Acceso	5	3
Actualización de Información	52	45
Administración de Solicitudes	33	18
Ejecución de Comandos	50	20

Tabla 7.1. Sinopsis de pruebas.

CONCLUSIÓN

La conclusión de este proyecto versa principalmente sobre tres puntos básicos: el área de negocios, la metodología de desarrollo, la construcción y las herramientas.

En cuanto al área de negocios se puede concluir que el desarrollo del presente proyecto para la empresa Supermercados Internacionales HEB, ha sido de gran utilidad en varios aspectos. Uno de estos aspectos es la distribución de los costos entre los departamentos. Como se explica en el segundo capítulo, anteriormente los costos del servicio telefónico se distribuían aplicando porcentajes a cada uno de los departamentos de la empresa. Lo que indica que en este tipo de servicios es necesario tener en cuenta un detalle o desglose de la información que genera los costos, ya que estos se generan dependiendo del giro de trabajo de cada persona o departamento y es imposible generalizar en la distribución de sus costos entre los departamentos de la empresa.

Otro aspecto en el que el proyecto es de gran utilidad es que este hace posible contar con información histórica de los gastos que genera cada una de las extensiones de la red y los socios y los departamentos que le corresponden estos gastos. De modo que esta información histórica permite que los departamentos de la empresa puedan realizar sus presupuestos de manera más exacta y confiable y con esto aprovechar mejor los recursos disponibles.

Un aspecto más que es importante mencionar es que, mediante la información de tiempos y costos de las llamadas generada por el sistema será posible

contar con un factor más para la evaluación de la productividad de los socios y los departamentos ya que se puede observar el tiempo dedicado al uso del servicio telefónico y los costos que genera contra la aportación que el socio o el departamento da a la empresa.

En cuanto a la metodología con la que fue desarrollado el proyecto, el Desarrollo Rápido de Aplicaciones (RAD), los beneficios que presenta son muy importantes. En las primeras dos etapas, el JRP y el JAD, el usuario se encuentra muy involucrado con el desarrollo de sistemas mediante el tipo de determinación de requerimientos y mediante el prototipo. Esto ayuda a que el sistema se ajuste más a las necesidades de los usuarios que cuando se utiliza el desarrollo de sistemas tradicional. Gracias a ese prototipo se puede ir conociendo como será el funcionamiento de su sistema logrando paulatinamente la capacitación del usuario.

Un problema encontrado con esta metodología es que los diagramas generados en las etapas de JRP y de JAD requieren de mucho trabajo para su creación, sobre todo cuando no se cuenta con las herramientas que manejen el estándar en lo que se refiere a diagramación para diseño.

En cuanto a la construcción y las herramientas utilizadas para el desarrollo del proyecto se pueden mencionar varios aspectos. Uno de ellos es el ambiente Intranet, este da muchas facilidades en cuanto a la operación del sistema ya que no hay que realizar instalaciones previas y se puede acceder desde cualquier PC que tenga un browser, sin embargo las interfaces y los procesos que se pueden construir son rústicos y limitados. Además, en este ambiente de desarrollo es necesario contar con un esquema de seguridad más robusto que en los sistemas tradicionales.

En lo referente al ambiente Visual Basic es posible mencionar varios aspectos importantes encontrados en el desarrollo. Uno de estos aspectos es el desarrollo de procesos de alta confiabilidad. En este tipo de procesos Visual Basic se comporta adecuadamente, por lo menos en un ambiente ciento por ciento Microsoft. Sin embargo, el desarrollo debe contener una rutina de manejo de errores en cada una de las funciones y procedimientos contenidos en el código del proceso, de modo que se pueda identificar exactamente donde se originó el error y aplicar una acción correctiva.

Por último, en cuanto a la interacción de la aplicación con el conmutador se obtuvieron buenos resultados con Visual Basic y sus componentes, no obstante, para realizar una interacción con dispositivos electrónicos no involucra solamente el desarrollo de un programa que envíe y reciba los datos del dispositivo, sino que también es necesario conocer la forma de operación del mismo. Ahora con la administración incluida en el proyecto el tiempo de respuesta a las solicitudes de servicio se disminuirá considerablemente evitando

la ejecución manual de comandos en el conmutador mediante el procesamiento automático de las solicitudes.

El diseño flexible y las ventajas que se obtienen de aplicaciones como la desarrollada en este proyecto hace posible que cualquier organización con una infraestructura tecnológica similar pueda hacer uso de la misma aplicación sin necesidad de realizar un nuevo desarrollo, sino que con un poco de mantenimiento es posible que la aplicación se ajuste específicamente a las necesidades de la nueva empresa.

TRABAJOS FUTUROS

Debido a la flexibilidad con la que fue diseñado el sistema, es posible incorporar nuevos procesos y opciones de manera sencilla. A continuación se describen algunas de las posibles mejoras al sistema.

El sistema genera la información de los costos en servicios telefónicos que tiene cada socio y detalla cuales son personales y cuales están involucrados con el negocio. Con esta información y mediante la construcción de una interfase con el sistema de nómina de la empresa se puede realizar el cobro de las llamadas personales que haya realizado un socio. Es necesario aclarar que para lo anterior tendría que establecerse una política de la organización que respalde el cobro de estas llamadas.

Las solicitudes de servicio para el departamento de comunicaciones creadas por el sistema, ayudan a que se cumplan algunas de las tareas que tiene este departamento. A través del análisis de la información de las solicitudes se puede incorporar al sistema un módulo estadístico que ayude a determinar y calcular el desempeño de este departamento en cuanto al servicio que se otorga a las solicitudes.

Actualmente se esta almacenando en la base de datos del sistema el teléfono al cual se realizó la llamada, la fecha cuando se realizó y la duración. Con estos datos es posible calcular mediante tarifas el costo de las llamadas, lo cual podría ser utilizado como medio de contingencia en caso de que ya no sea posible realizar la conciliación de las llamadas con la facturación de Telmex y que el sistema continúe calculando los costos de las llamadas.

La empresa actualmente cuenta con Power Play, que es una herramienta poderosa para el análisis de información y es utilizada en la empresa para analizar la información del Data Warehouse. Mediante esta herramienta y la información almacenada en la base de datos del sistema se puede construir un cubo de información y pudiendo posteriormente realizar análisis más profundos en dado caso que esto sea necesario, haciendo uso de técnicas drill-down en gráficos, clasificaciones por diferentes criterios, asociación con otros tipos de costos, entre otros.

Asimismo, con la infraestructura construida para la consulta de información y ejecución de comandos en el conmutador es posible continuar con el desarrollo de interfases que faciliten otros procesos realizados por el departamento de comunicaciones con el conmutador, como son agregar conmutadores a la red, la configuración de grupos de extensiones, configurar grabaciones y correos de voz, entre otros.

BIBLIOGRAFÍA DE REFERENCIA

- [BIGUS, 1997] Bigus, Joseph P. Constructing Intelligent Agents With Java : A Programmer's Guide to Smarter Applications. John Wiley & Sons. USA 1997.
- [BROPHY, 1997] Brophy, Keith y Koets. Aprendiendo VB Script en 21 días. Prentice Hall Hispanoamericana. México 1997.
- [CORNELL, 1997] Cornell, Gary. Manual Visual Basic 4 para Windows 95 Gary Cornell. McGraw-Hill. España 1997.
- [ERICSSON, 1998] Ericsson. Operation & Maintenance for Consono MD-110. Ericsson. USA 1998.
- [FRANCIS, 1998] Francis, Brian. Professional Active Server Pages 2.0. Worx Press. Reino Unido 1998.
- [GUREWICH, 1998] Gurewich, Nathan. Aprendiendo Visual Basic en 21 días. Prentice Hall. México 1998.
- [HARRINGTON, 1999] Harrington, Jan L. Relational Database Design Clearly Explained. Academic Press/Morgan Kaufmann. USA 1999.
- [HEB, 1999] SUPERMERCADOS INTERNACIONALES H.E.B., S.A. DE C.V. La compañía. En: http://home.hebmex.com/pages/La_compania.htm. México 1999.
- [KERNIGHAN, 1998] Kernighan, Brian W.. Timing Trials: Experiments with Scripting and User-Interface Languages. En: <http://cm.bell-labs.com/cm/cs/who/bwk/interps/pap.html>. Lucent Technologies. USA 1998.
- [LEMAY, 1999] Lemay, Laura. Aprendiendo Java 2 en 21 días. Prentice-Hall. México 1999.
- [MARTIN, 1991] Martin, James. Rapid Application Development. MacMillan. USA 1991
- [McCONNELL, 1996] McConnell, Steve C. Rapid Development : Taming Wild Software Schedules. Microsoft Press. USA 1996.

- [SERVATI, 1998] Servati, Al. La Biblia de Intranet. McGraw Hill. México 1998.
- [SURVEYER, 1999] Surveyer, Jacques. Java and Visual Basic: programming's new breed. En: <http://www.tcp.ca/1999/9906/software/javavb/javavb.html>. Canada Computer Paper Inc..Canada. 1999.
- [VISUAL MINING,1996] Visual Mining. Netcharts: Product Overview. Visual Mining. USA.1996.

BIBLIOGRAFÍA GENERAL

- [AFERGAN, 1998] Afergan, Michael [et al]. Programación en Web 6 en 1. Prentice Hall. México 1998.
- [ASHLEY, 1999] Ashley, Paul [et al]. Practical Intranet Security : Overview of the State of the Art and Available Technologies. Kluwer Academic Publishers. USA 1999.
- [BERG, 1998] Berg, Daniel J. Advanced Techniques for Java Developers. John Wiley & Sons. USA 1998.
- [BIGUS, 1997] Bigus, Joseph P. Constructing Intelligent Agents With Java : A Programmer's Guide to Smarter Applications. John Wiley & Sons. USA 1997.
- [BROPHY, 1997] Brophy, Keith y Koets. Aprendiendo VB Script en 21 días. Prentice Hall Hispanoamericana. México 1997.
- [CHANG, 1998] Chang, Dan. Client/Server Data Access With Java and Xml. John Wiley & Sons. USA 1998.
- [CORNELL, 1997] Cornell, Gary. Manual Visual Basic 4 para Windows 95. Gary Cornell. McGraw-Hill. España 1997.
- [CORNELL, 1998] Cornell, Gary. Aprenda Microsoft Visual Basic Script Ya. Microsoft Press. USA 1998.
- [DELANEY, 1999] Delaney, Kalen. Inside Microsoft Sql Server 7.0 (Mps). Microsoft Press. USA 1999.
- [ERICSSON, 1998] Ericsson. Operation & Maintenance for Consono MD-110. Ericsson. USA 1998.
- [FRANCIS, 1998] Francis, Brian. Professional Active Server Pages 2.0. Worx Press. Reino Unido 1998.
- [GUREWICH, 1998] Gurewich, Nathan. Aprendiendo Visual Basic en 21 días. Prentice Hall. México 1998.
- [HARRINGTON, 1998] Harrington, Jan L. Relational Database Design Clearly Explained. Academic Press/Morgan Kaufmann. USA 1998.

-
- [HEB, 1999] SUPERMERCADOS INTERNACIONALES H.E.B., S.A. DE C.V. La compañía. En: http://home.hebmex.com/pages/La_compania.htm. México 1999.
- [HILLIER, 1998] Hillier, Scott y Mezick, Daniel. Programación de Active Server Pages. McGraw Hill. España 1998.
- [JUMES, 1998] Jumes, James G. Microsoft Windows Nt 4.0 Security, Audit, and Control (Microsoft Technical Reference). Microsoft Press. USA 1998.
- [KERNIGHAN, 1998] Kernighan, Brian W.. Timing Trials: Experiments with Scripting and User-Interface Languages. En: <http://cm.bell-labs.com/cm/cs/who/bwk/interps/pap.html>. Lucent Technologies. USA 1998.
- [LADD, 1999] Ladd, Eric. Using HTML 4, Xml, and Java 1.2. MacMillan. USA 1999.
- [LEMAY, 1999] Lemay, Laura. Aprendiendo Java 2 en 21 días. Prentice-Hall. México 1999.
- [MARTIN, 1991] Martin, James. Rapid Application Development. MacMillan. USA 1991.
- [McCONNELL, 1996] McConnell, Steve C. Rapid Development : Taming Wild Software Schedules. Microsoft Press. USA 1996.
- [McMAHON 1999] McMahon, David. Rapid Application Development With Visual Basic 6. McGraw Hill. USA 1999.
- [MASO, 1998] Maso, Brian. Manual de Visual Interdev. McGraw Hill. México 1998.
- [ORFALI, 1998] Orfali, Robert. Client/Server Programming with Java and CORBA. John Wiley & Sons. USA 1998.
- [RAMALHO, 1999] Ramalho, José Antonio. Learn MS SQL Server 7.0. Wordware Publishing. USA 1999.
- [SERVATI, 1998] Servati, Al. La Biblia de Intranet. McGraw Hill. México 1998.
-

- [TEOREY, 1998] Teorey, Toby J. Database Modeling & Design. Academic Press/Morgan Kaufmann. USA 1998.
- [SURVEYER, 1999] Surveyer, Jacques. Java and Visual Basic: programming's new breed. En: <http://www.tcp.ca/1999/9906/software/javavb/javavb.html>. Canada Computer Paper Inc.. Canada 1999.
- [VISUAL MINING, 1996] Visual Mining. Netcharts: Product Overview. Visual Mining. USA 1996.

ANEXO A – INTERFACES

Acceso a Intranet

Sistema de Tanimación de la Red de Voz HEB Mexico - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address http://intranet/hebo/programa/

HEB

Supermercados Internacionales HEB, S.A de C.V.

INTRANET H-E-B

VALIDACION DE USUARIO

Login

Password

Local intranet

Menú de Intranet.

Sistema de Tanimación de la Red de Voz HEB Mexico - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address http://intranet/hebo/programa/Menu.asp

HEB

Menú General

☐ Catálogos - Catálogos -

☐ Reportes - Reportes -

☐ Solicitudes - Solicitudes -

☐ Utilidades - Utilidades -

Local intranet

Catálogo de Opciones del Sistema.

Actualizador de Opciones del Sistema - Microsoft Internet Explorer

http://Arcaenet/Pedidos/programas/dic_opcion.asp

H-E-B

Catálogo de Opciones del Sistema

Clave de la Opción: - Opciones -

Descripción:

Tipo de Opción: - Tipos de Opción -

Agrupación: - Agrupaciones -

Programa:

Catálogo de Perfiles.

Actualizador de Perfiles - Microsoft Internet Explorer

http://Arcaenet/Pedidos/programas/Mis_perfil.asp

H-E-B

Catálogo de Perfiles

Clave del Perfil: - Perfiles -

Descripción:

Opciones del Sistema

Opción	Tipo de Opción	Agrupación
☐ Catálogo de Conmutadores	Interno	Catálogo
☐ Catálogo de Tipos de Extensiones	Interno	Catálogo
☐ Catálogo de Tipos de Llamadas	Interno	Catálogo
☐ Catálogo de Clases de Llamadas	Interno	Catálogo
☐ Tipos de Solicitudes	Interno	Solicitudes
☐ Generación de Solicitudes de Serv.	Interno	Solicitudes
☐ Telefonos Permitidos	Interno	Entorno
☐ Plan Interno de Solicitudes	Interno	Entorno

Catálogo de Usuarios.

The screenshot shows a web browser window titled 'Actualizador de Usuarios - Microsoft Internet Explorer'. The address bar displays 'http://internet/hecho/programa/abc_usuario.asp'. The page content features the 'H-E-B' logo and the title 'Catálogo de Usuarios'. Below the title, there is a form with the following fields and dropdown menus: 'Clave del Usuario' (with a dropdown menu showing '- Usuarios -'), 'Password', 'Confirmación de Password', 'Sociedad' (with a dropdown menu showing '- Socios -'), and 'Perfil' (with a dropdown menu showing '- Perfiles -'). At the bottom of the form, there are four buttons: 'Consultar', 'Limpiar', 'Inventar', and '<< Registrar'.

Catálogo de Conmutadores.

The screenshot shows a web browser window titled 'Actualizador de Conmutadores - Microsoft Internet Explorer'. The address bar displays 'http://internet/hecho/programa/abc_con.asp'. The page content features the 'H-E-B' logo and the title 'Catálogo de Conmutadores'. Below the title, there is a form with the following fields and dropdown menus: 'Clave del Conmutador' (with a dropdown menu showing '- Conmutadores -') and 'Descripción'. At the bottom of the form, there are four buttons: 'Consultar', 'Limpiar', 'Inventar', and '<< Registrar'.

Catálogo de Tipos de Extensión.

Actualizador de Tipos de Extensión Microsoft Internet Explorer

http://Antares/Procesos/programas/Mtc_tipoext.asp

H-E-B

Catálogo de Tipos de Extensión

Clave del Tipo de Extensión:

Descripción:

Catálogo de Tipos de Llamadas

Actualizador de Tipos de Llamadas Microsoft Internet Explorer

http://Antares/Procesos/programas/Grupos_TipoLlam.asp

H-E-B

Catálogo de Tipos de Llamadas

Clave del Tipo de Llamada:

Descripción:

Prefijo:

Tipo de Extracción de Clave de Llamada:

Catálogo de Claves de Larga Distancia.

Actualizador de Claves de Larga Distancia Microsoft Internet Explorer

http://Armas/hector/programas/la_claveld.html

H-E-B

Catálogo de Claves de Larga Distancia

Tipo de llamada:

Clave de LD:

Descripción:

Catálogo de Áreas Genéricas.

Actualizador de Áreas Genéricas Microsoft Internet Explorer

http://Armas/hector/programas/la_area-gen.asp

H-E-B

Catálogo de Áreas Genéricas

Unidad de Negocio:

Departamento:

Descripción:

Catálogos de Tipos de Solicitud.

Actualizador de Tipos de Solicitud Microsoft Internet Explorer

H-E-B

Catálogo de Tipos de Solicitud

Clave del Tipo de Solicitud: - Tipos de Solicitud -

Descripción:

Generación de Solicitud de Servicio.

H-E-B

Solicitud de Servicio Telefónico

Clave de la Solicitud:

Socio que solicita: - Socios -

Commutador o Área: - Commutador o Área -

Extensión:

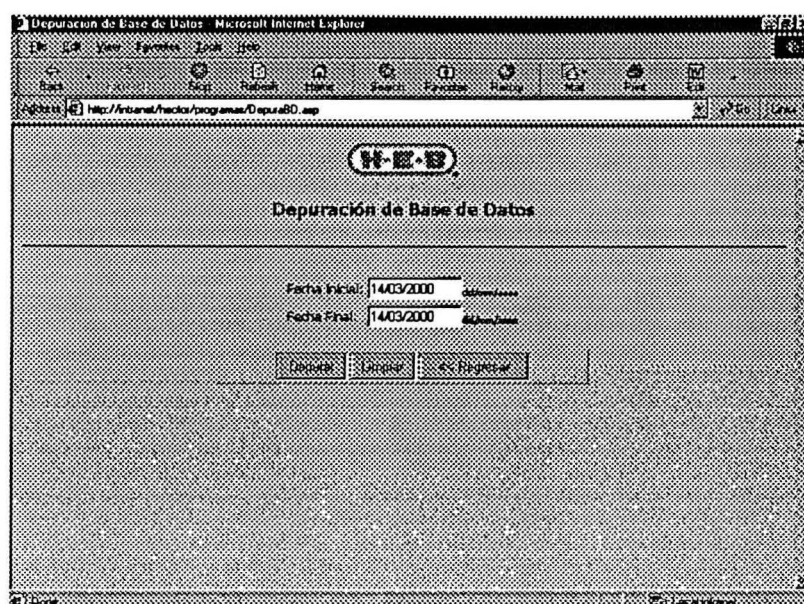
Número de Hoja (acuf):

Código de Costo:

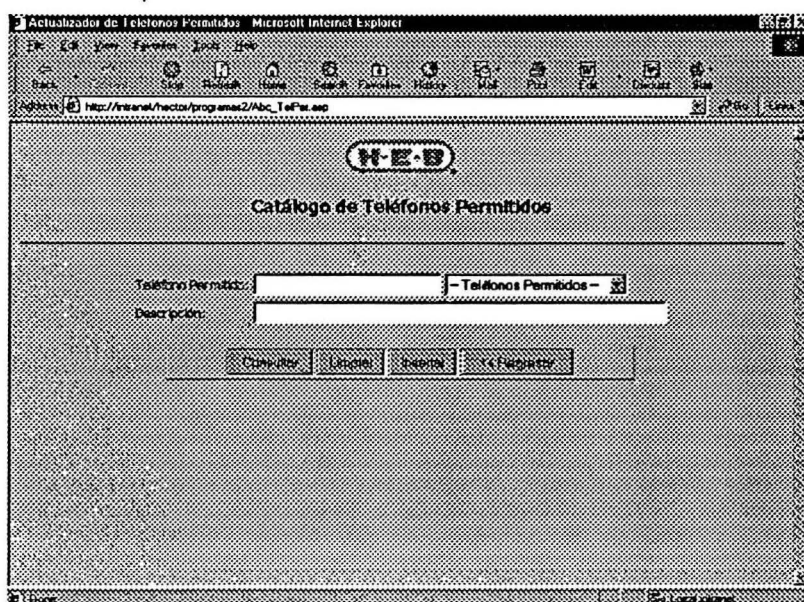
Motivo de la Solicitud:

ACTIVACIONES ☐				CAMBIOS ☐	
☐ Línea Telefónica	☐ Analógico	☐ Directo	☐ Código de Autorización	Código Área:	
	☐ Digital	☐ Ensamblado		Clase de Servid:	☐ Local ☐ Nacional ☐ Internacional
☐ Número en Display:	(Presionar de guita y poner lista de servicio público)		☐ Número en Display:	(Presionar de guita y poner lista de servicio público)	

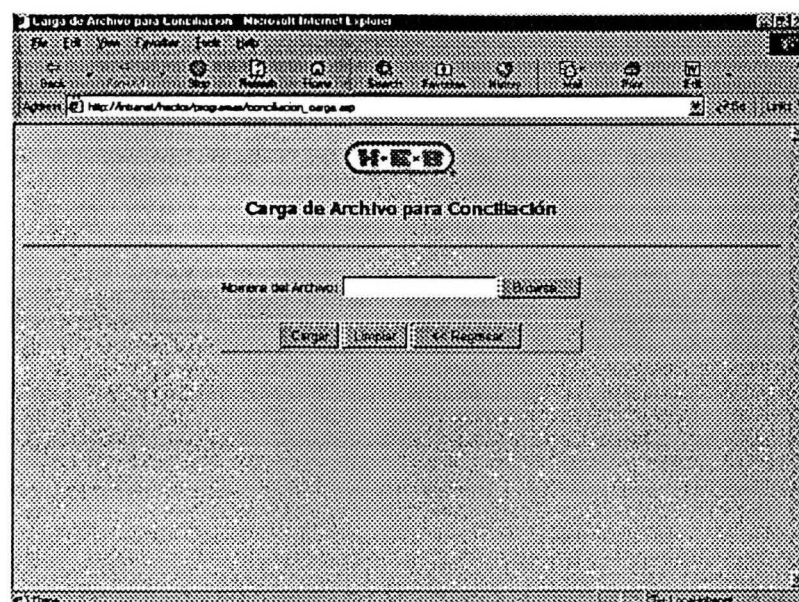
Depuración de la Base de Datos.



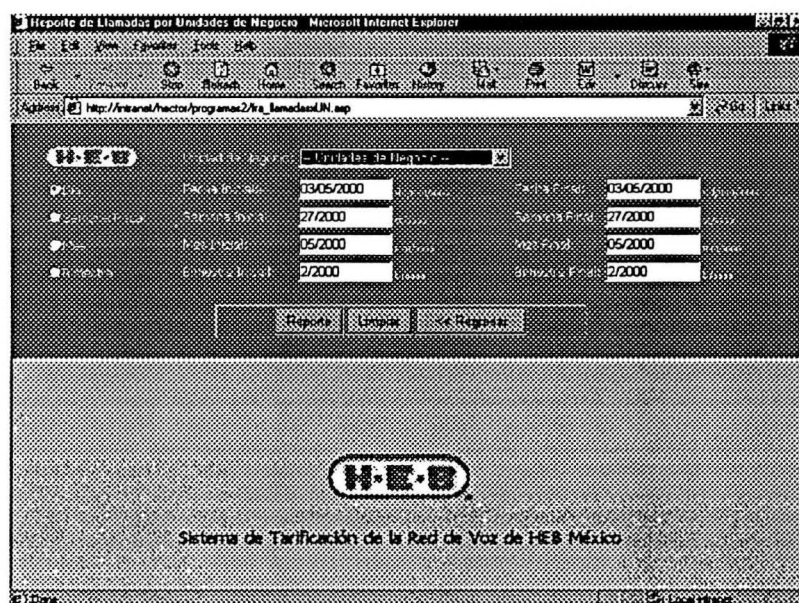
Teléfonos Permitidos.



Conciliación de Llamadas.



Reportes – Cuerpo.



Reportes – Unidad de Negocio.

Reporte de Llamadas por Unidades de Negocio - Microsoft Internet Explorer

Url: http://intranet/factos/programas2/ra_llamadasUN.asp

Unidad de Negocio: **Todas**

Fecha Inicio: 03/05/2000 Fecha Fin: 03/05/2000

Periodo: 27/2000

Fecha Inicio: 05/2000 Fecha Fin: 05/2000

Periodo: 2/2000

Reporte Limpio Regresar

Reporte de Llamadas por Unidades de Negocio

De: 01/05/2000 Hasta: 31/05/2000

Unidad de Negocio	Nombre de Servicio	No. Llamadas	Minutos	Importe Total
SALA - Centro de Informacion	SALA DE INFORMACION	5	12.00	7.50
	SALA DE INFORMACION	1	8.00	3.00
SALA - Apoyo Comercial	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
SALA - Apoyo Comercial	SALA DE INFORMACION	15	33.00	11.25
	SALA DE INFORMACION	15	33.00	11.25

Reportes - Departamentos.

Reporte de Llamadas por Unidades de Negocio - Microsoft Internet Explorer

Url: http://intranet/factos/programas2/ra_llamadasUN.asp

Unidad de Negocio: **Todas**

Fecha Inicio: 03/05/2000 Fecha Fin: 03/05/2000

Periodo: 27/2000

Fecha Inicio: 05/2000 Fecha Fin: 05/2000

Periodo: 2/2000

Reporte Limpio Regresar

Reporte de Llamadas por Unidades de Negocio (Detalle)

De: 01/05/2000 Hasta: 31/05/2000

Unidad de Negocio	Nombre de Servicio	No. Llamadas	Minutos	Importe Total
SALA - Centro de Informacion	SALA DE INFORMACION	5	12.00	7.50
	SALA DE INFORMACION	1	8.00	3.00
SALA - Apoyo Comercial	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
	SALA DE INFORMACION	3	11.00	6.25
SALA - Apoyo Comercial	SALA DE INFORMACION	15	33.00	11.25
	SALA DE INFORMACION	15	33.00	11.25

Reportes - Socios.

Reporte de Llamadas por Unidades de Negocio Microsoft Internet Explorer

http://arbatel/hecda/programas2/Rel_LlamadasLN.asp

Reporte de Llamadas por Unidades de Negocio

Desde: 03/05/2000 Hasta: 03/05/2000

Reporte de Llamadas por Unidades de Negocio (Detalle Departamentos)

De: 01/05/2000 Hasta: 31/05/2000

Cuenta LN	Unidad de Negocio	Cuenta Depar.	Departamento
TSIST	Sistema (58-43)	8121	Sistema de Información

Cuenta Propietario	Nombre	Cuenta de Servicio	No. Llamadas	Módulos	Reporte Total
8102	ELLAS VEDALARI MONICA	58023 58514/526 58137 581	2	4.08	7.08
8107	FERNANDEZ GONZALEZ GLORIA PATRICIA	58074 58137 581 581	2	1.48	2.96
		58075 58137 581 581	5	22.88	5.88

Reportes - Teléfonos.

Reporte de Llamadas por Unidades de Negocio Microsoft Internet Explorer

http://arbatel/hecda/programas2/Rel_LlamadasLN.asp

Reporte de Llamadas por Unidades de Negocio

Desde: 03/05/2000 Hasta: 03/05/2000

Reporte de Llamadas por Unidades de Negocio (Detalle de Extensión)

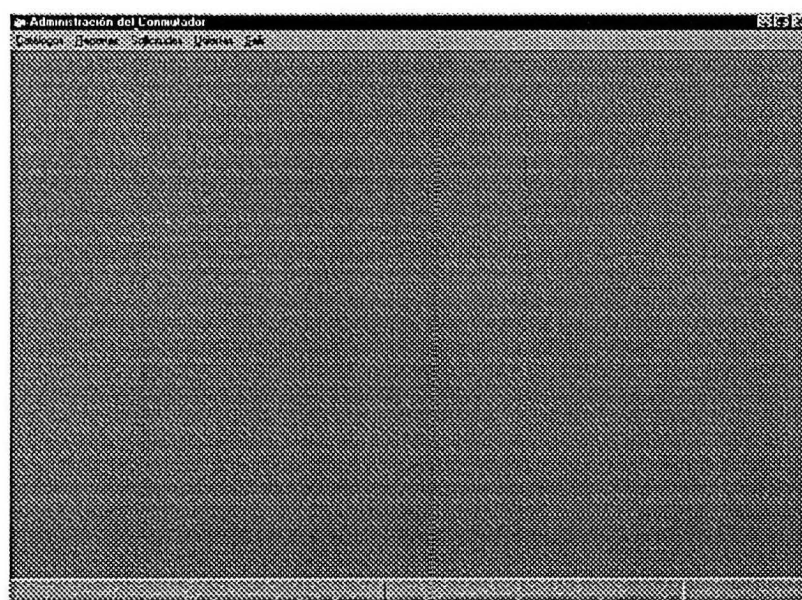
De: 01/05/2000 Hasta: 31/05/2000

Cuenta LN	Unidad de Negocio	Cuenta Depar.	Departamento	Cuenta Propietario	Propietario
TSIST	Sistema (58-43)	8121	Sistema de Información	8655	MEIRA PEREZ WILLIAM

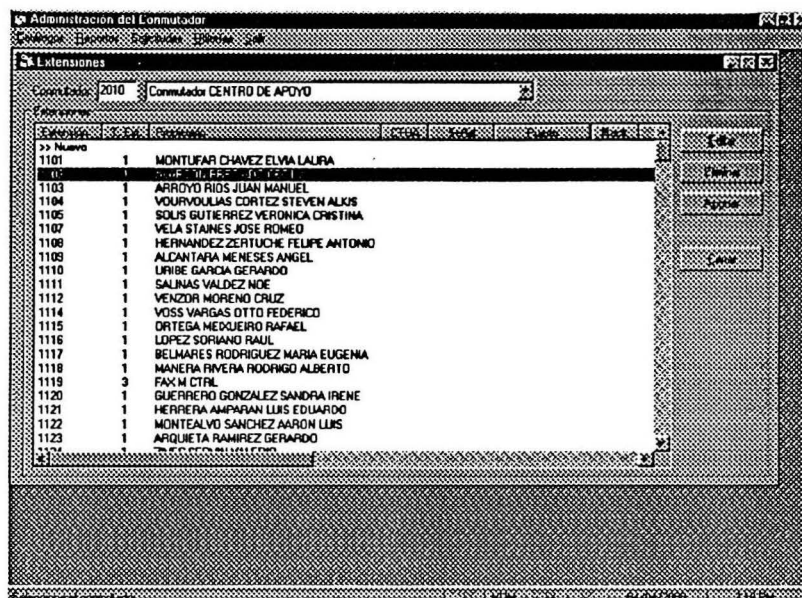
Cuenta de Servicio	Unidad	Nombre	No. Llamadas	Módulos	Reporte Total
58023 58514/526 58137 581	TSIST	MEIRA PEREZ WILLIAM	2	2.08	2.08
58074 58137 581 581	TSIST	MEIRA PEREZ WILLIAM	2	1.48	2.96
58075 58137 581 581	TSIST	MEIRA PEREZ WILLIAM	5	22.88	5.88

Directorio de Extensiones.

216

Acceso a Visual Basic.Menú de Visual Basic.

Actualización de Información de Extensiones.



Detalle de Extensión.

Datos de Extensión

Conmutador: 2010 - Conmutador CENTRO DE APOYO

Extensión: 1102

CTUA: Serial:

Puerto:

Rack: Regleta:

Fila: Par:

Raiser Cuarto:

Raiser:

Color Raiser:

Lugar Físico:

Tipo Teléfono:

Nombre Display:

Aceptar Cancelar

Asignación de Extensión.

The screenshot shows a dialog box titled "Asignación de Extensión". It contains the following fields and controls:

- Commutador:** A text box containing "2010 - Commutador CENTRO DE APOYO".
- Extensión:** A text box containing "1102".
- Tipo de Extensión:** A dropdown menu with "1" selected, and a second dropdown menu showing "EXTENSION DE SOCIO".
- Socio:** A text box containing "0006", and a second dropdown menu showing "ALARCON PRECIADO CECILIA".
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

Administración de Solicitudes.

The screenshot shows a window titled "Administración del Commutador" with a menu bar (Archivo, Edición, Formato, Herramientas, Ventana, Ayuda). Inside, a dialog box titled "Solicitudes de Servicio" is open. The dialog has a "Tipo de Solicitud:" dropdown set to "Activaciones". Below this is a table with the following data:

No. Sol.	Socio	Fecha	Estado
20	LE SALAS ALARCON	1-1-2010	

To the right of the table are buttons: "Nuevo", "Editar", "Eliminar", "Cancelar", and "Cerrar".

Detalle de Solicitud.

Detalle de Solicitud [X]

No. de Solicitud: 53 Tipo de Solicitud: Autorizaciones

Conmutador: Commutador HER SALTILLO Extensión:

Socio: DE SANTOS AVILA JOAQUIN

Plan, Node: 12 Centro de Costos:

Fecha: 15/04/2000 Estado: Nuevo

Motivo:

Línea Telefónica:

☐ Línea Telefónica ☐ Analógica ☐ Digital ☐ Extensión

Nombre Display:

☒ Código de Autorización: ☐ Local ☐ Nacional ☒ Internacional

Fecha Inicio de Labores: 10/04/2000

Aceptar

Actualización de Información en el Conmutador.

The screenshot shows the 'Administración del Computador' (Computer Management) window. The left pane lists system components like Dispositivos y recursos, Administradores de dispositivos, Servicios de almacenamiento, etc. The right pane displays the 'Actualización de información en Computador' (Update Computer Information) dialog box.

Actualización de información en Computador

Computadora:	HP-Compaq 16.0 C3950N-DE	EIPLite
Dirección:	142 Zog 13a 25A	Cg-Lima
Correo:	11-15	Agrupar
Tipo de conexión:	N/A	Descartar
Fuente:	1-2-10-18	Eliminar
IDU:	1-2-10-18	Código
Comentarios:	1	Detener
Trabaja cuando:		
Grupo de compatibilidad de hardware:		
Grupo de la conexión:		
Grupo de no trabajar:		
Número de control de dispositivo:		

Activar Extensión.

Activar Extensión

Extensiones Libres	EQL's Analógicos	EQL's Digitales
1001	001-2-30-11	001-0-20-14
1009	001-2-30-12	001-0-20-15
1010	001-2-30-13	
1013	001-2-30-14	
1014	001-2-30-15	
1015		
1016		
1023		
1026		

☒ Línea Analógica ☐ Línea Digital
☒ Tel. Normal ☐ Tel. Ejecutivo

Categoría:

Aceptar Cancelar

ANEXO B – CÓDIGO DEL SISTEMA

Stored Procedures

```

***** Object: Stored Procedure dbo.spRegPropietarioExtension Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spRegPropietarioExtension
    (@CommutadorID smallint,
    @Extension smallint)
AS
    *****
    / Nombre (Fisico): spRegPropietarioExtension /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: /
    / Descripción: Muestra el propietario de la extensión /
    *****

DECLARE @TipoExtensionID smallint
SET NOCOUNT ON

SELECT @TipoExtensionID = TipoExtensionID
FROM Extension
WHERE CommutadorID = @CommutadorID
AND Extension = @Extension

SET NOCOUNT OFF

IF @TipoExtensionID = 1
    SELECT
        B.SocioID,
        @TipoExtensionID as TipoExtensionID
    FROM
        WHERE B.CommutadorID = @CommutadorID
        AND B.Extension = @Extension
IF @TipoExtensionID = 2
    SELECT
        A.UnegocioID,
        A.DependienteID,
        @TipoExtensionID as TipoExtensionID
    FROM
        Extension_Departamento A
        WHERE A.CommutadorID = @CommutadorID
        AND A.Extension = @Extension
IF @TipoExtensionID = 3
    SELECT
        A.AreaGenericaID,
        @TipoExtensionID as TipoExtensionID
    FROM
        Extension_AreaGenerica A
        WHERE A.CommutadorID = @CommutadorID
        AND A.Extension = @Extension
IF @TipoExtensionID > 3 or @TipoExtensionID < 1 or @TipoExtensionID = NULL
    SELECT NULL as TipoExtensionID
GO

***** Object: Stored Procedure dbo.spAssigneExtension Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spAssigneExtension
    (@CommutadorID smallint,
    @Extension smallint,
    @TipoExtensionID smallint,
    @Cave1 varchar(11),
    @Cave2 varchar(11))
AS
    *****
    / Nombre (Fisico): spAssigneExtension /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: /
    / Descripción: Asignar la extensión a un propietario. /
    *****

SET NOCOUNT ON

DELETE FROM Extension_Socio
WHERE CommutadorID = @CommutadorID
AND Extension = @Extension

DELETE FROM Extension_Departamento
WHERE CommutadorID = @CommutadorID
AND Extension = @Extension

DELETE FROM Extension_AreaGenerica
WHERE CommutadorID = @CommutadorID
AND Extension = @Extension

UPDATE Extension
SET TipoExtensionID = @TipoExtensionID
WHERE CommutadorID = @CommutadorID
AND Extension = @Extension

SET NOCOUNT OFF

IF @TipoExtensionID = 1
    BEGIN
        INSERT INTO Extension_Socio (CommutadorID, Extension, SocioID)
        VALUES (@CommutadorID, @Extension, @Cave1)
        IF @@RowCount > 0
            SELECT Resultado = 1
        ELSE
            SELECT Resultado = 0
    END

IF @TipoExtensionID = 2
    BEGIN
        INSERT INTO Extension_Departamento (CommutadorID, Extension, UnegocioID,
        DependienteID)
        VALUES (@CommutadorID, @Extension,
        CONVERT(VARCHAR(5),@Cave1), CONVERT(VARCHAR(10),@Cave2))
        IF @@RowCount > 0
            SELECT Resultado = 1
        ELSE
            SELECT Resultado = 0
    END

IF @TipoExtensionID = 3
    BEGIN
        INSERT INTO Extension_AreaGenerica (CommutadorID, Extension, AreaGenericaID)
        VALUES (@CommutadorID, @Extension,
        CONVERT(smallint,@Cave1))
        IF @@RowCount > 0
            SELECT Resultado = 1
        ELSE
            SELECT Resultado = 0
    END
GO

***** Object: Stored Procedure dbo.spAutorizaSolicitud Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spAutorizaSolicitud
    (@SolicitudID int,
    @TipoSolicitud smallint)
AS
    *****
    / Nombre (Fisico): spAutorizaSolicitud /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 20 de Abril 2000 /
    / Proyecto: /
    / Descripción: /
    *****

UPDATE Solicitud
SET Estado = 2
WHERE SolicitudID = @SolicitudID
AND Tipo_SolicitudID = @TipoSolicitudID
IF @@RowCount > 0
    SELECT Resultado = 1
ELSE
    SELECT Resultado = 0
GO

***** Object: Stored Procedure dbo.spBuscaCaveID Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spBuscaCaveID
    (@Telefono varchar(20),
    @TipoLlamadaID smallint)
AS
    *****
    / Nombre (Fisico): spBuscaCaveID /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 1 de Abril 2000 /
    / Proyecto: /
    / Descripción: Busca la clave lada del telefono enviado /
    *****

DECLARE
    @TipoExtremLade smallint,
    @Preajo varchar(10),
    @Tel_proc varchar(20),
    @CaveLadeID int

SELECT
    @TipoExtremLade = 0,
    @Preajo = (null),
    @Tel_proc = Ltrim(@Telefono),
    @CaveLadeID = (null)

SELECT
    @TipoExtremLade = TipoExtremLade,
    @Preajo = Preajo,
    Tipo_Llamada
FROM
    Tipo_Llamada
WHERE
    TipoLlamadaID = @TipoLlamadaID

IF @TipoExtremLade > 0 AND ISNULL(@TipoExtremLade,-1) < -1
    IF @TipoExtremLade = 1
        BEGIN
            SELECT
                @Tel_proc = DATALENGTH(@Tel_proc) * 1,
                DATALENGTH(@Preajo),
                @CaveLadeID = 1
            WHILE ISNULL(@CaveLadeID,-1) = -1 AND @CaveLadeID >= 1
            BEGIN
                SELECT @CaveLadeID = CaveLadeID
                FROM CaveLade
                WHERE TipoLlamadaID = @TipoLlamadaID
                AND Cave = SUBSTRING(@Tel_proc,1,@CaveLadeID)
            END
            SELECT @CaveLadeID = @CaveLadeID - 1
        END
    IF @TipoExtremLade = 2
        BEGIN
            SELECT
                @Tel_proc = DATALENGTH(@Tel_proc) * 1,
                DATALENGTH(@Preajo),
                @CaveLadeID = 1
            WHILE ISNULL(@CaveLadeID,-1) = -1 AND @CaveLadeID >= 1
            BEGIN
                SELECT @CaveLadeID = CaveLadeID
                FROM CaveLade
                WHERE TipoLlamadaID = @TipoLlamadaID
                AND Cave = SUBSTRING(@Tel_proc,1,3)
            END
        END
    SELECT
        @TipoLlamadaID as TipoLlamadaID,
        @CaveLadeID as CaveLadeID
GO

***** Object: Stored Procedure dbo.spCancelaSolicitud Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spCancelaSolicitud
    (@SolicitudID int,
    @TipoSolicitud smallint)
AS
    *****
    / Nombre (Fisico): spCancelaSolicitud /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: /
    / Descripción: Cambia el estado de la solicitud para Cancelarla /
    *****

UPDATE Solicitud
SET Estado = 4
WHERE SolicitudID = @SolicitudID
AND Tipo_SolicitudID = @TipoSolicitudID
IF @@RowCount > 0
    SELECT Resultado = 1
ELSE
    SELECT Resultado = 0
GO

***** Object: Stored Procedure dbo.spClasificaTipoLlamada Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spClasificaTipoLlamada
    (@Telefono VARCHAR(20))
AS
    *****
    / Nombre (Fisico): spClasificaTipoLlamada /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 20 de Abril 2000 /
    / Proyecto: /
    / Descripción: Identifica el tipo de llamada del telefono enviado /
    *****

DECLARE @Tel_proc VARCHAR(20),
    @TipoLlamadaID SMALLINT,
    @Int INT

SELECT
    @Tel_proc = Ltrim(@Telefono),

```

```

    / Fecha: 20 de Abril 2000 /
    / Proyecto: /
    / Descripción: /
    / Actualiza el estado de la solicitud para autorizarla /

UPDATE Solicitud
SET Estado = 2
WHERE SolicitudID = @SolicitudID
AND Tipo_SolicitudID = @TipoSolicitudID
IF @@RowCount > 0
    SELECT Resultado = 1
ELSE
    SELECT Resultado = 0
GO

***** Object: Stored Procedure dbo.spBuscaCaveID Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spBuscaCaveID
    (@Telefono varchar(20),
    @TipoLlamadaID smallint)
AS
    *****
    / Nombre (Fisico): spBuscaCaveID /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 1 de Abril 2000 /
    / Proyecto: /
    / Descripción: Busca la clave lada del telefono enviado /
    *****

DECLARE
    @TipoExtremLade smallint,
    @Preajo varchar(10),
    @Tel_proc varchar(20),
    @CaveLadeID int

SELECT
    @TipoExtremLade = 0,
    @Preajo = (null),
    @Tel_proc = Ltrim(@Telefono),
    @CaveLadeID = (null)

SELECT
    @TipoExtremLade = TipoExtremLade,
    @Preajo = Preajo,
    Tipo_Llamada
FROM
    Tipo_Llamada
WHERE
    TipoLlamadaID = @TipoLlamadaID

IF @TipoExtremLade > 0 AND ISNULL(@TipoExtremLade,-1) < -1
    IF @TipoExtremLade = 1
        BEGIN
            SELECT
                @Tel_proc = DATALENGTH(@Tel_proc) * 1,
                DATALENGTH(@Preajo),
                @CaveLadeID = 1
            WHILE ISNULL(@CaveLadeID,-1) = -1 AND @CaveLadeID >= 1
            BEGIN
                SELECT @CaveLadeID = CaveLadeID
                FROM CaveLade
                WHERE TipoLlamadaID = @TipoLlamadaID
                AND Cave = SUBSTRING(@Tel_proc,1,@CaveLadeID)
            END
            SELECT @CaveLadeID = @CaveLadeID - 1
        END
    IF @TipoExtremLade = 2
        BEGIN
            SELECT
                @Tel_proc = DATALENGTH(@Tel_proc) * 1,
                DATALENGTH(@Preajo),
                @CaveLadeID = 1
            WHILE ISNULL(@CaveLadeID,-1) = -1 AND @CaveLadeID >= 1
            BEGIN
                SELECT @CaveLadeID = CaveLadeID
                FROM CaveLade
                WHERE TipoLlamadaID = @TipoLlamadaID
                AND Cave = SUBSTRING(@Tel_proc,1,3)
            END
        END
    SELECT
        @TipoLlamadaID as TipoLlamadaID,
        @CaveLadeID as CaveLadeID
GO

***** Object: Stored Procedure dbo.spCancelaSolicitud Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spCancelaSolicitud
    (@SolicitudID int,
    @TipoSolicitud smallint)
AS
    *****
    / Nombre (Fisico): spCancelaSolicitud /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: /
    / Descripción: Cambia el estado de la solicitud para Cancelarla /
    *****

UPDATE Solicitud
SET Estado = 4
WHERE SolicitudID = @SolicitudID
AND Tipo_SolicitudID = @TipoSolicitudID
IF @@RowCount > 0
    SELECT Resultado = 1
ELSE
    SELECT Resultado = 0
GO

***** Object: Stored Procedure dbo.spClasificaTipoLlamada Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spClasificaTipoLlamada
    (@Telefono VARCHAR(20))
AS
    *****
    / Nombre (Fisico): spClasificaTipoLlamada /
    / Autor: Hctor Manuel Gerza Gerza /
    / Fecha: 20 de Abril 2000 /
    / Proyecto: /
    / Descripción: Identifica el tipo de llamada del telefono enviado /
    *****

DECLARE @Tel_proc VARCHAR(20),
    @TipoLlamadaID SMALLINT,
    @Int INT

SELECT
    @Tel_proc = Ltrim(@Telefono),

```



```

@TipoLlamadaID = (NULL),
@ = 1

WHILE ISNULL(@TipoLlamadaID, -1) = -1 AND @ <= Datalength(@Tel_Proc)
BEGIN
    SELECT @TipoLlamadaID = TipoLlamadaID
    FROM Tipo_Llamada
    WHERE Preijo = SUBSTRING(@Tel_Proc, 1, @)

    SELECT @ = @ + 1
END

IF ISNULL(@TipoLlamadaID, -1) = -1
    SELECT @TipoLlamadaID = TipoLlamadaID
    FROM Tipo_Llamada
    WHERE Preijo IS NULL

SELECT @TipoLlamadaID as TipoLlamadaID
GO

***** Object: Stored Procedure dbo.spConciliacionCarga Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spConciliacionCarga
    (@Path varchar(100))
AS
    / Nombre (Pais): spConciliacionCarga /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 1 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / Descripción: Carga el archivo de conciliacion /
    / *****

    declare @Comando varchar(255)

    SELECT @Comando = 'top TARIFICACION_PasoConciliacion in ' + @Path + ' -Nhgaz/pasobcp.int -
    oNhgaz/pasobcp.del -Use -P -S.Bolobol'

    --EXEC master..xp_cmdshell 'top TARIFICACION_PasoConciliacion in Nhgaz/pasobcp.int -
    Nhgaz/pasobcp.del -Use -P -S.Bolobol'

    SET NOCOUNT ON

    DELETE FROM PasoConciliacion

    EXEC master..xp_cmdshell @Comando

    SET NOCOUNT OFF

    INSERT INTO Llamada_Telmx
    (Fecha,
    Origen,
    Destino,
    Duracion,
    Importe,
    Pob_Destino,
    Fecha_Factura)
    SELECT
        DATEADD(mi, CONVERT(INT, substring(A.Linea, 70, 2)), DATEADD(hi, CONVERT(INT, subse
        ing(A.Linea, 68, 2)), CONVERT(datetime, '200' + substring(A.Linea, 30, 5), 112)),
        SUBSTRING(A.Linea, 8, 7),
        SUBSTRING(A.Linea, 71, 7),
        CONVERT(decimal(10, 2), SUBSTRING(A.Linea, 30, 3)),
        CONVERT(decimal(15, 2), SUBSTRING(A.Linea, 42, 8)),
        SUBSTRING(A.Linea, 51, 17),
        CONVERT(SMALLDATETIME, '20' + SUBSTRING(A.Linea, 81, 4) + ' 01', 112)

    FROM PasoConciliacion A

    GO

***** Object: Stored Procedure dbo.spConciliacionInformacion Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spConciliacionInformacion
AS
    / Nombre (Pais): spConciliacionInformacion /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / Descripción: Muestra informacón específica para la /
    / conciliacion /
    / *****

    SELECT
        COUNT(*) as Cantidad,
        CONVERT(varchar(10), MIN(fecha), 103) as FechaI,
        CONVERT(varchar(10), MAX(fecha), 103) as FechaF

    FROM Llamada
    WHERE Llamada_TelmxID = NULL

    GO

***** Object: Stored Procedure dbo.spConciliacionLlamadas Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spConciliacionLlamadas
    (@Fin varchar(10),
    @FFin varchar(10))
AS
    / Nombre (Pais): spConciliacionLlamadas /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 20 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / Descripción: Muestra las llamadas por conciliar /
    / *****

    declare @FechaIni datetime,
    @FechaFin datetime

    SELECT
        @FechaIni = CONVERT(datetime, @Fin, 103),
        @FechaFin = DATEADD(day, 1, CONVERT(datetime, @FFin, 103))

    SELECT *
    INTO #Llamada_Telmx
    FROM Llamada_Telmx
    WHERE Fecha >= @FechaIni
    AND Fecha < @FechaFin

    UPDATE A
    SET A.Llamada_TelmxID = B.Llamada_TelmxID
    FROM Llamada A, #Llamada_Telmx B
    WHERE ISNULL(A.Llamada_TelmxID, -1) = -1
    AND A.Telefono = B.Telefono
    AND A.Fecha >= DATEADD(mi, -1, B.Fecha)
    AND A.Fecha <= DATEADD(mi, 1, B.Fecha)
    AND (A.Duracion / 60) >= B.Duracion - 1
    AND (A.Duracion / 60) <= B.Duracion + 1

    /

    -- Si la fecha es cuando inicio la llamada
    UPDATE A

```

```

SET A.Llamada_TelmxID = B.Llamada_TelmxID
FROM Llamada A, #Llamada_Telmx B
WHERE ISNULL(A.Llamada_TelmxID, -1) = -1
AND A.Telefono = B.Telefono
AND A.Fecha >= DATEADD(mi, -1, DATEADD(mi, A.Duracion * -1, B.Fecha))
AND A.Fecha <= DATEADD(mi, 1, DATEADD(mi, A.Duracion * -1, B.Fecha))
AND (A.Duracion / 60) >= B.Duracion - 1
AND (A.Duracion / 60) <= B.Duracion + 1

/
GO

***** Object: Stored Procedure dbo.spDepura Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spDepura
    (@FechaIni varchar(8),
    @FechaFin varchar(8))
AS
    / Nombre (Pais): spDepura /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 1 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / Descripción: Depura las tablas operativas /
    / *****

    declare @FechaIni datetime,
    @FechaFin datetime,
    @Cuenta int

    SELECT @FechaIni = convert(datetime, @FechaIni),
    @FechaFin = convert(datetime, @FechaFin),
    @FechaIni = dateadd(day, 1, @FechaIni),
    @Cuenta = 0

    DELETE FROM Llamada
    WHERE Fecha >= @FechaIni
    AND Fecha <= @FechaFin

    SELECT @Cuenta = @Cuenta + @@RowCount

    DELETE FROM Llamada_Telmx
    WHERE NOT EXISTS (SELECT * FROM Llamada WHERE Llamada.Llamada_TelmxID =
    Llamada_TelmxID)
    AND Fecha >= @FechaIni
    AND Fecha <= @FechaFin

    SELECT @Cuenta = @Cuenta + @@RowCount

    DELETE FROM Solicitud_Camb
    WHERE EXISTS (SELECT *
        FROM Solicitud
        WHERE Solicitud.Solicitud_Id = Solicitud_Id
        AND Solicitud.Tipo_SolicitudID = Tipo_SolicitudID
        AND Fecha >= @FechaIni
        AND Fecha <= @FechaFin)

    SELECT @Cuenta = @Cuenta + @@RowCount

    DELETE FROM Solicitud_Act
    WHERE EXISTS (SELECT *
        FROM Solicitud
        WHERE Solicitud.Solicitud_Id = Solicitud_Id
        AND Solicitud.Tipo_SolicitudID = Tipo_SolicitudID
        AND Fecha >= @FechaIni
        AND Fecha <= @FechaFin)

    SELECT @Cuenta = @Cuenta + @@RowCount

    DELETE FROM Solicitud
    WHERE Fecha >= @FechaIni
    AND Fecha <= @FechaFin

    SELECT @Cuenta = @Cuenta + @@RowCount

    SELECT @Cuenta as Resultado
    GO

***** Object: Stored Procedure dbo.spEliminaAsignacionExt Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spEliminaAsignacionExt
    (@ComunadorID smallint,
    @Extension smallint)
AS
    / Nombre (Pais): spEliminaAsignacionExt /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 8 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / Descripción: Elimina la asignacion de una extension a /
    / un propietario. /
    / *****

    SET NOCOUNT ON

    DELETE FROM Extension_Socio
    WHERE ComunadorID = @ComunadorID
    AND Extension = @Extension

    DELETE FROM Extension_Departamento
    WHERE ComunadorID = @ComunadorID
    AND Extension = @Extension

    DELETE FROM Extension_AreaGenerica
    WHERE ComunadorID = @ComunadorID
    AND Extension = @Extension

    SET NOCOUNT OFF

    UPDATE Extension
    SET TipoExtensionID = NULL
    WHERE ComunadorID = @ComunadorID
    AND Extension = @Extension

    IF @@RowCount > 0
        SELECT Resultado = 1
    ELSE
        SELECT Resultado = 0
    GO

***** Object: Stored Procedure dbo.spTipoSolicitud Script Date: 05/05/2000 8:53:44 PM *****
CREATE PROCEDURE spTipoSolicitud
    (@TipoSolicitud smallint,
    @Descripcion varchar(50))
AS
    / Nombre (Pais): spTipoSolicitud /
    / Autor: Hctor Manuel Garza Gerza /
    / Fecha: 20 de Abril 2000 /
    / Proyecto: / Tarificacion /
    / *****

    IF (SELECT count(*) FROM Tipo_Solicitud WHERE Tipo_SolicitudID = @TipoSolicitudID) <= 0
    BEGIN

```

```

INSERT INTO Tipo_Solicitud
VALUES (@TipoSolicitud, @Descripcion)
IF @@RowCount > 0
ELSE
    SELECT Resultado = 1
END
ELSE
    SELECT Resultado = 0
GO

/***** Object: Stored Procedure dbo.spIniciarUsuario Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spIniciarUsuario
    (@UsuarioID varchar(10),
    @Password varchar(20),
    @PermID int,
    @SocioID varchar(11))
AS
/* Nombre Páico: spIniciarUsuario */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: */
/* Descripción: */

BEGIN
    IF (SELECT count(*) FROM Usuario WHERE UsuarioID = @UsuarioID) <= 0
    BEGIN
        INSERT INTO Usuario (UsuarioID, Password, PermID, SocioID)
        VALUES (@UsuarioID, @Password, @PermID, @SocioID)
        IF @@RowCount > 0
        ELSE
            SELECT Resultado = 1
        END
        ELSE
            SELECT Resultado = 0
    END
    ELSE
        SELECT Resultado = 0
    END
GO

/***** Object: Stored Procedure dbo.spProcesarSolicitud Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spProcesarSolicitud
    (@SolicitudID int,
    @TipoSolicitudID smallint)
AS
/* Nombre Páico: spProcesarSolicitud */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: */
/* Descripción: Actualiza el estatus de la solicitud para identificarla como procesada */

UPDATE Solicitud
SET Estatus = 3
WHERE SolicitudID = @SolicitudID
AND Tipo_SolicitudID = @TipoSolicitudID
IF @@RowCount > 0
ELSE
    SELECT Resultado = 1
END
ELSE
    SELECT Resultado = 0
GO

/***** Object: Stored Procedure dbo.spRepAreaGenerica Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepAreaGenerica
    (@UnigocioID varchar(5),
    @DepartamentoID varchar(10))
AS
/* Nombre Páico: spRepAreaGenerica */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 20 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra las áreas genericas de un depto. */

SELECT AreaGenericaID, Descripcion, UnigocioID, DepartamentoID
FROM AreaGenerica
WHERE UnigocioID = @UnigocioID
AND DepartamentoID = @DepartamentoID
ORDER BY Descripcion
GO

/***** Object: Stored Procedure dbo.spRepClausLD Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepClausLD
    (@TipoLlamadaID smallint)
AS
/* Nombre Páico: spRepClausLD */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra las claves de LD de un tipo de llamada */

SELECT Clave, Descripcion, *
FROM ClausLD
WHERE TipoLlamadaID = @TipoLlamadaID
ORDER BY Descripcion
GO

/***** Object: Stored Procedure dbo.spRepConciliacion Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepConciliacion
    (@FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/* Nombre Páico: spRepConciliacion */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra la conciliación de llamadas */

DECLARE @Fini datetime,
        @FFin datetime

SELECT @Fini = CONVERT(datetime, @FechaIni, 103),
        @FFin = dateadd(dd, 1, CONVERT(datetime, @FechaFin, 103))

SELECT CONVERT(varchar(10), A.Fecha, 103) as Fecha,
        CONVERT(varchar(5), A.Fecha, 8) as Hora,
        A.Telefono,
        CONVERT(decimal(6,2), A.Duracion/60) Duracion,
        A.CommutadorID,
        A.Extension,
        CONVERT(varchar(10), B.Fecha, 103) as FechaT,
        CONVERT(varchar(5), B.Fecha, 8) as HoraT,
        B.Destino as TelefonoT,
        CONVERT(decimal(6,2), B.Duracion/60) as DuracionT,
        CONVERT(decimal(10,2), B.Importe) as ImporteT
FROM Llamada A,
     Llamada_Telmax B
WHERE A.Fecha >= @Fini
AND A.Fecha <= @FFin
AND A.Llamada_TelmaxID = B.Llamada_TelmaxID
ORDER BY A.Fecha
GO

/***** Object: Stored Procedure dbo.spRepConciliacionLlamadas Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepConciliacionLlamadas
    (@FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/* Nombre Páico: spRepConciliacionLlamadas */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 20 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra llamadas que no se conciliaron */

DECLARE @Fini datetime,
        @FFin datetime

SELECT @Fini = CONVERT(datetime, @FechaIni, 103),
        @FFin = dateadd(dd, 1, CONVERT(datetime, @FechaFin, 103))

SELECT LlamadaID,
        CONVERT(varchar(10), A.Fecha, 103) as Fecha,
        CONVERT(varchar(5), A.Fecha, 8) as Hora,
        CONVERT(char(20), Telefono) as Telefono,
        CONVERT(char(9), CONVERT(decimal(6,2), Duracion/60)) as Duracion,
        CONVERT(varchar(4), ComutadorID) as ComutadorID,
        CONVERT(varchar(4), Extension) as Descripcion
FROM Llamada
WHERE Fecha >= @Fini
AND Fecha <= @FFin
AND Llamada_TelmaxID = NULL
GO

/***** Object: Stored Procedure dbo.spRepConciliacionLlamadasT Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepConciliacionLlamadasT
    (@FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/* Nombre Páico: spRepConciliacionLlamadasT */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra llamadas de Telmax sin conciliar */

DECLARE @Fini datetime,
        @FFin datetime

SELECT @Fini = CONVERT(datetime, @FechaIni, 103),
        @FFin = dateadd(dd, 1, CONVERT(datetime, @FechaFin, 103))

SELECT Llamada_TelmaxID,
        CONVERT(varchar(10), A.Fecha, 103) as Fecha,
        CONVERT(varchar(5), A.Fecha, 8) as Hora,
        CONVERT(char(20), Destino) as Destino,
        CONVERT(char(9), CONVERT(decimal(6,2), Duracion/60)) as Duracion,
        CONVERT(varchar(13), CONVERT(decimal(10,2), Importe)) as Descripcion
FROM Llamada_Telmax
WHERE Fecha >= @Fini
AND Fecha <= @FFin
AND Llamada_TelmaxID = NULL
GO

/***** Object: Stored Procedure dbo.spRepDirectorio Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepDirectorio
    (@Buscar varchar(100))
AS
/* Nombre Páico: spRepDirectorio */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra las extensiones que coinciden con la búsqueda */

SET NOCOUNT ON

SELECT @Buscar = '%' + @Buscar + '%'

CREATE TABLE #Unigocio (UnigocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint NULL)

INSERT INTO #Unigocio
EXEC EMPRESA..SpAllUnigocio

CREATE TABLE #Unig_Depto (UnigocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Unig_Depto
EXEC EMPRESA..SpAllUnig_Depto

CREATE TABLE #Depto (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

INSERT INTO #Depto
EXEC EMPRESA..SpAllDepto

CREATE TABLE #Socio (UnigocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, Materno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA..SpAllSocio

CREATE TABLE #Ubicacion (UbicacionID smallint NULL, Descripcion varchar(50) NULL, ComutadorID smallint NULL)

INSERT INTO #Ubicacion
EXEC EMPRESA..SpAllUbicacion

SELECT CONVERT(varchar(11), C.DepartamentoID) as Clave,
        CONVERT(varchar(75), C.DescDpt) as Descripcion,

```

```

CONVERT(varchar(10), B.Fecha, 103) as FechaT,
CONVERT(varchar(5), B.Fecha, 8) as HoraT,
B.Destino as TelefonoT,
CONVERT(decimal(6,2), B.Duracion/60) as DuracionT,
CONVERT(decimal(10,2), B.Importe) as ImporteT
Llamada A,
Llamada_Telmax B
WHERE A.Fecha >= @Fini
AND A.Fecha <= @FFin
AND A.Llamada_TelmaxID = B.Llamada_TelmaxID
ORDER BY A.Fecha
GO

/***** Object: Stored Procedure dbo.spRepConciliacionLlamadasT Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepConciliacionLlamadasT
    (@FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/* Nombre Páico: spRepConciliacionLlamadasT */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra llamadas de Telmax sin conciliar */

DECLARE @Fini datetime,
        @FFin datetime

SELECT @Fini = CONVERT(datetime, @FechaIni, 103),
        @FFin = dateadd(dd, 1, CONVERT(datetime, @FechaFin, 103))

SELECT Llamada_TelmaxID,
        CONVERT(varchar(10), A.Fecha, 103) as Fecha,
        CONVERT(varchar(5), A.Fecha, 8) as Hora,
        CONVERT(char(20), Destino) as Destino,
        CONVERT(char(9), CONVERT(decimal(6,2), Duracion/60)) as Duracion,
        CONVERT(varchar(13), CONVERT(decimal(10,2), Importe)) as Descripcion
FROM Llamada_Telmax
WHERE Fecha >= @Fini
AND Fecha <= @FFin
AND Llamada_TelmaxID = NULL
GO

/***** Object: Stored Procedure dbo.spRepDirectorio Script Date: 05/05/2000 8:53:45 PM *****/
CREATE PROCEDURE spRepDirectorio
    (@Buscar varchar(100))
AS
/* Nombre Páico: spRepDirectorio */
/* Autor: Hctor Manuel Garza Gerza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: */
/* Descripción: Muestra las extensiones que coinciden con la búsqueda */

SET NOCOUNT ON

SELECT @Buscar = '%' + @Buscar + '%'

CREATE TABLE #Unigocio (UnigocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint NULL)

INSERT INTO #Unigocio
EXEC EMPRESA..SpAllUnigocio

CREATE TABLE #Unig_Depto (UnigocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Unig_Depto
EXEC EMPRESA..SpAllUnig_Depto

CREATE TABLE #Depto (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

INSERT INTO #Depto
EXEC EMPRESA..SpAllDepto

CREATE TABLE #Socio (UnigocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, Materno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA..SpAllSocio

CREATE TABLE #Ubicacion (UbicacionID smallint NULL, Descripcion varchar(50) NULL, ComutadorID smallint NULL)

INSERT INTO #Ubicacion
EXEC EMPRESA..SpAllUbicacion

SELECT CONVERT(varchar(11), C.DepartamentoID) as Clave,
        CONVERT(varchar(75), C.DescDpt) as Descripcion,

```

```

D.CommutadorID,
E.Descripcion as DescUbl,
D.Extension
#Extensiones
#Negocio A,
#Neg_Depto B,
#Departamento C,
Extension_Departamento D,
#Ubicacion E

WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
AND D.CommutadorID = E.CommutadorID
UNION
SELECT
    CONVERT(varchar(11),C.SocioID) as Clave,
    CONVERT(varchar(73),C.Apellido + ' ' + C.Nombre) as Descripción,
    D.CommutadorID,
    F.Descripcion as DescUbl,
    D.Extension
FROM
    #Negocio A,
    #Neg_Depto B,
    #Socio C,
    Extension_Socio D,
    #Departamento E,
    #Ubicacion F
WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID
AND D.CommutadorID = F.CommutadorID
UNION
SELECT
    CONVERT(varchar(11),C.AreaGenericaID) as Clave,
    CONVERT(varchar(73),C.Descripcion) as Descripción,
    D.CommutadorID,
    F.Descripcion as DescUbl,
    D.Extension
FROM
    #Negocio A,
    #Neg_Depto B,
    AreaGenerica C,
    Extension_AreaGenerica D,
    #Departamento E,
    #Ubicacion F
WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID
AND D.CommutadorID = F.CommutadorID
SET NOCOUNT OFF
SELECT *
FROM #Extensiones
WHERE Descripcion LIKE @Buscar
GO
/***** Object: Stored Procedure dbo.spRepExtensionesCommutador    Script Date: 05/05/2000 8:53:45 PM
*****/
CREATE PROCEDURE spRepExtensionesCommutador
    (@CommutadorID smallint)
AS
    /*-----*/
    /* Nombre Físico:  spRepExtensionesCommutador */
    /* Autor:          Hctor Manuel Garza Garza */
    /* Fecha:          20 de Abril 2000 */
    /* Proyecto:       Tarificación */
    /* Descripción:     Muestra las extensiones de un comunicador */
    /*-----*/
SET NOCOUNT ON
CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #UNegocio
EXEC EMPRESA.SpAllNegocio

CREATE TABLE #UNeg_Depto (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
INSERT INTO #UNeg_Depto
EXEC EMPRESA.SpAllNeg_Depto

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescOpt varchar(50) NULL)
INSERT INTO #Departamento
EXEC EMPRESA.SpAllDepartamento

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, ApellidoM varchar(20) NULL, Mail varchar(50) NULL)
INSERT INTO #Socio
EXEC EMPRESA.SpAllSocio

SELECT
    A.CommutadorID,
    A.Extension,
    A.CTUA,
    A.Signal,
    A.Puerto,
    A.Rack,
    A.Regleta,
    A.File,
    A.Par,
    A.Raiser,
    A.Craiser,
    A.LFisco,
    A.TipoTel,
    A.NombreDisplay,
    A.TipoExtensionID,
    B.Apellido + ' ' + B.Nombre as Propietario
FROM
    #Extensiones
    #Socio B,
    Extension_Socio C
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = 1
AND A.CommutadorID = C.CommutadorID
AND A.Extension = C.Extension
AND C.SocioID = B.SocioID
UNION
SELECT
    A.CommutadorID,
    A.Extension,
    A.CTUA,
    A.Signal,
    A.Puerto,
    A.Rack,
    A.Regleta,
    A.File,
    A.Par,
    A.Raiser,
    A.Craiser,
    A.LFisco,
    A.TipoTel,
    A.NombreDisplay,
    A.TipoExtensionID,
    B.Descripcion as Propietario
FROM
    #Extensiones
    AreaGenerica B,
    Extension_AreaGenerica C
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = 3
AND A.CommutadorID = C.CommutadorID
AND A.Extension = C.Extension
AND C.AreaGenericaID = B.AreaGenericaID
UNION
SELECT
    A.CommutadorID,
    A.Extension,
    A.CTUA,
    A.Signal,
    A.Puerto,
    A.Rack,
    A.Regleta,
    A.File,
    A.Par,
    A.Raiser,
    A.Craiser,
    A.LFisco,
    A.TipoTel,
    A.NombreDisplay,
    A.TipoExtensionID,
    B.Descripcion as Propietario
FROM
    #Extensiones
    Extension A,
    Extension_AreaGenerica B
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = NULL
ORDER BY A.CommutadorID, A.Extension
SET NOCOUNT OFF
SELECT *
FROM #Extensiones
GO
/***** Object: Stored Procedure dbo.spRepLlamadasDepartamento    Script Date: 05/05/2000 8:53:45 PM
*****/
CREATE PROCEDURE spRepLlamadasDepartamento
    (@UNegocioID varchar(5),
    @DepartamentoID varchar(11),
    @FechaIni datetime(10),
    @FechaFin datetime(10))
AS
    /*-----*/
    /* Nombre Físico:  spRepLlamadasDepartamento */
    /* Autor:          Hctor Manuel Garza Garza */
    /* Fecha:          1 de Abril 2000 */
    /* Proyecto:       Tarificación */
    /* Descripción:     Muestra las llamadas por departamento */
    /*-----*/
    Declare @FechaIni datetime,
           @FechaFin datetime,
           @TerServMedio decimal(10,4)
SET NOCOUNT ON
SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = datediff(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #UNegocio
EXEC EMPRESA.SpAllNegocio

CREATE TABLE #UNeg_Depto (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
INSERT INTO #UNeg_Depto
EXEC EMPRESA.SpAllNeg_Depto

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescOpt varchar(50) NULL)
INSERT INTO #Departamento
EXEC EMPRESA.SpAllDepartamento

```

```

A.Extension,
A.CTUA,
A.Signal,
A.Puerto,
A.Rack,
A.Regleta,
A.File,
A.Par,
A.Raiser,
A.Craiser,
A.LFisco,
A.TipoTel,
A.NombreDisplay,
A.TipoExtensionID,
C.DescUn + ' ' + D.DescOpt as Propietario
FROM
    Extension A,
    #Neg_Depto B,
    #Negocio C,
    #Departamento D,
    Extension_Departamento E
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = 2
AND A.CommutadorID = E.CommutadorID
AND A.Extension = E.Extension
AND E.UnegocioID = B.UnegocioID
AND E.DepartamentoID = B.DepartamentoID
AND B.UnegocioID = C.UnegocioID
AND B.DepartamentoID = D.DepartamentoID
UNION
SELECT
    A.CommutadorID,
    A.Extension,
    A.CTUA,
    A.Signal,
    A.Puerto,
    A.Rack,
    A.Regleta,
    A.File,
    A.Par,
    A.Raiser,
    A.Craiser,
    A.LFisco,
    A.TipoTel,
    A.NombreDisplay,
    A.TipoExtensionID,
    B.Descripcion as Propietario
FROM
    #Extensiones
    Extension A,
    AreaGenerica B,
    Extension_AreaGenerica C
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = 3
AND A.CommutadorID = C.CommutadorID
AND A.Extension = C.Extension
AND C.AreaGenericaID = B.AreaGenericaID
UNION
SELECT
    A.CommutadorID,
    A.Extension,
    A.CTUA,
    A.Signal,
    A.Puerto,
    A.Rack,
    A.Regleta,
    A.File,
    A.Par,
    A.Raiser,
    A.Craiser,
    A.LFisco,
    A.TipoTel,
    A.NombreDisplay,
    A.TipoExtensionID,
    NULL as Propietario
FROM
    Extension A
WHERE A.CommutadorID = @CommutadorID
AND A.TipoExtensionID = NULL
ORDER BY A.CommutadorID, A.Extension
SET NOCOUNT OFF
SELECT *
FROM #Extensiones
GO
/***** Object: Stored Procedure dbo.spRepLlamadasCommutador    Script Date: 05/05/2000 8:53:45 PM
*****/
CREATE PROCEDURE spRepLlamadasCommutador
    (@CommutadorID smallint,
    @FechaIni datetime(10),
    @FechaFin datetime(10))
AS
    /*-----*/
    /* Nombre Físico:  spRepLlamadasCommutador */
    /* Autor:          Hctor Manuel Garza Garza */
    /* Fecha:          1 de Abril 2000 */
    /* Proyecto:       Tarificación */
    /* Descripción:     Muestra las llamadas por comunicador */
    /*-----*/
    Declare @FechaIni datetime,
           @FechaFin datetime,
           @TerServMedio decimal(10,4)
SET NOCOUNT ON
SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = datediff(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #UNegocio
EXEC EMPRESA.SpAllNegocio

CREATE TABLE #UNeg_Depto (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
INSERT INTO #UNeg_Depto
EXEC EMPRESA.SpAllNeg_Depto

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescOpt varchar(50) NULL)
INSERT INTO #Departamento
EXEC EMPRESA.SpAllDepartamento

```

```

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
Apellido varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedio = (SELECT Tente FROM Tipo_Llamada WHERE Priori = NULL and TipoTente = 1)

SELECT
    A.UNegocioID,
    A.DescUN,
    C.DepartamentoID,
    C.DescDpt,
    D.ComunadorID,
    D.Extension
INTO #TEExtensiones
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    #Departamento C,
    Extension_Departamento D

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (C.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID

AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
UNION
SELECT
    A.UNegocioID,
    A.DescUN,
    E.DepartamentoID,
    E.DescDpt,
    D.ComunadorID,
    D.Extension
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    #Socio C,
    Extension_Socio D,
    #Departamento E

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID

AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
AND C.SocioID = D.SocioID
UNION
SELECT
    A.UNegocioID,
    A.DescUN,
    E.DepartamentoID,
    E.DescDpt,
    D.ComunadorID,
    D.Extension
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    AreaGenerica C,
    Extension_AreaGenerica D,
    #Departamento E

WHERE (B.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT
    A.UNegocioID,
    A.DescUN,
    A.DepartamentoID,
    A.DescDpt,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
            1
        ELSE
            0
        END) AS Alerta,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTente = 1 then
            D.Tente
        when D.TipoTente = 2 then
            D.Tente * CEILING(B.Duracion / 60) +
    @TerServMedio
        else
            0
        END) as Importe
    /*CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/
INTO
FROM
    #TEExtensiones A,
    Llamada B,
    Llamada_Telme C,
    Tipo_Llamada D,
    Extension_TipoLL E

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmeID = C.Llamada_TelmeID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.DescUN, A.DepartamentoID, A.DescDpt, D.TipoLlamadaID, D.Descripcion

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

***** Object: Stored Procedure dbo.spRepLlamadasDepartamentoGraf Script Date: 05/05/2000 8:53:45
Pg: 1/1
CREATE PROCEDURE spRepLlamadasDepartamentoGraf
    (@UNegocioID varchar(5),
    @TipoLlamadaID smallint,
    @FechaInicial varchar(10),
    @FechaFinal varchar(10))
AS
    *****
    /* Nombre (Pais): spRepLlamadasDepartamentoGraf */
    /* Autor: Hctor Manuel Garza Garza */
    /* Fecha: 8 de Abril 2000 */
    /* Proyecto: Tarificación */
    /* Descripción: Muestra las llamadas por departamento para reporte grafico */
    *****
    Declare @FechaInicial datetime,
    @FechaFinal datetime,

```

```

@TerServMedio decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicial = CONVERT(DATETIME, @FechaInicial, 103),
    @FechaFinal = dateadd(day,1,CONVERT(DATETIME, @FechaFinal, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint
NULL)

INSERT INTO #UNegocio
EXEC EMPRESA.SpASUNegocio

CREATE TABLE #IDepartamento (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

INSERT INTO #IDepartamento
EXEC EMPRESA.SpASIDepartamento

CREATE TABLE #UNeg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #UNeg_Departamento
EXEC EMPRESA.SpASUNeg_Departamento

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
Apellido varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedio = (SELECT Tente FROM Tipo_Llamada WHERE Priori = NULL and TipoTente = 1)

SELECT
    A.UNegocioID,
    A.DescUN,
    C.DepartamentoID,
    C.DescDpt,
    D.ComunadorID,
    D.Extension
INTO #TEExtensiones
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    #Departamento C,
    Extension_Departamento D

WHERE A.UNegocioID = @UNegocioID
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID
AND B.DepartamentoID = D.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
UNION
SELECT
    A.UNegocioID,
    A.DescUN,
    E.DepartamentoID,
    E.DescDpt,
    D.ComunadorID,
    D.Extension
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    #Socio C,
    Extension_Socio D,
    #Departamento E

WHERE A.UNegocioID = @UNegocioID
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
AND C.SocioID = D.SocioID
UNION
SELECT
    A.UNegocioID,
    A.DescUN,
    E.DepartamentoID,
    E.DescDpt,
    D.ComunadorID,
    D.Extension
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    AreaGenerica C,
    Extension_AreaGenerica D,
    #Departamento E

WHERE A.UNegocioID = @UNegocioID
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT
    A.DepartamentoID,
    A.DescDpt,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTente = 1 then
            D.Tente
        when D.TipoTente = 2 then
            D.Tente * CEILING(B.Duracion / 60) +
    @TerServMedio
        else
            0
        END) as Importe
    /*CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/
INTO
FROM
    #TEExtensiones A,
    Llamada B,
    Llamada_Telme C,
    Tipo_Llamada D,
    Extension_TipoLL E

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmeID = C.Llamada_TelmeID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.DepartamentoID, A.DescDpt, D.TipoLlamadaID, D.Descripcion

SET NOCOUNT OFF

SELECT
    A.DepartamentoID,
    A.DescDpt
INTO #OJEPTOS
FROM
    #UNeg_Departamento B,
    #Departamento A

WHERE B.UNegocioID = @UNegocioID
AND B.DepartamentoID = A.DepartamentoID

SELECT A.DepartamentoID,
    A.DescDpt,
    B.TipoLlamadaID
INTO
    #UN_TLL
FROM
    Tipo_Llamada B,
    #OJEPTOS A

WHERE B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1

SET NOCOUNT OFF

```

```

SELECT      A.DepartamentoID,
            A.DescOpt,
            A.TipoLlamadaID,
            ISNULL(B.Cantidad,0) as Cantidad,
            ISNULL(B.Tiempo,0) as Tiempo,
            ISNULL(B.Importe,0) as Importe
FROM        #Llamadas B
WHERE A.DepartamentoID = B.DepartamentoID
AND A.TipoLlamadaID = B.TipoLlamadaID

AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
AND EXISTS (SELECT * FROM #Llamadas C WHERE A.DepartamentoID = C.DepartamentoID)
GO

***** Object: Stored Procedure dbo.spRepLlamadasDepctoDetalle    Script Date: 05/05/2000 6:53:45 PM
CREATE PROCEDURE spRepLlamadasDepctoDetalle
    (@UnegocioID varchar(5),
    @DepartamentoID varchar(10),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
    /***** Nombre Pkco: spRepLlamadasDepctoDetalle          */
    /**** Autor: Hctor Manuel Garza Gerza                    */
    /**** Fecha: 20 de Abril 2000                             */
    /**** Proyecto: Tarificación                               */
    /**** Descripción: Muestra a detalle las llamadas de un departamento */
    /*****
    DECLARE @FechaIni datetime,
            @FechaFin datetime,
            @TarServMedio decimal(10,4)

SET NOCOUNT ON

SELECT      @FechaIni = CONVERT(DATETIME,@FechaIni,103),
            @FechaFin = dateadd(day,1,CONVERT(DATETIME,@FechaFin,103))

CREATE TABLE #Unegocio (UnegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #Unegocio
EXEC EMPRESA..SpAUUnegocio

CREATE TABLE #Uneg_Departamento (UnegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
INSERT INTO #Uneg_Departamento
EXEC EMPRESA..SpAUUneg_Departamento

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)
INSERT INTO #Departamento
EXEC EMPRESA..SpAUDepartamento

CREATE TABLE #Socio (UnegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL, AMaterno varchar(20) NULL, Mail varchar(50) NULL)
INSERT INTO #Socio
EXEC EMPRESA..SpAUSocio

SELECT @TarServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE PkLlpa = NULL and TipoTarifa = 1)

SELECT      A.UnegocioID,
            A.DescUN,
            C.DepartamentoID,
            C.DescDpt,
            TipoExtensiones2,
            CONVERT(varchar(11),C.DepartamentoID) as Clave,
            CONVERT(varchar(73),C.DescDpt) as Descripción,
            D.ComutadorID,
            D.Extension,
            #Extensiones
            #Unegocio A,
            #Uneg_Departamento B,
            #Departamento C,
            Extension_Socio D

WHERE (A.UnegocioID = @UnegocioID OR @UnegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UnegocioID = B.UnegocioID
AND B.UnegocioID = D.UnegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID
UNION
SELECT      A.UnegocioID,
            A.DescUN,
            E.DepartamentoID,
            E.DescDpt,
            TipoExtensiones1,
            CONVERT(varchar(11),C.SocioID) as Clave,
            CONVERT(varchar(73),C.APaterno + " * " + C.Amaterno + " * " + C.Nombre) as Descripción,
            D.Extension,
            #Unegocio A,
            #Uneg_Departamento B,
            #Socio C,
            Extension_Socio D,
            #Departamento E

WHERE (A.UnegocioID = @UnegocioID OR @UnegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UnegocioID = B.UnegocioID
AND B.UnegocioID = C.UnegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID
UNION
SELECT      A.UnegocioID,
            A.DescUN,
            E.DepartamentoID,
            E.DescDpt,
            TipoExtensiones3,
            CONVERT(varchar(11),C.AreaGenericaID) as Clave,
            CONVERT(varchar(73),C.Descripción) as Descripción,
            D.ComutadorID,
            D.Extension,
            #Unegocio A,
            #Uneg_Departamento B,
            AreaGenerica C,
            Extension_AreaGenerica D,
            #Departamento E

WHERE (A.UnegocioID = @UnegocioID OR @UnegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")

```

```

AND A.UnegocioID = B.UnegocioID
AND B.UnegocioID = C.UnegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = E.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT      A.UnegocioID,
            A.DescUN,
            A.DepartamentoID,
            A.DescDpt,
            A.TipoExtension,
            A.Clave,
            A.Descripción,
            D.TipoLlamadaID,
            D.Descripción as DescTipoll,
            SUM(Case
                WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
                    1
                ELSE
                    0
            END) as Alerta,
            COUNT(*) as Cantidad,
            CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
            SUM(Case
                when D.TipoTarifa = 1 then
                    D.Tarifa
                when D.TipoTarifa = 2 then
                    D.Tarifa * CEILING(B.Duracion / 60) +
            @TarServMedio
            else
                0
            END) as Importe
            /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
FROM        #Llamadas
            #Extensiones A,
            Llamada B,
            Llamada_Telme C,
            Tipo_Llamada D,
            Extension_Tipoll E
WHERE A.ComutadorID = B.ComutadorID
AND A.Extension = B.Extension
AND B.ComutadorID = E.ComutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaIni
AND B.Fecha < @FechaFin

AND B.Llamada_TelmeID = C.Llamada_TelmeID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UnegocioID, A.DescUN, A.DepartamentoID, A.DescDpt, A.TipoExtension, A.Clave,
            A.Descripción, D.TipoLlamadaID, D.Descripción

SET NOCOUNT OFF

SELECT *
FROM #Llamadas

***** Object: Stored Procedure dbo.spRepLlamadasSocio    Script Date: 05/05/2000 6:53:45 PM *****
CREATE PROCEDURE spRepLlamadasSocio
    (@SocioID varchar(11),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
    /***** Nombre Pkco: spRepLlamadasSocio          */
    /**** Autor: Hctor Manuel Garza Gerza                    */
    /**** Fecha: 1 de Abril 2000                             */
    /**** Proyecto: Tarificación                               */
    /**** Descripción: Muestra las llamadas de un socio */
    /*****
    DECLARE @FechaIni datetime,
            @FechaFin datetime,
            @TarServMedio decimal(10,4)

SET NOCOUNT ON

SELECT      @FechaIni = CONVERT(DATETIME,@FechaIni,103),
            @FechaFin = dateadd(day,1,CONVERT(DATETIME,@FechaFin,103))

CREATE TABLE #Socio (UnegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL, AMaterno varchar(20) NULL, Mail varchar(50) NULL)
INSERT INTO #Socio
EXEC EMPRESA..SpAUSocio

SELECT @TarServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE PkLlpa = NULL and TipoTarifa = 1)

SELECT      A.SocioID,
            A.APaterno + " * " + A.Amaterno + " * " + A.Nombre as Nombre,
            B.ComutadorID,
            B.Extension,
            #Extensiones
            #Socio A,
            Extension_Socio B

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT      A.SocioID,
            A.Nombre,
            D.TipoLlamadaID,
            D.Descripción as DescTipoll,
            SUM(Case
                WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
                    1
                ELSE
                    0
            END) as Alerta,
            COUNT(*) as Cantidad,
            CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
            SUM(Case
                when D.TipoTarifa = 1 then
                    D.Tarifa
                when D.TipoTarifa = 2 then
                    D.Tarifa * CEILING(B.Duracion / 60) +
            @TarServMedio
            else
                0
            END) as Importe
            /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
FROM        #Llamadas
            #Extensiones A,
            Llamada B,

```

```

Llamada_Telmer C,
Tipo_Llamada D,
Extension_TipoLL E
WHERE A.ComputadorID = B.ComputadorID
AND A.Extension = B.Extension
AND B.ComputadorID = E.ComputadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin

AND B.Llamada_TelmedID = C.Llamada_TelmedID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

***** Object: Stored Procedure dbo.spRepLlamadasSocioDetalle Script Date: 05/05/2000 8:53:45 PM *****
CREATE PROCEDURE spRepLlamadasSocioDetalle
    (@SocioID varchar(11),
    @FechaInicio datetime(10),
    @FechaFin datetime(10))
AS
    /*****
    / Nombre Páramo: spRepLlamadasSocioDetalle
    / Autor: Hctor Manuel Garza Garza
    / Fecha: 8 de Abril 2000
    / Proyecto: Tarificación
    / Descripción: Muestra llamadas a detalle de un socio
    *****/

    DECLARE @FechaInicio datetime,
            @FechaFin datetime,
            @TarServMedido decimal(10,4)

    SET NOCOUNT ON

    SELECT
        @FechaInicio = CONVERT(DATETIME, @FechaInicio, 103),
        @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

    CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
        varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
        ApellidoMater varchar(20) NULL, Mail varchar(50) NULL)

    INSERT INTO #Socio
    EXEC EMPRESA.SpAISocio

    SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

    SELECT
        A.SocioID,
        A.Apellido + ' ' + A.ApellidoMater + ' ' + A.Nombre as Nombre,
        B.ComputadorID,
        B.Extension,
        #TEExtensiones,
        #Socio A,
        Extension_Socio B

    INTO
    FROM

    WHERE A.SocioID = B.SocioID
    AND (A.SocioID = @SocioID OR @SocioID = 'ALLSOC')

    SELECT
        A.SocioID,
        A.Nombre,
        D.TipoLlamadaID,
        D.Descripcion as DescTipoLL,
        B.Telefono,
        ISNULL(F.TelefonoID, -1) as Permido,
        SUM(Case
            WHEN ISNULL(F.TelefonoID, -1) = -1 THEN
                1
            ELSE
                0
            END) AS Alerte,
        Count(*) as Cantid,
        CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
        SUM(Case
            when D.TipoTarifa = 1 then
                D.Tarifa
            when D.TipoTarifa = 2 then
                D.Tarifa * CEILING(B.Duracion / 60) +
                @TarServMedido
            else
                0
            END) as Importe
        /CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as ImporteY

    INTO
    FROM
        #Llamadas
        #TEExtensiones A,
        Llamada B,
        Llamada_Telmer C,
        Tipo_Llamada D,
        Extension_TipoLL E,
        Tel_Permido F

    WHERE A.ComputadorID = B.ComputadorID
    AND A.Extension = B.Extension
    AND B.ComputadorID = E.ComputadorID
    AND B.Extension = E.Extension
    AND B.TipoLlamadaID = E.TipoLlamadaID
    AND B.Fecha >= @FechaInicio
    AND B.Fecha < @FechaFin

    AND B.Llamada_TelmedID = C.Llamada_TelmedID
    AND B.TipoLlamadaID = D.TipoLlamadaID
    AND B.Telefono = F.TelefonoID
    GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, F.TelefonoID

    SET NOCOUNT OFF

    SELECT *
    FROM #Llamadas
    GO

    /***** Object: Stored Procedure dbo.spRepLlamadasSocioGraf Script Date: 05/05/2000 8:53:45 PM *****
    CREATE PROCEDURE spRepLlamadasSocioGraf
        (@NegocioID varchar(5),
        @DepartamentoID varchar(10),
        @TipoLlamadaID smallint,
        @FechaInicio datetime(10),
        @FechaFin datetime(10))
    AS
        /*****
        / Nombre Páramo: spRepLlamadasSocioGraf
        / Autor: Hctor Manuel Garza Garza
        / Fecha: 20 de Abril 2000
        / Proyecto: Tarificación
        *****/

```

```

/ Descripción: Muestra las llamadas de un socio para el
/ reporte grafico

/*****
/ Declaro @FechaInicio datetime,
/ @FechaFin datetime,
/ @TarServMedido decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicio = CONVERT(DATETIME, @FechaInicio, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Unegocio (UNegocioID varchar(5) NULL, DescUn varchar(50) NULL, UbicacionID smallint
    NULL)

INSERT INTO #Unegocio
EXEC EMPRESA.SpAISUnegocio

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

INSERT INTO #Departamento
EXEC EMPRESA.SpAISDepartamento

CREATE TABLE #Neg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Neg_Departamento
EXEC EMPRESA.SpAISNeg_Departamento

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
    varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
    ApellidoMater varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpAISocio

SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

SELECT
    C.SocioID,
    C.Apellido + ' ' + C.ApellidoMater + ' ' + C.Nombre as Nombre,
    D.ComputadorID,
    D.Extension,
    #TEExtensiones,
    #Unegocio A,
    #Neg_Departamento B,

    #Socio C,
    Extension_Socio D,
    #Departamento E

WHERE A.UNegocioID = @UNegocioID
AND E.DepartamentoID = @DepartamentoID
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID

AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = E.DepartamentoID
AND C.SocioID = D.SocioID

SELECT
    A.SocioID,
    A.Nombre,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    Count(*) as Cantid,
    CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
            @TarServMedido
        else
            0
        END) as Importe
        /CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as ImporteY

    INTO
    FROM
        #Llamadas
        #TEExtensiones A,
        Llamada B,
        Llamada_Telmer C,
        Tipo_Llamada D

    WHERE A.ComputadorID = B.ComputadorID
    AND A.Extension = B.Extension
    AND B.Fecha >= @FechaInicio
    AND B.Fecha < @FechaFin
    AND B.Llamada_TelmedID = C.Llamada_TelmedID
    AND B.TipoLlamadaID = D.TipoLlamadaID
    AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
    GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion

    SELECT
        A.SocioID,
        A.Nombre,
        A.TipoLlamadaID,
        ISNULL(B.Cantid, 0) as Cantid,
        ISNULL(B.Tiempo, 0) as Tiempo,
        ISNULL(B.Importe, 0) as Importe

    FROM
        #UN_TLL A,
        #Llamadas B

    WHERE A.SocioID = B.SocioID
    AND A.TipoLlamadaID = B.TipoLlamadaID

    AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
    AND EXISTS (SELECT * FROM #Llamadas C WHERE A.SocioID = C.SocioID)
    GO

    /***** Object: Stored Procedure dbo.spRepLlamadasSocioTelefono Script Date: 05/05/2000 8:53:45 PM *****
    CREATE PROCEDURE spRepLlamadasSocioTelefono
        (@SocioID varchar(11),
        @Telefono varchar(20),
        @FechaInicio datetime(10),
        @FechaFin datetime(10))
    AS
        /*****

```



```

when D.TipTarifa = 2 then
    D.Tarifa * CEILING(B.Duracion / 80) +
else
    0
END) as Importe
/CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/
INTO
FROM

Llamada B,
Llamada_Telmas C,
Tipo_Llamada D,
Extension_TipLL E,
Tel_Permiso F
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin

AND B.Llamada_TelmasID = C.Llamada_TelmasID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = F.Telefono
GROUP BY A.UnnegocioID, A.DescUN, A.DepartamentoID, A.DescOpt, A.TipoExtension, A.Clave,
A.Descripcion, D.TipoLlamadaID, D.Descripcion, B.Telefono, F.TelefonoID

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepLlamadasTelefonoDetalle Script Date: 05/05/2000 6:53:45 PM *****/
CREATE PROCEDURE spRepLlamadasTelefonoDetalle
    (@UnnegocioID varchar(5),
    @DepartamentoID varchar(10),
    @Clave varchar(11),
    @Telefono varchar(20),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*****
Nombre: spRepLlamadasTelefonoDetalle
Autor: Hctor Manuel Gerza Gerza
Fecha: 8 de Abril 2000
Proyecto: Tarificación
Descripción: Muestra las llamadas con el detalle de telefonos marcados
*****/

Declare @FechaInicio datetime,
        @FechaFin datetime,
        @TerServMedio decimal(10,4),
        @ClaveAreaGen int

SET NOCOUNT ON

SELECT
    @FechaInicio = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Unnegocio (UnnegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #Unnegocio
EXEC EMPRESA.SpABUnnegocio

CREATE TABLE #Unneg_Departamento (UnnegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
INSERT INTO #Unneg_Departamento
EXEC EMPRESA.SpABUnneg_Departamento

CREATE TABLE #Departamento (DepartamentoID varchar(10), DescOpt varchar(50))
INSERT INTO #Departamento
EXEC EMPRESA.SpABDepartamento

CREATE TABLE #Socio (UnnegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, ApellidoNac varchar(20) NULL, Mail varchar(50) NULL)
INSERT INTO #Socio
EXEC EMPRESA.SpABSocio

IF ISNUMERIC(@Clave) = 0
    SELECT @ClaveAreaGen = 0
ELSE
    SELECT @ClaveAreaGen = CONVERT(SMALLINT, @Clave)

SELECT @TerServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.UnnegocioID,
    A.DescUN,
    C.DepartamentoID,
    C.DescOpt,
    TipoExtension=2,
    CONVERT(varchar(11),C.DepartamentoID) as Clave,
    CONVERT(varchar(13),C.DescOpt) as Descripción,
    D.CommutadorID,
    D.Extension,
    #TEExtensiones
    #Unnegocio A,
    #Unneg_Departamento B,
    #Departamento C,
    Extension_Departamento D

WHERE (A.UnnegocioID = @UnnegocioID OR @UnnegocioID = 'ALLUN')
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = 'ALLDPT')
AND A.UnnegocioID = D.UnnegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin

AND B.Llamada_TelmasID = C.Llamada_TelmasID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = @Telefono
AND B.Telefono = F.Telefono
GROUP BY A.UnnegocioID, A.DescUN, A.DepartamentoID, A.DescOpt, A.TipoExtension, A.Clave,
A.Descripcion, D.TipoLlamadaID, D.Descripcion, B.Telefono, B.Fecha, F.TelefonoID

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepLlamadasUnnegocio Script Date: 05/05/2000 6:53:46 PM *****/
CREATE PROCEDURE spRepLlamadasUnnegocio
    (@UnnegocioID varchar(5),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*****
Nombre: spRepLlamadasUnnegocio
Autor: Hctor Manuel Gerza Gerza
Fecha: 1 de Abril 2000
Proyecto: Tarificación
Descripción: Muestra las llamadas por unidad de neg.
*****/

Declare @FechaInicio datetime,
        @FechaFin datetime,
        @TerServMedio decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicio = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Unnegocio (UnnegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #Unnegocio

```

```

CONVERT(varchar(11),C.Apellido + ' ' + C.Nombre) as Descripción,
D.CommutadorID,
D.Extension,
#Unnegocio A,
#Unneg_Departamento B,
#Socio C,
Extension_Socio D,
#Departamento E

WHERE (A.UnnegocioID = @UnnegocioID OR @UnnegocioID = 'ALLUN')
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = 'ALLDPT')
AND A.UnnegocioID = B.UnnegocioID

AND B.UnnegocioID = C.UnnegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.CommutadorID = C.CommutadorID
AND C.SocioID = D.SocioID
AND @TipoExtensionID = 1
AND (C.SocioID = @Clave OR @Clave = 'ALL')

UNION
SELECT
    A.UnnegocioID,
    A.DescUN,
    E.DepartamentoID,
    E.DescOpt,
    TipoExtension=3,
    CONVERT(varchar(11),C.AreaGenericaID) as Clave,
    CONVERT(varchar(13),C.Descripcion) as Descripción,
    D.CommutadorID,
    D.Extension

FROM
    #Unnegocio A,
    #Unneg_Departamento B,
    AreaGenerica C,
    Extension_AreaGenerica D,
    #Departamento E

WHERE (A.UnnegocioID = @UnnegocioID OR @UnnegocioID = 'ALLUN')
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = 'ALLDPT')
AND A.UnnegocioID = B.UnnegocioID
AND B.UnnegocioID = C.UnnegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.CommutadorID = D.CommutadorID
AND C.AreaGenericaID = D.AreaGenericaID
AND @TipoExtensionID = 3
AND (C.AreaGenericaID = @ClaveAreaGen OR @Clave = 'ALL')

SELECT
    A.UnnegocioID,
    A.DescUN,
    A.DepartamentoID,
    A.DescOpt,
    A.TipoExtension,
    A.Clave,
    A.Descripcion,
    D.TipoLlamadaID,
    D.Descripcion as DescTipLL,
    B.Telefono,
    CONVERT(varchar(10),B.Fecha,103) + ' ' + CONVERT(varchar(8),B.Fecha,106) as
Fecha,
    ISNULL(F.TelefonoID,1) as Permiso,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadaID,1) = -1 THEN
            1
        ELSE
            0
    END) AS Alerta,
    CONVERT(decimal(20,2),SUM(B.Duracion / 80)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 80) +
    else
        0
    END) as Importe
/CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/
INTO
FROM

Llamada B,
Llamada_Telmas C,
Tipo_Llamada D,
Extension_TipLL E,
Tel_Permiso F
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin

AND B.Llamada_TelmasID = C.Llamada_TelmasID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = @Telefono
AND B.Telefono = F.Telefono
GROUP BY A.UnnegocioID, A.DescUN, A.DepartamentoID, A.DescOpt, A.TipoExtension, A.Clave,
A.Descripcion, D.TipoLlamadaID, D.Descripcion, B.Telefono, B.Fecha, F.TelefonoID

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepLlamadasUnnegocio Script Date: 05/05/2000 6:53:46 PM *****/
CREATE PROCEDURE spRepLlamadasUnnegocio
    (@UnnegocioID varchar(5),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*****
Nombre: spRepLlamadasUnnegocio
Autor: Hctor Manuel Gerza Gerza
Fecha: 1 de Abril 2000
Proyecto: Tarificación
Descripción: Muestra las llamadas por unidad de neg.
*****/

Declare @FechaInicio datetime,
        @FechaFin datetime,
        @TerServMedio decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicio = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Unnegocio (UnnegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)
INSERT INTO #Unnegocio

```

EXEC EMPRESA, SpAllNegocio

CREATE TABLE #Neg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Neg_Departamento
EXEC EMPRESA, SpAllNeg_DepartamentoCREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
ApellidoMaterno varchar(20) NULL, Mail varchar(50) NULL)INSERT INTO #Socio
EXEC EMPRESA, SpAllSocio

SELECT @TerServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

SELECT
A.UNegocioID,
A.DescUN,
D.ComputadorID,
D.Extension

INTO #Extensiones

FROM

#Negocio A,
#Neg_Departamento B,
Extension_Departamento DWHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID
AND B.DepartamentoID = D.DepartamentoID

UNION

SELECT

A.UNegocioID,
A.DescUN,
D.ComputadorID,
D.Extension

FROM

#Negocio A,
#Neg_Departamento B,
#Socio C,
Extension_Socio DWHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID

UNION

SELECT

A.UNegocioID,
A.DescUN,
D.ComputadorID,
D.Extension

FROM

#Negocio A,
#Neg_Departamento B,
AreaGenerica C,
Extension_AreaGenerica DWHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT

A.UNegocioID,
A.DescUN,
D.TipoLlamadaID,
D.Descripcion as DescTipoLL,

SUM(Case

WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN

1

ELSE

0

END) AS Alerta,

Count(*) as Cantidad,

CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,

SUM(Case

when D.TipoTarifa = 1 then

D.Tarifa

when D.TipoTarifa = 2 then

D.Tarifa * CEILING(B.Duracion / 60) +

else

0

END) as Importe

/*CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/

INTO

FROM

#Negocio A,
#Extensiones B,
Llamada_Telmax C,
Tipo_Llamada D,
Extension_TipoLL E

WHERE A.ComputadorID = B.ComputadorID

AND A.Extension = B.Extension

AND B.ComputadorID = E.ComputadorID

AND B.Extension = E.Extension

AND B.TipoLlamadaID = E.TipoLlamadaID

AND B.Fecha >= @FechaInicial

AND B.Fecha < @FechaFinal

AND B.Llamada_TelmaxID = C.Llamada_TelmaxID

AND B.TipoLlamadaID = D.TipoLlamadaID

GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion

/*

SELECT

A.UNegocioID,
A.DescUN,
D.TipoLlamadaID,
D.Descripcion as DescTipoLL,

SUM(Case

WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN

1

ELSE

0

END) AS Alerta

/*Llamadas2

#Extensiones A,
Llamada B,
Llamada_Telmax C,
Tipo_Llamada D,
Extension_TipoLL E

WHERE A.ComputadorID = B.ComputadorID

AND A.Extension = B.Extension

AND B.ComputadorID = E.ComputadorID

AND B.Extension = E.Extension

AND B.TipoLlamadaID = E.TipoLlamadaID

AND B.Fecha >= @FechaInicial

AND B.Fecha < @FechaFinal

AND B.Llamada_TelmaxID = C.Llamada_TelmaxID

AND B.TipoLlamadaID = D.TipoLlamadaID

GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion

*/

SET NOCOUNT OFF

SELECT *

FROM #Llamadas

GO

/*===== Object: Stored Procedure dbo.spRepLlamadasNegocioDetalle Script Date: 05/05/2000 4:53:46 PM

=====

CREATE PROCEDURE spRepLlamadasNegocioDetalle

(@UNegocioID varchar(5),

@FechaIni varchar(10),

@FechaFin varchar(10))

AS

/* Nombre Filtro: spRepLlamadasNegocioDetalle

/* Autor: Ingrid Manuel Garza Garza

/* Fecha: 8 de Abril 2000

/* Proyecto: Tarificacion

/* Descripcion: Muestra las llamadas e detalle por

/* Negocio

/*

Declare @FechaInicial datetime,

@FechaFinal datetime,

@TerServMedio decimal(10,4)

SET NOCOUNT ON

SELECT

@FechaInicial = CONVERT(DATETIME, @FechaIni, 103),

@FechaFinal = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID int(4)

NULL)

INSERT INTO #UNegocio

EXEC EMPRESA, SpAllUNegocio

CREATE TABLE #Neg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Neg_Departamento

EXEC EMPRESA, SpAllNeg_Departamento

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

INSERT INTO #Departamento

EXEC EMPRESA, SpAllDepartamento

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID

varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
ApellidoMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio

EXEC EMPRESA, SpAllSocio

SELECT @TerServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

SELECT

A.UNegocioID,
A.DescUN,
C.DepartamentoID,
C.DescDpt,
D.ComputadorID,
D.Extension

INTO #Extensiones

FROM

#Negocio A,
#Neg_Departamento B,
#Departamento C,
Extension_Departamento D

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")

AND A.UNegocioID = B.UNegocioID

AND B.UNegocioID = C.UNegocioID

AND B.DepartamentoID = C.DepartamentoID

AND B.DepartamentoID = D.DepartamentoID

UNION

SELECT

A.UNegocioID,
A.DescUN,
E.DepartamentoID,
E.DescDpt,
D.ComputadorID,
D.Extension

FROM

#Negocio A,
#Neg_Departamento B,
#Socio C,
Extension_Socio D,
#Departamento E

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")

AND A.UNegocioID = B.UNegocioID

AND B.UNegocioID = C.UNegocioID

AND B.DepartamentoID = E.DepartamentoID

AND B.DepartamentoID = C.DepartamentoID

AND C.SocioID = D.SocioID

UNION

SELECT

A.UNegocioID,
A.DescUN,
E.DepartamentoID,
E.DescDpt,
D.ComputadorID,
D.Extension

FROM

#Negocio A,
#Neg_Departamento B,
AreaGenerica C,
Extension_AreaGenerica D,
#Departamento E

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")

AND A.UNegocioID = B.UNegocioID

AND B.UNegocioID = C.UNegocioID

AND B.DepartamentoID = E.DepartamentoID

AND B.DepartamentoID = C.DepartamentoID

AND C.AreaGenericaID = D.AreaGenericaID

SELECT

A.UNegocioID,
A.DescUN,
A.DepartamentoID,
A.DescDpt,
D.TipoLlamadaID,
D.Descripcion as DescTipoLL,

SUM(Case

WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN

1

ELSE

0

END) AS Alerta,

Count(*) as Cantidad,

CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,

SUM(Case

when D.TipoTarifa = 1 then

D.Tarifa

when D.TipoTarifa = 2 then

D.Tarifa * CEILING(B.Duracion / 60) +

else

0

END) as Importe

/*CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/

SET NOCOUNT OFF

SELECT *

```

D.Tarifa * CEILING(B.Duracion / 60) *

@TarServMedido
    else
        0
    END) as Importe
    /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
#Llamadas
#TEExtensiones A,
Llamada B,
Llamada_Telmax C,
Tipo_Llamada D,
Extension_TipoLL E
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension

AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UnnegocioID, A.DescUN, A.DepartamentoID, A.DescDpt, D.TipoLlamadaID, D.Descripcion

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepLlamadasUnnegocioGraf Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepLlamadasUnnegocioGraf
    (@TipoLlamadaID smalint,
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*-----*/
/* Nombre Fkico: spRepLlamadasUnnegocioGraf */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 20 de Abril 2000 */
/* Proyecto: Tarificacion */
/* Descripción: Muestra las llamadas por unidad de negocio para el reporte grafico */
/*-----*/

Declare @FechaInicial datetime,
@FechaFinal datetime,
@TarServMedido decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicial = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFinal = datediff(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Unnegocio (UnnegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smalint NULL)

INSERT INTO #Unnegocio
EXEC EMPRESA.SpAnUnnegocio

CREATE TABLE #Unneg_Departamento (UnnegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #Unneg_Departamento
EXEC EMPRESA.SpAnUnneg_Departamento

CREATE TABLE #Socio (UnnegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, Aleterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpAnSocio

SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.UnnegocioID,
    A.DescUN,
    D.CommutadorID,
    D.Extension
FROM
    #Unnegocio A,
    #Unneg_Departamento B,
    Extension_Departamento D

WHERE A.UnnegocioID = B.UnnegocioID
AND B.UnnegocioID = D.UnnegocioID
AND B.DepartamentoID = D.DepartamentoID

UNION
SELECT
    A.UnnegocioID,
    A.DescUN,
    D.CommutadorID,
    D.Extension
FROM
    #Unnegocio A,
    #Unneg_Departamento B,
    AreaGenerica C,
    Extension_AreaGenerica D

WHERE A.UnnegocioID = B.UnnegocioID
AND B.UnnegocioID = C.UnnegocioID
AND B.DepartamentoID = C.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT
    A.UnnegocioID,
    A.DescUN,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) *
    @TarServMedido
    else
        0
    END) as Importe
    /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
#Llamadas

```

```

FROM
    #TEExtensiones A,
    Llamada B,
    Llamada_Telmax C,
    Tipo_Llamada D
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
GROUP BY A.UnnegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion

SELECT
    A.UnnegocioID,
    A.DescUN,
    A.TipoLlamadaID,
    ISNULL(B.Cantidad,0) as Cantidad,
    ISNULL(B.Tiempo,0) as Tiempo,
    ISNULL(B.Importe,0) as Importe
FROM #Un_TLL A, #Llamadas B
WHERE A.UnnegocioID = B.UnnegocioID
AND A.TipoLlamadaID = B.TipoLlamadaID
AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
AND EXISTS (SELECT * FROM #Llamadas C WHERE A.UnnegocioID = C.UnnegocioID)
GO

/***** Object: Stored Procedure dbo.spRepLlamadasUbicacion Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepLlamadasUbicacion
    (@UbicacionID smalint,
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*-----*/
/* Nombre Fkico: spRepLlamadasUbicacion */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: Tarificacion */
/* Descripción: Muestra las llamadas por ubicacion */
/*-----*/

Declare @FechaInicial datetime,
@FechaFinal datetime,
@TarServMedido decimal(10,4)

SET NOCOUNT ON

SELECT
    @FechaInicial = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFinal = datediff(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Ubicacion (UbicacionID smalint NULL, Descripcion varchar(50) NULL, CommutadorID smalint NULL)

INSERT INTO #Ubicacion
EXEC EMPRESA.SpAnUbicacion

SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.UbicacionID,
    A.Descripcion,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
            1
        ELSE
            0
    END) AS Alerta,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) *
    @TarServMedido
    else
        0
    END) as Importe
    /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
#Llamadas
#Ubicacion A,
Llamada B,
Llamada_Telmax C,
Tipo_Llamada D,
Extension_TipoLL E
WHERE A.CommutadorID = B.CommutadorID
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal

AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND (A.UbicacionID = @UbicacionID OR @UbicacionID = -1)
GROUP BY A.UbicacionID, A.Descripcion, D.TipoLlamadaID, D.Descripcion

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepLlamadasUbicacionDetalle Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepLlamadasUbicacionDetalle
    (@UbicacionID smalint,
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*-----*/
/* Nombre Fkico: spRepLlamadasUbicacionDetalle */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: Tarificacion */

```

```

-- Descripción: Muestra las llamadas a detalle por ubi
--

Declare @FechaIniciol datetime,
        @FechaFinal datetime,
        @TerServMedido decimal(10,4)

SET NOCOUNT ON

SELECT @FechaIniciol = CONVERT(DATETIME,@FechaIni,103),
        @FechaFinal = dateadd(day,1,CONVERT(DATETIME,@FechaFin,103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)

INSERT INTO #UNegocio
EXEC EMPRESA.SpAUNegocio

CREATE TABLE #UNeg_Depto (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #UNeg_Depto
EXEC EMPRESA.SpAUNeg_Depto

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, Anulado varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

SELECT A.UNegocioID,
        A.DescUN,
        D.ComunadorID,
        D.Extension
INTO #Extensiones
FROM #UNegocio A,
        #UNeg_Depto B,
        Extension_Departamento D

WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID
AND A.UbicacionID = (UbicacionID)
AND B.DepartamentoID = D.DepartamentoID
UNION
SELECT A.UNegocioID,
        A.DescUN,
        D.ComunadorID,
        D.Extension
FROM #UNegocio A,
        #UNeg_Depto B,
        #Socio C,
        Extension_Socio D
WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND A.UbicacionID = (UbicacionID)
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID
UNION
SELECT A.UNegocioID,
        A.DescUN,
        D.ComunadorID,
        D.Extension
FROM #UNegocio A,
        #UNeg_Depto B,
        AreaGenerica C,
        Extension_AreaGenerica D
WHERE A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND A.UbicacionID = (UbicacionID)
AND B.DepartamentoID = C.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT A.UNegocioID,
        A.DescUN,
        D.TipoLlamadaID,
        D.Descripcion as DescTipoLL,
        SUM(Case
                WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
                        1
                ELSE
                        0
                END) AS Alerta,
        Count(*) as Cantidad,
        CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
        SUM(Case
                when D.TipoTarifa = 1 then
                        D.Tarifa
                when D.TipoTarifa = 2 then
                        D.Tarifa * CEILING(B.Duracion / 60) +
                else
                        0
                END) as Importe
        /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/
INTO #Llamadas
FROM #Extensiones A,
        Llamada B,
        Llamada_Telmax C,
        Tipo_Llamada D,
        Extension_TipoLL E
WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaIniciol
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion

--
SELECT A.UNegocioID,
        A.DescUN,
        D.TipoLlamadaID,
        D.Descripcion as DescTipoLL,
        SUM(Case
                WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
                        1
                ELSE
                        0
                END) AS Alerta
INTO #Llamadas2
FROM #Extensiones A,
        Llamada B,
        Llamada_Telmax C,
        Tipo_Llamada D,
        Extension_TipoLL E
WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaIniciol
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion

```

```

--
Extension_TipoLL E
WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaIniciol
AND B.Fecha < @FechaFinal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripcion
--

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

--**** Object: Stored Procedure dbo.spRepLlamadasUbicacionGruf
--**** Script Date: 05/05/2000 8:53:46 PM
CREATE PROCEDURE spRepLlamadasUbicacionGruf
    (@TipoLlamadaID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
    /*****
    Nombre (Funcio): spRepLlamadasUbicacionGruf
    Autor: Hctor Manuel Garza Garza
    Fecha: 20 de Abril 2000
    Proyecto: Tarificacion
    Descripción: Muestra las llamadas por ubicacion para el
    reporte grafico
    *****/

    Declare @FechaIniciol datetime,
            @FechaFinal datetime,
            @TerServMedido decimal(10,4)

    SET NOCOUNT ON

    SELECT @FechaIniciol = CONVERT(DATETIME,@FechaIni,103),
            @FechaFinal = dateadd(day,1,CONVERT(DATETIME,@FechaFin,103))

    CREATE TABLE #Ubicacion (UbicacionID smallint NULL, Descripcion varchar(50) NULL, ComunadorID smallint NULL)

    INSERT INTO #Ubicacion
    EXEC EMPRESA.SpAUbicacion

    SELECT @TerServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Preijo = NULL and TipoTarifa = 1)

    SELECT A.UbicacionID,
            A.Descripcion,
            D.TipoLlamadaID,
            D.Descripcion as DescTipoLL,
            Count(*) as Cantidad,
            CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
            SUM(Case
                    when D.TipoTarifa = 1 then
                            D.Tarifa
                    when D.TipoTarifa = 2 then
                            D.Tarifa * CEILING(B.Duracion / 60) +
                    else
                            0
                    END) as Importe
            /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe*/
    INTO #Llamadas
    FROM #Ubicacion A,
            Llamada B,
            Llamada_Telmax C,
            Tipo_Llamada D
    WHERE A.ComunadorID = B.ComunadorID
    AND B.Fecha >= @FechaIniciol
    AND B.Fecha < @FechaFinal
    AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
    AND B.TipoLlamadaID = D.TipoLlamadaID
    GROUP BY A.UbicacionID, A.Descripcion, D.TipoLlamadaID, D.Descripcion

    SELECT A.UbicacionID, A.Descripcion, B.TipoLlamadaID
    INTO #Llamadas2
    FROM #Ubicacion A,
            Llamada B,
            Llamada_Telmax C,
            Tipo_Llamada D
    WHERE B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1

    SET NOCOUNT OFF

    SELECT A.UbicacionID,
            A.Descripcion,
            D.TipoLlamadaID,
            D.Descripcion as DescTipoLL,
            ISNULL(B.Cantidad,0) as Cantidad,
            ISNULL(B.Tiempo,0) as Tiempo,
            ISNULL(B.Importe,0) as Importe
    FROM #Llamadas A,
            #Llamadas2 B
    WHERE A.UbicacionID = B.UbicacionID
    AND A.TipoLlamadaID = B.TipoLlamadaID
    AND (B.TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1)
    AND EXISTS (SELECT * FROM #Llamadas C WHERE A.UbicacionID = C.UbicacionID)
    ORDER BY A.TipoLlamadaID, A.UbicacionID
    GO

    /*****
    Object: Stored Procedure dbo.spRepOpcionTipoAgruacion
    Script Date: 05/05/2000 8:53:46 PM
    *****/
    CREATE PROCEDURE spRepOpcionTipoAgruacion
        (@TipoOpcion smallint,
        @Agruacion smallint,
        @PartID smallint)
    AS
        /*****
        Nombre (Funcio): spRepOpcionTipoAgruacion
        Autor: Hctor Manuel Garza Garza
        Fecha: 1 de Abril 2000
        Proyecto: Tarificacion
        Descripción: Muestra las opciones por el tipo de
        agruacion
        *****/

        SELECT A.*
        FROM Opc_sistema A,Parti_Opcion B
        WHERE A.TipoOpcion = @TipoOpcion
        AND A.Agruacion = @Agruacion

```

```

AND A.OpcionID = B.OpcionID
AND B.PermisoID = @PermisoID
GO

/***** Object: Stored Procedure dbo.spRepSolicitud Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepSolicitud
    (@TipoSolicitudID int)
AS
    /*****
    Nombre: spRepSolicitud
    Autor: Hctor Manuel Garza Garza
    Fecha: 20 de Abril 2000
    Proyecto: Tercerizacion
    Descripción: Muestra la información de una solicitud
    *****/

SET NOCOUNT ON

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FecInsc datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SET NOCOUNT OFF

SELECT A.Solicitud_ID,
       B.APaterno + ' ' + B.AMaterno + ' ' + B.Nombre as Nombre,
       CONVERT(varchar(10),A.Fecha,103) as Fecha
FROM   #Socio A
WHERE  A.SocioID = B.SocioID
AND A.Tipo_SolicitudID = @TipoSolicitudID
AND A.Estado = @Estado
ORDER BY A.Solicitud_ID
GO

/***** Object: Stored Procedure dbo.spRepTelSocio Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTelSocio
    (@SocioID varchar(11),
    @FechIni datetime,
    @FechFin datetime)
AS
    /*****
    Nombre: spRepTelSocio
    Autor: Hctor Manuel Garza Garza
    Fecha: 1 de Abril 2000
    Proyecto: Tercerizacion
    Descripción: Muestra las llamadas a los tel. no involucrados por socio
    *****/

DECLARE @FechIni datetime,
        @FechFin datetime,
        @TerServMedido decimal(10,4)

SET NOCOUNT ON

SELECT @FechIni = CONVERT(DATETIME,@FechIni,103),
       @FechFin = dateadd(day,1,CONVERT(DATETIME,@FechFin,103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FecInsc datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedido = (SELECT TerSoc FROM Tipo_Llamada WHERE Prejio = NULL and TipoTerSoc = 1)

SELECT A.SocioID,
       A.Nombre,
       B.ComunadorID,
       B.Extension,
       #TerSoc,
       #Socio A,
       Extension_Socio B

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = 'ALLSOC')

SELECT A.SocioID,
       A.Nombre,
       D.TipoLlamadaID,
       D.Descripcion as DescTipoLL,
       B.Telefono,
       ISNULL(F.TelefonoID,-1) as PermisoID,
       SUM(Case
           WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
               1
           ELSE
               0
       END) as Alerta,
       CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
       SUM(Case
           when D.TipoTerSoc = 1 then
               D.TerSoc
           when D.TipoTerSoc = 2 then
               D.TerSoc * CEILING(B.Duracion / 60) +
       @TerServMedido
       else
               0
       END) as Importe
       /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/'

INTO #Llamadas
FROM #TerSoc A,
     Llamada B,
     Llamada_TelSoc C,
     Tipo_Llamada D,
     Extension_TipoLL E,
     Tel_Permiso F

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechIni
AND B.Fecha < @FechFin

AND B.Llamada_TelSocID = C.Llamada_TelSocID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = @Telefono
AND B.Telefono = F.Telefono
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, B.Fecha, F.TelefonoID

SET NOCOUNT OFF

SELECT A.*
FROM #Llamadas A
WHERE NOT EXISTS (SELECT * FROM Tel_Permiso B WHERE A.Telefono = B.Telefono)
GO

/***** Object: Stored Procedure dbo.spRepTelSocioGraf Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTelSocioGraf
    (@SocioID varchar(11),
    @TipoLlamadaID int,
    @FechIni datetime,
    @FechFin datetime)
AS
    /*****
    Nombre: spRepTelSocioGraf
    Autor: Hctor Manuel Garza Garza
    Fecha: 20 de Abril 2000
    Proyecto: Tercerizacion
    Descripción: Muestra las llamadas a los tel. no involucrados por socio para reporte graf/
    *****/

DECLARE @FechIni datetime,
        @FechFin datetime,
        @TerServMedido decimal(10,4)

SET NOCOUNT ON

```

```

SELECT A.*
FROM #Llamadas A
WHERE NOT EXISTS (SELECT * FROM Tel_Permiso B WHERE A.Telefono = B.Telefono)
GO

/***** Object: Stored Procedure dbo.spRepTelSocioDetalle Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTelSocioDetalle
    (@SocioID varchar(11),
    @Telefono varchar(20),
    @FechIni datetime,
    @FechFin datetime)
AS
    /*****
    Nombre: spRepTelSocioDetalle
    Autor: Hctor Manuel Garza Garza
    Fecha: 8 de Abril 2000
    Proyecto: Tercerizacion
    Descripción: Muestra las llamadas a detalle a los tel. no involucrados por socio
    *****/

DECLARE @FechIni datetime,
        @FechFin datetime,
        @TerServMedido decimal(10,4)

SET NOCOUNT ON

SELECT @FechIni = CONVERT(DATETIME,@FechIni,103),
       @FechFin = dateadd(day,1,CONVERT(DATETIME,@FechFin,103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FecInsc datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedido = (SELECT TerSoc FROM Tipo_Llamada WHERE Prejio = NULL and TipoTerSoc = 1)

SELECT A.SocioID,
       A.Nombre,
       D.TipoLlamadaID,
       D.Descripcion as DescTipoLL,
       B.Telefono,
       ISNULL(F.TelefonoID,-1) as PermisoID,
       SUM(Case
           WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
               1
           ELSE
               0
       END) as Alerta,
       CONVERT(varchar(10),B.Fecha,103) + ' ' + CONVERT(varchar(8),B.Fecha,108) as
Fecha,
       CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
       SUM(Case
           when D.TipoTerSoc = 1 then
               D.TerSoc
           when D.TipoTerSoc = 2 then
               D.TerSoc * CEILING(B.Duracion / 60) +
       @TerServMedido
       else
               0
       END) as Importe
       /CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/'

INTO #Llamadas
FROM #TerSoc A,
     Llamada B,
     Llamada_TelSoc C,
     Tipo_Llamada D,
     Extension_TipoLL E,
     Tel_Permiso F

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechIni
AND B.Fecha < @FechFin

AND B.Llamada_TelSocID = C.Llamada_TelSocID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = @Telefono
AND B.Telefono = F.Telefono
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, B.Fecha, F.TelefonoID

```

```

SELECT      @FechaInicial = CONVERT(DATETIME, @FechaIni, 103),
            @FechaFinal = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TerServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT      A.SocioID,
            A.APaterno + ' ' + A.AMaterno + ' ' + A.Nombre as Nombre,
            B.CommutadorID,
            B.Extension
            #Extensiones
FROM        #Socio A,
            Extension_Socio B

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT      A.SocioID,
            A.Nombre,
            D.TipoLlamadaID,
            D.Descripcion as DescTipoLL,
            B.Telefono,
            Count(*) as Cantidad,
            CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
            SUM(Case
                when D.TipoTarifa = 1 then
                    D.Tarifa
                when D.TipoTarifa = 2 then
                    D.Tarifa * CEILING(B.Duracion / 60) +
            @TerServMedio
            else
                0
            END) as Importe
            /*CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as Importe*/
FROM        #Llamadas
            #Extensiones A,
            Llamada B,
            Llamada_Telmo C,
            Tipo_Llamada D
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal

AND B.Llamada_TelmoID = C.Llamada_TelmoID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.TipoLlamadaID = @TipoLlamadaID
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono

SET NOCOUNT OFF

SELECT A *
FROM #Llamadas A
WHERE NOT EXISTS (SELECT * FROM Tel_Permiso B WHERE A.Telefono = B.Telefono)
GO

/***** Object: Stored Procedure dbo.spRepTipoLlamada Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTipoLlamada
(@TipoLlamadaID smallint)
AS
/*-----*/
/* Nombre Pkco: spRepTipoLlamada */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 1 de Abril 2000 */
/* Proyecto: Tarificación */
/* Descripción: Selecciona un tipo de llamada */
SELECT TipoLlamadaID, Descripción, Prefijo, TipoExtensLlame
FROM Tipo_Llamada
WHERE TipoLlamadaID = @TipoLlamadaID OR @TipoLlamadaID = -1
ORDER BY TipoLlamadaID
GO

/***** Object: Stored Procedure dbo.spRepTipoLlamadaExtension Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTipoLlamadaExtension
(@CommutadorID smallint,
@Extension smallint)
AS
/*-----*/
/* Nombre Pkco: spRepTipoLlamadaExtension */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 8 de Abril 2000 */
/* Proyecto: Tarificación */
/* Descripción: Muestra los tipos de llamada asignados a/ una extension */
/* ----- */
SELECT      A.TipoLlamadaID,
            A.Descripción
FROM        Tipo_Llamada A,
            Extension_TipoLL B
WHERE B.CommutadorID = @CommutadorID
AND B.Extension = @Extension
AND B.TipoLlamadaID = A.TipoLlamadaID
ORDER BY A.TipoLlamadaID
GO

/***** Object: Stored Procedure dbo.spRepTopDepctoDetalle Script Date: 05/05/2000 8:53:46 PM *****/
CREATE PROCEDURE spRepTopDepctoDetalle
(@UNegocioID varchar(5),
@DepartamentoID varchar(10),
@TipoLlamadaID smallint,
@FechaIni varchar(10),
@FechaFin varchar(10),
@Ordenamiento smallint)
AS
/*-----*/
/* Nombre Pkco: spRepTopDepctoDetalle */
/* Autor: Hctor Manuel Garza Garza */
/* Fecha: 20 de Abril 2000 */
/* Proyecto: Tarificación */
/* Descripción: Muestra el top de llamadas por depcto. */
/* ----- */
DECLARE @FechaInicial datetime,
        @FechaFinal datetime,
        @TerServMedio decimal(10,4)

```

SET NOCOUNT ON

```

SELECT      @FechaInicial = CONVERT(DATETIME, @FechaIni, 103),
            @FechaFinal = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

```

```

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint
NULL)

```

```

INSERT INTO #UNegocio
EXEC EMPRESA.SpASUNegocio

```

```

CREATE TABLE #UNeg_Depto (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

```

```

INSERT INTO #UNeg_Depto
EXEC EMPRESA.SpASUNeg_Depto

```

```

CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescDpt varchar(50) NULL)

```

```

INSERT INTO #Departamento
EXEC EMPRESA.SpASDepartamento

```

```

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

```

```

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

```

```

SELECT @TerServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

```

```

SELECT      A.UNegocioID,
            A.DescUN,
            C.DepartamentoID,
            C.DescDpt,
            TipoExtension=2,
            CONVERT(varchar(11), C.SocioID) as Clave,
            CONVERT(varchar(73), C.DescDpt) as Descripción,
            D.CommutadorID,
            D.Extension
FROM        #UNegocio A,
            #UNeg_Depto B,
            #Departamento C,
            Extension_Departamento D

```

```

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = D.DepartamentoID

```

```

UNION
SELECT      A.UNegocioID,
            A.DescUN,
            E.DepartamentoID,
            E.DescDpt,
            TipoExtension=1,
            CONVERT(varchar(11), C.SocioID) as Clave,
            CONVERT(varchar(73), C.Apaterno + ' ' + C.AMaterno) as Descripción,
            D.CommutadorID,
            D.Extension
FROM        #UNegocio A,
            #UNeg_Depto B,
            #Socio C,
            Extension_Socio D,
            #Departamento E

```

```

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID

```

```

AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID

```

```

UNION
SELECT      A.UNegocioID,
            A.DescUN,
            E.DepartamentoID,
            E.DescDpt,
            TipoExtension=3,
            CONVERT(varchar(11), C.AreaGenericaID) as Clave,
            CONVERT(varchar(73), C.Descripción) as Descripción,
            D.CommutadorID,
            D.Extension
FROM        #UNegocio A,
            #UNeg_Depto B,
            AreaGenerica C,
            Extension_AreaGenerica D,
            #Departamento E

```

```

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = E.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

```

```

SELECT      A.UNegocioID,
            A.DescUN,
            A.DepartamentoID,
            A.DescDpt,
            A.TipoExtension,
            A.Clave,
            A.Descripción,
            D.TipoLlamadaID,
            D.Descripción as DescTipoLL,
            SUM(Case
                WHEN ISNULL(E.TipoLlamadaID, -1) = -1 THEN
                    1
                ELSE
                    0
            END) AS Alerta,
            Count(*) as Cantidad,
            CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
            SUM(Case
                when D.TipoTarifa = 1 then
                    D.Tarifa
                when D.TipoTarifa = 2 then
                    D.Tarifa * CEILING(B.Duracion / 60) +
            @TerServMedio
            )

```

```

    ELSE
        0
    END) as Importe
    /CONVERT((decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
FROM
    #Llamadas
    #Extensiones A,
    Llamada B,
    Llamada_Telmx C,
    Tipo_Llamada D,
    Extension_TipoLL E
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND D.TipoLlamadaID = @TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal

AND B.Llamada_TelmxID = C.Llamada_TelmxID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.Descrip, A.DepartamentoID, A.Descrip, A.TipoExtension, A.Clave,
A.Descrip, D.TipoLlamadaID, D.Descrip

SET NOCOUNT OFF

IF @Ordenamiento = 1
    SELECT *
    FROM #Llamadas
    ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Descripcion

IF @Ordenamiento = 2
    SELECT *
    FROM #Llamadas
    ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, Descripcion

IF @Ordenamiento = 3
    SELECT *
    FROM #Llamadas
    ORDER BY Importe DESC, Cantidad DESC, Tiempo DESC, Descripcion

GO

***** Object: Stored Procedure dbo.spRepTopoSocio Script Date: 05/05/2000 8:53:47 PM *****
CREATE PROCEDURE spRepTopoSocio
    (@SocioID varchar(11),
    @TipoLlamadaID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10),
    @Ordenamiento smallint)
AS
    /Nombre Pkco: spRepTopoSocio
    / Autor: Hctor Manuel Garza Garza
    / Fecha: 1 de Abril 2000
    / Proyecto: Tarificacion
    / Descripcion: Muestra el top de llamadas por socio
    /

    Declare @FechaIni datetime,
    @FechaFin datetime,
    @TarServMedido decimal(10,4)

    SET @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = datediff(day,1,CONVERT(DATETIME, @FechaFin, 103))

    SET NOCOUNT ON

    SELECT
        @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
        @FechaFin = datediff(day,1,CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, Fchahac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.SocioID,
    A.APaterno + ' ' + A.AMaterno + ' ' + A.Nombre as Nombre,
    B.CommutadorID,
    B.Extension
    #Extensiones
    #Socio A,
    Extension_Socio B

INTO
FROM
    #Socio A,
    Extension_Socio B

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT
    A.SocioID,
    A.Nombre,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
            1
        ELSE
            0
    END) as Alerta,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
    @TarServMedido
    ELSE
        0
    END) as Importe
    /CONVERT((decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
FROM
    #Llamadas
    #Extensiones A,
    Llamada B,
    Llamada_Telmx C,
    Tipo_Llamada D,
    Extension_TipoLL E
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal

AND B.Llamada_TelmxID = C.Llamada_TelmxID
AND B.TipoLlamadaID = @TipoLlamadaID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = F.Telefono
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, F.TelefonoID
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, B.Telefono, Nombre

SET NOCOUNT OFF

IF @Ordenamiento = 1
    SELECT *
    FROM #Llamadas
    ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Telefono, Nombre

IF @Ordenamiento = 2
    SELECT *
    FROM #Llamadas
    ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, Telefono, Nombre

IF @Ordenamiento = 3

```

```

GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Nombre

SET NOCOUNT OFF

IF @Ordenamiento = 1
    SELECT *
    FROM #Llamadas
    ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Nombre

IF @Ordenamiento = 2
    SELECT *
    FROM #Llamadas
    ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, Nombre

IF @Ordenamiento = 3
    SELECT *
    FROM #Llamadas
    ORDER BY Importe DESC, Cantidad DESC, Tiempo DESC, Nombre

GO

***** Object: Stored Procedure dbo.spRepTopoSocioDetalle Script Date: 05/05/2000 8:53:47 PM *****
CREATE PROCEDURE spRepTopoSocioDetalle
    (@SocioID varchar(11),
    @TipoLlamadaID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10),
    @Ordenamiento smallint)
AS
    /Nombre Pkco: spRepTopoSocioDetalle
    / Autor: Hctor Manuel Garza Garza
    / Fecha: 8 de Abril 2000
    / Proyecto: Tarificacion
    / Descripcion: Muestra el top de llamadas por socio
    /

    Declare @FechaIni datetime,
    @FechaFin datetime,
    @TarServMedido decimal(10,4)

    SET NOCOUNT ON

    SELECT
        @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
        @FechaFin = datediff(day,1,CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, Fchahac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpASocio

SELECT @TarServMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.SocioID,
    A.APaterno + ' ' + A.AMaterno + ' ' + A.Nombre as Nombre,
    B.CommutadorID,
    B.Extension
    #Extensiones
    #Socio A,
    Extension_Socio B

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT
    A.SocioID,
    A.Nombre,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    B.Telefono,
    ISNULL(F.TelefonoID,-1) as Permido,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadaID,-1) = -1 THEN
            1
        ELSE
            0
    END) as Alerta,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
    @TarServMedido
    ELSE
        0
    END) as Importe
    /CONVERT((decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe"/
INTO
FROM
    #Llamadas
    #Extensiones A,
    Llamada B,
    Llamada_Telmx C,
    Tipo_Llamada D,
    Extension_TipoLL E
WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaInicial
AND B.Fecha < @FechaFinal

AND B.Llamada_TelmxID = C.Llamada_TelmxID
AND B.TipoLlamadaID = @TipoLlamadaID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = F.Telefono
GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, F.TelefonoID
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, B.Telefono, Nombre

SET NOCOUNT OFF

IF @Ordenamiento = 1
    SELECT *
    FROM #Llamadas
    ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Telefono, Nombre

IF @Ordenamiento = 2
    SELECT *
    FROM #Llamadas
    ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, Telefono, Nombre

IF @Ordenamiento = 3

```



```

SELECT *
FROM #Llamadas
ORDER BY Importe DESC, Cantidad DESC, Tiempo DESC, Telefono, Nombre

GO

/***** Object: Stored Procedure dbo.spRepTopoSocioGraf Script Date: 05/05/2000 6:53:47 PM *****/
CREATE PROCEDURE spRepTopoSocioGraf
    (@SocioID varchar(11),
    @TipoLlamadaID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*=====
Nombre Pkico: spRepTopoSocioGraf
Autor: Hctor Manuel Garza Garza
Fecha: 20 de Abril 2000
Proyecto: Tarificacion
Descripcion: Muestra el top de llamadas por socio
para reporte grafico
*/

Declare @FechaIni datetime,
        @FechaFin datetime,
        @TarServMedio decimal(10,4),
        @Tiempo real,
        @Importe real

SET NOCOUNT ON

SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpAISocio

SELECT @TarServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.SocioID,
    A.APaterno + ' * ' + A.AMaterno + ' * ' + A.Nombre as Nombre,
    B.ComunadorID,
    B.Extension,
    #Extensiones
    #Socio A,
    Extension_Socio B

INTO
FROM

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT
    A.SocioID,
    A.Nombre,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    B.Telefono,
    ISNULL(F.TelefonoID, -1) as Permiso,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
    @TarServMedio
    else
        0
    END) as Importe
    /*CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as Importe*/

INTO
FROM
    #Extensiones A,
    Llamada B,
    Llamada_Telmx C,
    Tipo_Llamada D,
    Tel_Permiso F

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.Fecha >= @FechaIni
AND B.Fecha < @FechaFin

AND B.Llamada_TelmxID = C.Llamada_TelmxID
AND B.TipoLlamadaID = @TipoLlamadaID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = F.Telefono

GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, F.TelefonoID
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, Telefono, Nombre

SELECT (@Cantidad = SUM(Cantidad),
        @Tiempo = SUM(Tiempo),
        @Importe = SUM(Importe))
FROM #Llamadas

SET NOCOUNT OFF

SELECT
    A.SocioID,
    A.Nombre,
    A.Telefono,
    CASE @Cantidad
        WHEN 0 THEN 0
        ELSE ISNULL(A.Cantidad / @Cantidad * 100, 0)
    END as PCantidad,
    CASE @Tiempo
        WHEN 0 THEN 0
        ELSE ISNULL(A.Tiempo / @Tiempo * 100, 0)
    END as PTiempo,
    CASE @Importe
        WHEN 0 THEN 0
        ELSE ISNULL(A.Importe / @Importe * 100, 0)
    END as PImporte

FROM #Llamadas A
GO

/***** Object: Stored Procedure dbo.spRepTopoSocioTelefono Script Date: 05/05/2000 6:53:47 PM *****/
CREATE PROCEDURE spRepTopoSocioTelefono
    (@SocioID varchar(11),
    @TipoLlamadaID smallint,
    @Telefono varchar(20),
    @FechaIni varchar(10),
    @FechaFin varchar(10))
AS
/*=====
Nombre Pkico: spRepTopoSocioTelefono
Autor: Hctor Manuel Garza Garza
Fecha: 1 de Abril 2000
Proyecto: Tarificacion
Descripcion: Muestra el top de llamadas por socio
para reporte grafico
*/

Declare @FechaIni datetime,
        @FechaFin datetime,
        @TarServMedio decimal(10,4),
        @Tiempo real,
        @Importe real

SET NOCOUNT ON

SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint
NULL)

INSERT INTO #UNegocio
EXEC EMPRESA.SpAISUNegocio

CREATE TABLE #UNeg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #UNeg_Departamento
EXEC EMPRESA.SpAISUNeg_Departamento

```

```

/* Proyecto: Tarificacion
/* Descripcion: Muestra el top de llamadas por socio
a detalle de telefonos marcados

/*=====
Nombre Pkico: spRepTopoSocioTelefono
Autor: Hctor Manuel Garza Garza
Fecha: 20 de Abril 2000
Proyecto: Tarificacion
Descripcion: Muestra el top de llamadas por socio
para reporte grafico
*/

Declare @FechaIni datetime,
        @FechaFin datetime,
        @TarServMedio decimal(10,4),
        @Tiempo real,
        @Importe real

SET NOCOUNT ON

SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID
varchar(10) NULL, Nombre varchar(25) NULL, APaterno varchar(20) NULL, FechaNac datetime NULL,
AMaterno varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpAISocio

SELECT @TarServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

SELECT
    A.SocioID,
    A.APaterno + ' * ' + A.AMaterno + ' * ' + A.Nombre as Nombre,
    B.ComunadorID,
    B.Extension,
    #Extensiones
    #Socio A,
    Extension_Socio B

INTO
FROM

WHERE A.SocioID = B.SocioID
AND (A.SocioID = @SocioID OR @SocioID = "ALLSOC")

SELECT
    A.SocioID,
    A.Nombre,
    D.TipoLlamadaID,
    D.Descripcion as DescTipoLL,
    B.Telefono,
    ISNULL(F.TelefonoID, -1) as Permiso,
    SUM(Case
        when ISNULL(E.TipoLlamadaID, -1) = -1 THEN
            1
        ELSE
            0
    END) as Alerta,
    CONVERT(varchar(10), B.Fecha, 103) + ' * ' + CONVERT(varchar(8), B.Fecha, 108) as
Fecha,
    CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
    @TarServMedio
    else
        0
    END) as Importe
    /*CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as Importe*/

INTO
FROM
    #Extensiones A,
    Llamada B,
    Llamada_Telmx C,
    Tipo_Llamada D,
    Extension_TipoLL E,
    Tel_Permiso F

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND B.Fecha >= @FechaIni
AND B.Fecha < @FechaFin

AND B.Llamada_TelmxID = C.Llamada_TelmxID
AND B.TipoLlamadaID = @TipoLlamadaID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = F.Telefono

GROUP BY A.SocioID, A.Nombre, D.TipoLlamadaID, D.Descripcion, B.Telefono, B.Fecha, F.TelefonoID
ORDER BY Tiempo DESC, Importe DESC, Fecha

SET NOCOUNT OFF

SELECT *
FROM #Llamadas
GO

/***** Object: Stored Procedure dbo.spRepTopoTelDetalle Script Date: 05/05/2000 6:53:47 PM *****/
CREATE PROCEDURE spRepTopoTelDetalle
    (@UNegocioID varchar(5),
    @DepartamentoID smallint,
    @TipoExtensionID smallint,
    @Clave varchar(11),
    @TipoLlamadaID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10),
    @Ordenamiento smallint)
AS
/*=====
Nombre Pkico: spRepTopoTelDetalle
Autor: Hctor Manuel Garza Garza
Fecha: 20 de Abril 2000
Proyecto: Tarificacion
Descripcion: Muestra el top de llamadas por socio
para reporte grafico
*/

Declare @FechaIni datetime,
        @FechaFin datetime,
        @TarServMedio decimal(10,4),
        @ClaveAreaGen int

SET NOCOUNT ON

SELECT
    @FechaIni = CONVERT(DATETIME, @FechaIni, 103),
    @FechaFin = dateadd(day, 1, CONVERT(DATETIME, @FechaFin, 103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint
NULL)

INSERT INTO #UNegocio
EXEC EMPRESA.SpAISUNegocio

CREATE TABLE #UNeg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #UNeg_Departamento
EXEC EMPRESA.SpAISUNeg_Departamento

```



```

A.DescUN,
E.DepartamentoD,
E.DescOpt,
TipoExtension1,
CONVERT(varchar(11),C.SocioD) as Clave,
CONVERT(varchar(73),C.Apellido + ' ' + C.Apellido + ' ' + C.Nombre) as Descripción,
D.CommutadorD,
D.Extension,
#Negocio A,
#Neg_Departamento B,
#Socio C,
Extension_Socio D,

#Departamento E
WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID

AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID
AND @TipoExtensionID = 1
AND (C.SocioID = @Clave OR @Clave = "ALL")

UNION
SELECT
A.UNegocioID,
A.DescUN,
E.DepartamentoD,
E.DescOpt,
TipoExtension3,
CONVERT(varchar(11),C.ArreGenericaD) as Clave,
CONVERT(varchar(73),C.ArreGenericaD) as Descripción,
D.CommutadorID,
D.Extension

FROM
#Negocio A,
#Neg_Departamento B,
ArreGenerica C,
Extension_ArreGenerica D,

#Departamento E
WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND (B.DepartamentoID = @DepartamentoID OR @DepartamentoID = "ALLDPT")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = E.DepartamentoID
AND B.DepartamentoID = C.DepartamentoID
AND C.ArreGenericaID = D.ArreGenericaID
AND @TipoExtensionID = 3
AND (C.ArreGenericaID = @ClaveArreGen OR @Clave = "ALL")

SELECT
A.UNegocioID,
A.DescUN,
A.DepartamentoD,
A.DescOpt,
A.TipoExtension,
A.Clave,
A.Descripción,
D.TipoLlamadaD,
D.Descripción as DescTipoLL,
B.Telefono,
CONVERT(varchar(10),B.Fecha,103) + ' ' + CONVERT(varchar(8),B.Fecha,108) as
Fecha,
ISNULL(F.TelefonoD,-1) as PermisoD,
SUM(Case
WHEN ISNULL(E.TipoLlamadaD,-1) = -1 THEN
1
ELSE
0
END) AS Alerta,
CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
SUM(Case
when D.TipoTarifa = 1 then
D.Tarifa
when D.TipoTarifa = 2 then
D.Tarifa * CEILING(B.Duracion / 60) +
@TarServMedio
else
0
END) as Importe
/CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/Y
INTO
FROM
#Llamadas
#TEExtensiones A,
Llamada B,
Llamada_Telnet C,
Tipo_Llamada D,
Extension_TipoLL E,
Tel_Permiso F

WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND D.TipoLlamadaID = @TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin

AND B.Llamada_TelnetID = C.Llamada_TelnetID
AND B.TipoLlamadaID = D.TipoLlamadaID
AND B.Telefono = @Telefono
AND B.Telefono = F.Telefono
GROUP BY A.UNegocioID, A.DescUN, A.DepartamentoD, A.DescOpt, A.TipoExtension, A.Clave,
A.Descripción, D.TipoLlamadaID, D.Descripción, B.Telefono, B.Fecha, F.TelefonoD

SET NOCOUNT OFF

IF @Ordenamiento = 1
SELECT *
FROM #Llamadas
ORDER BY Tiempo DESC, Importe DESC, Fecha, Telefono

IF @Ordenamiento = 2
SELECT *
FROM #Llamadas
ORDER BY Tiempo DESC, Importe DESC, Fecha, Telefono

IF @Ordenamiento = 3
SELECT *
FROM #Llamadas
ORDER BY Importe DESC, Tiempo DESC, Fecha, Telefono

GO

***** Object: Stored Procedure dbo.spRepTopoNegocio Script Date: 05/05/2000 8:53:47 PM *****
CREATE PROCEDURE spRepTopoNegocio
@UNegocioID varchar(5),
@TipoLlamadaID smallint,
@FechaIni varchar(10),
@FechaFin varchar(10),
@Ordenamiento smallint)
AS
/*****
Nombre Físico: spRepTopoNegocio
Autor: Hctor Manuel Garza Garza
Fecha: 1 de Abril 2000
*****/

```

```

/* Proyecto: Tarificación
/* Descripción: Muestra el top de llamadas por Unegocio */

DECLARE @FechaInicio datetime,
@FechaFin datetime,
@TarServMedio decimal(10,4)

SET NOCOUNT ON

SELECT
@FechaInicio = CONVERT(datetime,@FechaIni,103),
@FechaFin = datediff(day,1,CONVERT(datetime,@FechaFin,103))

CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UbicacionID smallint NULL)

INSERT INTO #UNegocio
EXEC EMPRESA.SpAllNegocio

CREATE TABLE #UNeg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)

INSERT INTO #UNeg_Departamento
EXEC EMPRESA.SpAllNeg_Departamento

CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL, ApellidoM varchar(20) NULL, Mail varchar(50) NULL)

INSERT INTO #Socio
EXEC EMPRESA.SpAllSocio

SELECT @TarServMedio = (SELECT Tarifa FROM Tipo_Llamada WHERE Prejio = NULL and TipoTarifa = 1)

SELECT
A.UNegocioID,
A.DescUN,
D.CommutadorID,
D.Extension

INTO #TEExtensiones

FROM
#Negocio A,
#Neg_Departamento B,
Extension_Departamento D

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = D.UNegocioID
AND B.DepartamentoID = D.DepartamentoID

UNION
SELECT
A.UNegocioID,
A.DescUN,
D.CommutadorID,
D.Extension

FROM
#Negocio A,
#Neg_Departamento B,
#Socio C,
Extension_Socio D

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND C.SocioID = D.SocioID

UNION
SELECT
A.UNegocioID,
A.DescUN,
D.CommutadorID,
D.Extension

FROM
#Negocio A,
#Neg_Departamento B,
ArreGenerica C,
Extension_ArreGenerica D

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.ArreGenericaID = D.ArreGenericaID

SELECT
A.UNegocioID,
A.DescUN,
D.TipoLlamadaID,
D.Descripción as DescTipoLL,
SUM(Case
WHEN ISNULL(E.TipoLlamadaD,-1) = -1 THEN
1
ELSE
0
END) AS Alerta,
COUNT(*) as Cantidad,
CONVERT(decimal(10,2),SUM(B.Duracion / 60)) as Tiempo,
SUM(Case
when D.TipoTarifa = 1 then
D.Tarifa
when D.TipoTarifa = 2 then
D.Tarifa * CEILING(B.Duracion / 60) +
@TarServMedio
else
0
END) as Importe
/CONVERT(decimal(20,2),ISNULL(SUM(C.Importe),0)) as Importe/Y
INTO
FROM
#Llamadas
#TEExtensiones A,
Llamada B,
Llamada_Telnet C,
Tipo_Llamada D,
Extension_TipoLL E

WHERE A.CommutadorID = B.CommutadorID
AND A.Extension = B.Extension
AND B.CommutadorID = E.CommutadorID
AND B.Extension = E.Extension
AND B.TipoLlamadaID = E.TipoLlamadaID
AND D.TipoLlamadaID = @TipoLlamadaID
AND B.Fecha >= @FechaInicio
AND B.Fecha < @FechaFin
AND B.Llamada_TelnetID = C.Llamada_TelnetID
AND B.TipoLlamadaID = D.TipoLlamadaID
GROUP BY A.UNegocioID, A.DescUN, D.TipoLlamadaID, D.Descripción

SET NOCOUNT OFF

IF @Ordenamiento = 1
SELECT *
FROM #Llamadas
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, UNegocioID

IF @Ordenamiento = 2
SELECT *
FROM #Llamadas
ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, UNegocioID

IF @Ordenamiento = 3
SELECT *

```

```

FROM #Llamadas
ORDER BY Importe DESC, Cantidad DESC, Tiempo DESC, UNegocioID

GO

/***** Object: Stored Procedure dbo.spRepTopNegocioDetalle Script Date: 05/05/2000 8:53:47 PM *****/
CREATE PROCEDURE spRepTopNegocioDetalle
    (@UNegocioID varchar(5),
    @TipoLlamadoID smallint,
    @FechaIni varchar(10),
    @FechaFin varchar(10),
    @Ordenamiento smallint)
AS
    /*****
    / Nombre Pkco: spRepTopNegocioDetalle
    / Autor: Hctor Manuel Garza Garza
    / Fecha: 8 de Abril 2000
    / Proyecto: Tarificación
    / Descripción: Muestra el top de llamadas por UNegocio */
    /*****/

    DECLARE @FechaIniLocal datetime,
            @FechaFinLocal datetime,
            @TarSenMedido decimal(10,4)

    SET NOCOUNT ON

    SELECT
        @FechaIniLocal = CONVERT(DATETIME, @FechaIni, 103),
        @FechaFinLocal = datetrunc(day, 1, CONVERT(DATETIME, @FechaFin, 103))

    CREATE TABLE #UNegocio (UNegocioID varchar(5) NULL, DescUN varchar(50) NULL, UnicaocioID smallint NULL)
    INSERT INTO #UNegocio
    EXEC EMPRESA.SpAUUNegocio

    CREATE TABLE #UNeg_Departamento (UNegocioID varchar(5) NULL, DepartamentoID varchar(10) NULL)
    INSERT INTO #UNeg_Departamento
    EXEC EMPRESA.SpAUUNeg_Departamento

    CREATE TABLE #Departamento (DepartamentoID varchar(10) NULL, DescOpt varchar(50) NULL)
    INSERT INTO #Departamento
    EXEC EMPRESA.SpAUDepartamento

    CREATE TABLE #Socio (UNegocioID varchar(5) NULL, SocioID varchar(11) NULL, DepartamentoID varchar(10) NULL, Nombre varchar(25) NULL, Apellido varchar(20) NULL, FechaNac datetime NULL,
    Atelemo varchar(20) NULL, Mail varchar(50) NULL)
    INSERT INTO #Socio
    EXEC EMPRESA.SpAUSocio

    SELECT @TarSenMedido = (SELECT Tarifa FROM Tipo_Llamada WHERE Prefijo = NULL and TipoTarifa = 1)

    SELECT
        A.UNegocioID,
        A.DescUN,
        C.DepartamentoID,
        C.DescOpt,
        D.ComunadorID,
        D.Extension
    INTO #TEstensiones
    FROM
        #UNegocio A,
        #UNeg_Departamento B,
        #Departamento C,
        Extension_Departamento D

    WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
    AND A.UNegocioID = B.UNegocioID
    AND B.UNegocioID = D.UNegocioID
    AND B.DepartamentoID = C.DepartamentoID
    AND B.DepartamentoID = D.DepartamentoID

    UNION

    SELECT
        A.UNegocioID,
        A.DescUN,
        E.DepartamentoID,
        E.DescOpt,
        D.ComunadorID,
        D.Extension
    FROM
        #UNegocio A,
        #UNeg_Departamento B,
        #Socio C,
        Extension_Socio D,
        #Departamento E

    WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
    AND A.UNegocioID = B.UNegocioID
    AND B.UNegocioID = C.UNegocioID
    AND B.DepartamentoID = E.DepartamentoID
    AND B.DepartamentoID = C.DepartamentoID
    AND C.SocioID = D.SocioID

    UNION

    SELECT
        A.UNegocioID,
        A.DescUN,
        E.DepartamentoID,

```

```

E.DescOpt,
D.ComunadorID,
D.Extension
FROM
    #UNegocio A,
    #UNeg_Departamento B,
    AreaGenerica C,
    Extension_AreaGenerica D,
    #Departamento E

WHERE (A.UNegocioID = @UNegocioID OR @UNegocioID = "ALLUN")
AND A.UNegocioID = B.UNegocioID
AND B.UNegocioID = C.UNegocioID
AND B.DepartamentoID = C.DepartamentoID
AND B.DepartamentoID = E.DepartamentoID
AND C.AreaGenericaID = D.AreaGenericaID

SELECT
    A.UNegocioID,
    A.DescUN,
    A.DepartamentoID,
    A.DescOpt,
    D.TipoLlamadoID,
    D.Descripcion as DescTipoll,
    SUM(Case
        WHEN ISNULL(E.TipoLlamadoID, -1) = -1 THEN
            1
        ELSE
            0
    END) AS Alerta,
    Count(*) as Cantidad,
    CONVERT(decimal(10,2), SUM(B.Duracion / 60)) as Tiempo,
    SUM(Case
        when D.TipoTarifa = 1 then
            D.Tarifa
        when D.TipoTarifa = 2 then
            D.Tarifa * CEILING(B.Duracion / 60) +
            0
    END) as Importe
    /CONVERT(decimal(20,2), ISNULL(SUM(C.Importe), 0)) as Importe */
INTO
FROM
    #TEstensiones A,
    Llamada B,
    Llamada_Telmax C,
    Tipo_Llamada D,
    Extension_TipoLL E

WHERE A.ComunadorID = B.ComunadorID
AND A.Extension = B.Extension
AND B.ComunadorID = E.ComunadorID
AND B.Extension = E.Extension

AND B.TipoLlamadoID = E.TipoLlamadoID
AND D.TipoLlamadoID = @TipoLlamadoID
AND B.Fecha >= @FechaIniLocal
AND B.Fecha < @FechaFinLocal
AND B.Llamada_TelmaxID = C.Llamada_TelmaxID
AND B.TipoLlamadoID = D.TipoLlamadoID
GROUP BY A.UNegocioID, A.DescUN, A.DepartamentoID, A.DescOpt, D.TipoLlamadoID, D.Descripcion

SET NOCOUNT OFF

/* @Ordenamiento = 1
SELECT *
FROM #Llamadas
ORDER BY Cantidad DESC, Tiempo DESC, Importe DESC, DepartamentoID

/* @Ordenamiento = 2
SELECT *
FROM #Llamadas
ORDER BY Tiempo DESC, Cantidad DESC, Importe DESC, DepartamentoID

/* @Ordenamiento = 3
SELECT *
FROM #Llamadas
ORDER BY Importe DESC, Cantidad DESC, Tiempo DESC, DepartamentoID

GO

/***** Object: Stored Procedure dbo.spSeguridadOpcionPerfil Script Date: 05/05/2000 8:53:47 PM *****/
CREATE PROCEDURE spSeguridadOpcionPerfil
    (@OpcionID smallint,
    @PerfilID smallint)
AS
    /*****
    / Nombre Pkco: spSeguridadOpcionPerfil
    / Autor: Hctor Manuel Garza Garza
    / Fecha: 20 de Abril 2000
    / Proyecto: Tarificación
    / Descripción: Valida que la opcion este habilitada para el perfil
    /*****/

    DECLARE @AuxOpcion SMALLINT

    SELECT
        @AuxOpcion = OpcionID
    FROM Perfil_Opcion
    WHERE OpcionID = @OpcionID
    AND PerfilID = @PerfilID

    IF @AuxOpcion = @OpcionID
    SELECT Resultado = 1
    ELSE
    SELECT Resultado = 0

    GO

```

ASP's

```
<!--include file="edoxbs.inc"-->
<!--
***** Nombre del Autor: Hector Manuel Gerza Gerza.
***** Nombre del Programa: Defaul.asp *****
***** Fecha Creacion: 2/04/2000.
***** Descripción: Pantalla de inicio.
-->

<html>
<head>
<title>Sistema de Tarificación de la Red de Voz HEB México</title>
</head>

<body Script Language="VBScript" bgcolor="#E7E7D6" onLoad="document.loginfrm.Login.Focus()">

<CENTER>
<TABLE WIDTH="90%" border="0">
<TR>
 <TD WIDTH="50%"> <center> <p align="Center"> </center> </TD> <TD> <center> <p align="Center"> </center> </TD> </TR> <TR>  <TD colspan="2"> <center> <font color="#000000" size="3" face="Verdana, Arial, Helvetica"> <br> <br> <strong>Supermercados Internacionales HEB, S.A de C.V.</Strong> <br> <br> <strong>INTRANET H-E-B </strong> <br> </center> </td> </tr> </TABLE>  <br> <br> <FORM NAME="loginfrm" ACTION="Valida_Usuario.asp" METHOD=POST> <TABLE WIDTH=300 BORDER=1 CELLSPACING=0 ALIGN=CENTER> <TR>  <TD BGCOLOR="#336699" COLSPAN = "2"> <FONT FACE="Verdana, Helvetica" SIZE=2 COLOR = "WHITE"><CENTER><B>VALIDACION DE USUARIO</B></CENTER></FONT> </TD> </TR> <TR>  <TD WIDTH=50 ALIGN=LEFT VALIGN=TOP> <FONT FACE="Verdana, Helvetica" SIZE=2><B>Login</B></FONT> </TD> <TD> <INPUT TYPE=TEXT NAME="Login" SIZE=10 MAXLENGTH=10 VALUE=""%>=<Request.Form("Login") %> </TD> </TR> <TR>  <TD WIDTH=50 ALIGN=LEFT VALIGN=TOP> <FONT FACE="Verdana, Helvetica" SIZE=2><B>Password</B></FONT> </TD> <TD> <INPUT TYPE=PASSWORD NAME="Password" SIZE=10 MAXLENGTH=10> </TD> </TR> </TABLE> <BR> <TABLE border="1"> |  |  | | --- | --- | | <TD width="50"> </TD> <TD> <center> <INPUT TYPE="SUBMIT" VALUE="Aceptar"> </center> </TD> | | | <center> <INPUT TYPE="Reset" Value="Cancelar"> </center> | <TD width="50"> </TD> | | |    </FORM> <BODY> </HTML>  <%@ Language="VBScript"%> <% Response.Buffer = True %> <!-- ***** Nombre del Autor: Hector Manuel Gerza Gerza. ***** Nombre del Programa: Menu.asp ***** ***** Fecha Creacion: 21/04/2000. ***** Descripción: Crea el menu dinamicamente ***** Dependencia: Defaul.asp --> | | | | | | | | | |
```

```

<!--#include file = "Lienas_combo.asp" -->
<!--#include file = "seguridad.asp" -->

<%
usuario = "sa"
password = "sa"
set con = server.createObject("ADODB.Connection")
stron = "DSN=Terntificador,UID=" & usuario & "PWD=" & password & ";DATABASE= CORPORATIVA"
con.open stron
%>

<center>

</center>

<center>
<font color="black" face="Tahoma" size="5">
<div style="text-align: center;">
</div>
</center>

<tr color="Red">
<td colspan="2">
<form Action = "procesa_opcion.asp" Name = "FrmMenu" Method = "Post">

<center>
<table>
<tr>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "AGrupacion" Value="Cat" Checked>
Categorías
</div>
</td>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<% Call Lienas_Combo ("spRepOptionsTiposAgrupacion 1,1," &
request.cookies("TarUsuario")(Par1),"OpcCat","Click_OpcCat()", "Categorias") %>
</div>
</td>
</tr>
<tr>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "AGrupacion" Value="Rep">
Reportes
</div>
</td>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<% Call Lienas_Combo ("spRepOptionsTiposAgrupacion 1,2," &
request.cookies("TarUsuario")(Par1),"OpcRep","Click_OpcRep()", "Reportes") %>
</div>
</td>
</tr>
<tr>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "AGrupacion" Value="Sol">
Soluciones
</div>
</td>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<% Call Lienas_Combo ("spRepOptionsTiposAgrupacion 1,3," &
request.cookies("TarUsuario")(Par1),"OpcSol","Click_OpcSol()", "Soluciones") %>
</div>
</td>
</tr>
<tr>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "AGrupacion" Value="UI">
Utilidades
</div>
</td>
<td>
<div style="text-align: center;">
<font Face = "Tahoma" Size = "2" Color = "Black">
<% Call Lienas_Combo ("spRepOptionsTiposAgrupacion 1,4," &
request.cookies("TarUsuario")(Par1),"OpcUI","Click_OpcUI()", "Utilidades") %>
</div>
</td>
</tr>
</table>
</center>

<div style="text-align: center;">
<table border="1">
<tr>
<td width="50">
<div>
<input Type = "Button" Name = "cmdAceptar" Value = "Aceptar" Onclick = "Actualizar()">
</div>
<div>
<input Type = "Button" Name = "cmdRegresar" Value = "<< Regresar" Onclick =
"Regresar()">
</div>
<td width="50">
<div>
</div>
</td>
</tr>
</table>
</center>
<div style="text-align: center;">
<div style="border: 1px solid black; width: 100px; height: 20px; margin: 0 auto;">
```


[illegible]

```
%>
If LlamadoID_8 <> "<!--" and LlamadoTID_8 <> "<!--" then
    eqquery = "declare @call varchar(255) exec spCreaCallPrimaria
M,SpUpdConciliacionManual " & LlamadoID_8 & "," & LlamadoTID_8 & ", @call out select @call 'r'"
    set resp = conn.execute(eqquery)
    eqquery = resp("r")
    set rcConn = conn.execute(eqquery)
end if

If LlamadoID_9 <> "<!--" and LlamadoTID_9 <> "<!--" then
    eqquery = "declare @call varchar(255) exec spCreaCallPrimaria
M,SpUpdConciliacionManual " & LlamadoID_9 & "," & LlamadoTID_9 & ", @call out select @call 'r'"
    set resp = conn.execute(eqquery)
    eqquery = resp("r")
    set rcConn = conn.execute(eqquery)
end if

If LlamadoID_10 <> "<!--" and LlamadoTID_10 <> "<!--" then
    eqquery = "declare @call varchar(255) exec spCreaCallPrimaria
M,SpUpdConciliacionManual " & LlamadoID_10 & "," & LlamadoTID_10 & ", @call out select @call 'r'"
    set resp = conn.execute(eqquery)
    eqquery = resp("r")
    set rcConn = conn.execute(eqquery)
end if

%>

<Script Language="VBScript">
<--
MagBox "Se actualizaron los registros", vbInformation, "Información"

-->
</Script>

<End If %>

</Form>

<Script Language="VBScript">
<--
Sub Conciliar()
If Document.Forms.Fini.Value <="" and Document.Forms.FFin.Value
If EtfFecha(Document.Forms.Fini.Value) and
EtfFecha(Document.Forms.FFin.Value) then
FechaEntera(Document.Forms.Fini.Value) <= FechaEntera(Document.Forms.FFin.Value) then
Document.forms.valor.value = 1
Document.forms.submit
else
magbox
"Error en rango", vbCritical,"Error"
end if
else
magbox "Error en fechas",
vbCritical,"Error"
end if
End Sub

Sub Actualizar()
Document.forms.valor.value = 2
Document.forms.submit
End Sub

Sub Limpiar()
document.forms.fini.value = fechaextus()
document.forms.fin.value = fechaextus()
End Sub

Sub Regresar()
document.location.href = "MENU.ASP"
End Sub

-->
</Script>

<Script Language="VBScript">
<--
Sub Focused()
document.forms.fini.value = fechaextus()
document.forms.fin.value = fechaextus()
End Sub

-->
</Script>

</Body>
</Html>

<%@ LANGUAGE="VBSCRIPT"%>
<% response.buffer = True %>
<--
***** Nombre del Autor: Hector Manuel Gerza Gerza. *****
***** Nombre del Programa: Conciliacion_Reporte.asp *****
***** Fecha Creacion: 20/04/2009. *****

***** Descripcion: Muestra la conciliacion de llamadas. *****
***** Dependencia: Conciliacion_Carga2.asp *****

-->

<-- #include file = "Llama_combo.asp"-->
<-- #include file = "fechas.asp"-->
<-- #include file = "fechas2.asp"-->
<%#SegOptionID = 0 %>
<-- #include file = "seguridad.asp"-->

<%
Fini = request.form("Fini")
FFin = request.form("FFin")
%>

<html>
<head>
<Title>Reporte de Conciliaci&oscar;n de Llamadas</Title>

<BODY BGCOLOR="#f7e7d8" Script Language="VBScript" OnLoad="focused()">
<center>

</center>
<center>
<font color="black" face="Tahoma" size="5">
<h3>Reporte de Conciliaci&oscar;n de Llamadas</h3>
</font>
</Center>
<br color="Red">
```

```
<Form Action = "Conciliacion_reporte.asp" Name = "Form" Method = "Post">
```

```
<Center>
```

```
%
    usuario = "sa"
    password = "sa"
    set cnn = server.createObject("ADODB.Connection")
    stron = "DSN=Telecomor,UID=" & usuario & ";PWD=" & password
    &"DATABASE= CORPORATIVA"
    cnn.open stron
```

```
sqlquery = "declare @call varchar(255) exec spCreateCallPrimario
16,SpConciliacionInformacion, @call out select @call ' "
set rsap = cnn.execute(sqlquery)
sqlquery = "resp('a)"
response.write sqlquery
set rsConc = cnn.execute(sqlquery)
```

```
%
```

```
<Font Face = "Tahoma" Size = "2" Color = "black">
```

```
Existen <strong><%=rsConc("ComunID") %></strong> llamadas en conciliar desde
<strong><%=rsConc("Fecha") %></strong> hasta <strong><%=rsConc("Fecha") %></strong>
```

```
<br><br>
```

```
<Font>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
Extensi&osculen
```

```
<Font>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```

```
<td>
```



```

        ArDatos =
        Agregal_nro(ArDatos, CreaGlos(10, formatnumber(nrRp("Cantidad"), 2), nrRp("DescUN")))
        and if
        if Valores = "2" then
            ArDatos =
            Agregal_nro(ArDatos, CreaGlos(10, formatnumber(nrRp("Tiempo"), 2), nrRp("DescUN")))
            and if
            if Valores = "3" then
                ArDatos =
                Agregal_nro(ArDatos, CreaGlos(10, formatnumber(nrRp("Importe"), 2), nrRp("DescUN")))
                and if
                nrRp.MoveNext
            end if
        end if
    %>
    <center>
    <applet name=barchart
        codebase=../Ncharts/classes
        code=NFBarChartApp.class
        width=762 height=600>
    <param
        name=NFParamScript values=
    Background
        = (white);
    LabelPos
        = 1, 1;
    Dwell_label
        = (white, SHADOW, 3);
    Pie3DDepth
        = 25;
    PieAngle
        = 10;
    Strokes
        = <%=ArDatos %>;
    </applet>
    </center>

    <%Else %>
    <%If not nrRp.EOF then
        UN = ""
        (TLL = CuenteRegistros(nrTLL))
        if TLL > 0 then
            nrTLL.MoveFirst
            Tipoinicial = nrTLL("TipoLlamadaID")
            while not nrTLL.EOF
                if TipoLlamadaID = "-1" then
                    CorDatos =
                    Agregal_lista(CorDatos, CreaTituloBarras(nrTLL("Descripcion"), ""))
                else
                    CorDatos =
                    Agregal_lista(CorDatos, CreaTituloBarras(nrTLL("Descripcion"), "lightgray"))
                end if
                nrTLL.MoveNext
            end if
            nrRp.MoveFirst
            TLLID = nrRp("TipoLlamadaID")
            i = 1
            While not nrRp.EOF
                if UN <> nrRp("UNegocioID") and Tipoinicial =
                    UN = nrRp("UNegocioID")
                    Barras = Agregal_lista(Barras, chr(34) &
                    nrRp("DescUN") & chr(34))
                    End if
                    if Valores = "1" then
                        Lista = Agregal_lista(Lista, nrRp("Cantidad"))
                    and if
                    if Valores = "2" then
                        Lista = Agregal_lista(Lista, nrRp("Tiempo"))
                    end if
                    if Valores = "3" then
                        Lista = Agregal_lista(Lista, nrRp("Importe"))
                    end if
                    nrRp.MoveNext
                    if not nrRp.EOF then
                        if TLLID <> nrRp("TipoLlamadaID") then
                            ArDatos = ArDatos &
                            i = i + 1
                            TLLID =
                        else
                            and if
                            ArDatos = ArDatos & CreaDataSel((Lista)
                        end if
                    end if
                Wend
            %>

            <%If Valores = 1 then
                Unidad = "Llamadas"
            and if
            if Valores = 2 then
                Unidad = "Minutos"
            and if
            if Valores = 3 then
                Unidad = "Pesos"
            end if
            Barras = replace(Barras, "<img src="">")
            Barras = replace(Barras, "<img src="">")
            Response.write barras
            %>

            <center>
            <applet name=barchart
                codebase=../Ncharts/classes
                code=NFBarChartApp.class
                width=762 height=600>
            <param
                name=NFParamScript
                value =
            Background
                = (white, NONE, 8);
            BottomTitle
                = ("ON", black, "Tahome", 12, 90);
            LeftTitle
                = ("ON", black, "Tahome", 12);
            LeftFormat
                = (FLOAT);
            Dwell_label
                = ("ON", black, "Tahome", 20);
            DwellBox
                = (white, SHADOW, 3);
            BarLabels
                = <%=Barras %>;
            BarWidth
                = 80;
            LeftTitle
                = ("ON", black, "Tahome", 20, 90);
            LeftTitleBox
                = (white, NONE, 5);
            Legend
                = ("ON", black, "Tahome", 14);
            LegendBox
                = (white, NONE, 5);
            LegendLayout
                = (HORIZONTAL, BOTTOM, RIGHT, 0, 0, LEFT, 2);
            GraphType
                = GROUP;
            Bar3DDepth
                = 85;
            DataSets
                = <%=CorDatos %>;
            <%=ArDatos %>

```

```

    >
    </applet>

    <%End if %>
    <%End if %>
    <%Conn.close %>

    <%@ Language=VBScript %>
    <!--
    ***** Nombre del Autor: Hector Manuel Garza Garza.
    *****
    ***** Nombre del Programa: Enc_LlamadasOpto.asp
    *****
    ***** Fecha Creacion: 6/04/2000.
    *****
    ***** Descripción: Encabezado de reporte de llamadas por depto.
    ***** Dependencia: Menu.asp
    *****

    -->

    <% Response.Buffer = True %>
    <%SegOptionID = 20 %>
    <!-- Anclade file = "seguridad.asp" -->
    <!-- Anclade file = "adobe.inc" -->

    <HTML>
    <HEAD>
    <TITLE>Reporte de Llamadas por Departamento </TITLE>
    </HEAD>

    <frameset rows="25%, 75%" border="0">
        <frameset cols="57%, 43%" border="0">
            <frame src="Enc_llamadasopto1.asp" name="Unegocio" noresize scrolling=no>
            <frame src="Enc_llamadasopto2.asp" name="Departamento" noresize scrolling=no>
            </frameset>
            <frame src="Enc_llamadasopto3.asp" name="Tecnos" noresize scrolling=no>
        </frameset>

    <html>

    <%@ LANGUAGE=VBSCRIPT %>
    <% response.buffer = True %>
    <!--
    ***** Nombre del Autor: Hector Manuel Garza Garza.
    *****
    ***** Nombre del Programa: Fie_LlamadasOptoUN.asp
    *****
    ***** Fecha Creacion: 5/04/2000.
    *****
    ***** Descripción: Frames para reporte de
    ***** Dependencia: Menu.asp
    *****

    -->

    <%SegOptionID = 19 %>
    <!-- Anclade file = "seguridad.asp" -->
    <!-- Anclade file = "adobe.inc" -->

    <HTML>
    <HEAD>
    <TITLE>Reporte de Llamadas por Unidades de Negocio</TITLE>
    </HEAD>

    <frameset rows="1050, 1000" border="1">
        <frame src="Enc_LlamadasOptoUN.asp" name="encabezado" scrolling="auto">
        <frame src="Cuerpo.asp" name="cuerpo" scrolling="auto">
    </frameset>

    <html>

    <%@ LANGUAGE=VBSCRIPT %>
    <% response.buffer = True %>
    <!--
    ***** Nombre del Autor: Hector Manuel Garza Garza.
    *****
    ***** Nombre del Programa: Enc_LlamadasOptoUN.asp
    *****
    ***** Fecha Creacion: 6/04/2000.
    *****
    ***** Descripción: Encabezado de reporte de llamadas por Unegocio
    ***** Dependencia: Fie_LlamadasOptoUN.asp
    *****

    -->

    <% SegOptionID = 19 %>
    <!-- Anclade file = "seguridad.asp" -->
    <!-- Anclade file = "tecnos.asp" -->
    <!-- Anclade file = "tecnos2.asp" -->

    <%
    usuario = "sa"
    password = "sa"
    set cnn = server.createObject("ADODB.Connection")
    strcn = "DSN=Tranificador;UID=" & usuario & "&PWD=" & password & ";DATABASE=CORPORATIVA"
    cnn.open strcn
    %>

    <HTML>
    <base target="cuerpo">
    <BODY bgcolor="#339999" Script Language="VBScript" OnLoad="Focused()">
    <Form Name="MyForm" Action="Tip_LlamadasOptoUN.asp" method="post">
    <center>
    <table border="0">
    <tr>
    <td>
    
    </td>
    <td>
    <img alt="30" width="30" height="30">
    </td>
    <td>
    <img alt="30" width="30" height="30">
    </td>
    <td>
    <img alt="30" width="30" height="30">
    </td>
    <td>
    <img alt="30" width="30" height="30">
    </td>
    <td>
    <img alt="30" width="30" height="30">
    </td>
    </tr>
    <tr>
    <td colspan="6">
    <div style="text-align: center; font-size: 12px; font-weight: bold; color: white;>
        14. SeReportaUnegocioUsuario & chr(34) & request.cookies("TerUsuario") & chr(34) & ", @call out
        select @call 'r'
        set rsap = cnn.execute(sqquery)
        sqquery = rsap("r")
        set nlNegocio = cnn.execute(sqquery)
        %>
        <SELECT Name=UnegocioID>
        <OPTION VALUE="0">-- Unidades de Negocio --
        <If not nlNegocio.EOF then %>
        <If nlNegocio("UN") = "TODAS" then %>
        <OPTION
        VALUE="ALLUN">-- Todas --
        <End if %>
        <End if %>
        <While not nlNegocio.EOF %>
        <OPTION
        VALUE="<=nlNegocioID>UnegocioID">--<=nlNegocioID>UnegocioID">
        <nlNegocio.Description %>
        <nlNegocio.Movement %>
        <Wend %>
        </SELECT>

```



```
</?>
<br>
<?>
    <?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input type="radio" name="TipoFecha" value="Dur" checked=Odecole.s>
            </?>
        </?>
        <?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Fecha Inicial:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "FIni" Size = "10" Maxlength = "10"
            Value=""%=<FechaActual%>" OnClick=document.myform.TipoFecha(0).checked=true"
            onKeyPress="FechaKeyPress(&)> dd/mm/yyyy
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Fecha Final:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "FFin" Size = "10" Maxlength = "10"
            Value=""%=<FechaActual%>" OnClick=document.myform.TipoFecha(1).checked=true"
            onKeyPress="FechaKeyPress(&)> dd/mm/yyyy
            </?>
        </?>
    </?>
    <br>
    <?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input type="radio" name="TipoFecha" value="Sem" checked=Semens Fiscal
            </?>
        </?>
        <?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Semena Inicial:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "SIni" Size = "10" Maxlength = "7"
            Value=""%=<SemenaActual%>" OnClick=document.myform.TipoFecha(1).checked=true"
            onKeyPress="FechaKeyPress(&)> ss/ssss
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Semena Final:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "SFin" Size = "10" Maxlength = "7"
            Value=""%=<SemenaActual%>" OnClick=document.myform.TipoFecha(1).checked=true"
            onKeyPress="FechaKeyPress(&)> ss/ssss
            </?>
        </?>
    </?>
    <br>
    <?>
        <?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input type="radio" name="TipoFecha" value="Mes">Mes
            </?>
        </?>
        <?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Mes Inicial:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "MIni" Size = "10" Maxlength = "7"
            Value=""%=<Md(FechaActual),4,7)>" OnClick=document.myform.TipoFecha(2).checked=true"
            onKeyPress="FechaKeyPress(&)> mm/ssss
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Mes Final:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "MFin" Size = "10" Maxlength = "7"
            Value=""%=<Md(FechaActual),4,7)>" OnClick=document.myform.TipoFecha(2).checked=true"
            onKeyPress="FechaKeyPress(&)> mm/ssss
            </?>
        </?>
    </?>
    <br>
    <?>
        <?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input type="radio" name="TipoFecha" value="Bim">Bimestre
            </?>
        </?>
        <?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Bimestre Inicial:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "BIni" Size = "10" Maxlength = "8"
            Value=""%=<BimestraActual%>" OnClick=document.myform.TipoFecha(3).checked=true"
            onKeyPress="FechaKeyPress(&)> b/sss
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            Bimestre Final:
            </?>
        </?>
        <Font Face = "Tahoma" Size = "7" Color = "White">
            <input Type = "Text" Name = "BFin" Size = "10" Maxlength = "8"
            Value=""%=<BimestraActual%>" OnClick=document.myform.TipoFecha(3).checked=true"
            onKeyPress="FechaKeyPress(&)> b/sss
            </?>
        </?>
    </?>
    <br>
    <?>
    </Center>
    <br>
    <Center>
    <table border="1">
```

```

</?>
<td width = "50">
</td>
<td>
<input Type = "button" Name = "cmdReporte" Value = "Reporte" OnClick = "Reporte()">
</td>
<td>
<input Type = "button" Name = "cmdClear" Value = "Limpiar" OnClick = "Limpiar()">
</td>
<td>
<input Type = "button" Name = "cmdRegresar" Value = "<< Regresar" OnClick =
</td>
</tr>
</table>
</center>
</form>
<Script Language="VBScript">
<!--
Sub Reporte()
Dim fechaIni
Dim fechaFin
If document.myForm.UnegocioID.value <> "0" Then
If document.myForm.TipoFecha(0).checked then
If EsFecha(Document.myForm.Fini.value) and
FechaEntera(Document.myForm.Fini.value) <= FechaEntera(Document.myForm.Fin.value) then
Document.myForm.submit
else
MsgBox
"La fecha inicial debe ser menor o igual que la final",vbExclamation,"Informaciyn"
document.myForm.Fini.focus
end if
else
MsgBox "Error an
Call LimpiarFechaes
document.myForm.Fini.focus
and if
and if document.myForm.TipoFecha(1).checked then
If EsSemana(Document.myForm.Sini.value) and
SemanaEntera(Document.myForm.Sini.value) <= SemanaEntera(Document.myForm.SFin.value) then
Document.myForm.submit
else
MsgBox
"La semana inicial debe ser menor o igual que la final",vbExclamation,"Informaciyn"
document.myForm.Sini.focus
end if
else
MsgBox "Error an
Call LimpiarFechaes
document.myForm.Sini.focus
and if
and if document.myForm.TipoFecha(2).checked then
If EsMes(Document.myForm.Mini.value) and
MesEntero(Document.myForm.Mini.value) <= MesEntero(Document.myForm.MFin.value) then
Document.myForm.submit
else
MsgBox
"El Mes inicial debe ser menor o igual que la final",vbExclamation,"Informaciyn"
document.myForm.Mini.focus
end if
else
MsgBox "Error an
Call LimpiarFechaes
document.myForm.Mini.focus
and if
and if document.myForm.TipoFecha(3).checked then
If EsBimestre(Document.myForm.Bini.value) and
BimestreEntero(Document.myForm.Bini.value) <= BimestreEntero(Document.myForm.BFin.value) then
Document.myForm.submit
else
MsgBox
"El Bimestre inicial debe ser menor o igual que la final",vbExclamation,"Informaciyn"
document.myForm.Bini.focus
end if
else
MsgBox "Error an
Call LimpiarFechaes
document.myForm.Bini.focus
and if
and if
End Sub
End If
Sub LimpiarFechaes()
Call LimpiarFechaes
Document.myForm.Fini.value = FechaActual()
Document.myForm.Fin.value = FechaActual()
Document.myForm.Sini.value = ""
Document.myForm.SFin.value = ""
Document.myForm.Mini.value = MesActual()
Document.myForm.MFin.value = MesActual()
Document.myForm.Bini.value = BimestreActual()
Document.myForm.BFin.value = BimestreActual()
End Sub
Sub Limpiar()
Document.myForm.UnegocioID.value = "0"
Call LimpiarFechaes
End Sub
Sub Regresar()
top.location = "MENU.ASP"
End Sub
-->
</Script>
<Script Language="VBScript">

```



```
<title>
</title>
</center>
</tr>

<% orn close %>

<Script Language=“VBScript”>
<--
Sub Atres()
history back
End Sub
-->
</Script>

</orn>
</BODY>
</HTML>
<%@ LANGUAGE=“VBSRIPIT” %>
<% Response.Buffer = True %>
<--
***** Nombre del Autor: Hector Manuel Garza Garza.
*****
***** Nombre del Programa: Rpt_LlamadosDeploDetalle.asp *****
***** Fecha Creacion: 75/04/2000. *****
*****
***** Descripcion: Reporte de Llamadas por Departamento (Detalle) *****
***** Dependencias: Rpt_LlamadosLNDetalle.asp *****
-->

<% SetQconntID = 18 %>
<-- &Include file = “seguridad.asp” -->
<-- &Include file = “fechas.asp” -->
<-- &Include file = “techas2.asp” -->

<%
usuario = “sa”
password = “sa”
set con = server.createObject(“ADODB.Connection”)
stron = “DSN=Leticadora;UID=& usuario &”;PWD=”& password &”;DATABASE=CORPORATIVA”
con.open stron
%>

<%
UnegocioID = request.querystring(“UNID”)
DepartamentoID = request.querystring(“DPT”)
Fini = request.querystring(“Fin”)
FFin = request.querystring(“FFin”)
response.write UnegocioID & “ ” & DepartamentoID & “ ” & Fini & “ ” & FFin
%>

<%
sqlquery = “declare @call varchar(255) exec spCreaCellPrimario 16, ‘SpRepLlamadosDeploDetalle ’ & chr(34) & UnegocioID & chr(34) & “ ” & chr(34) & DepartamentoID & chr(34) & “ ” & chr(34) & Fini & chr(34) & “ ” & chr(34) & FFin & chr(34) & “ ”. @call out select substring(@call, 1, 150)”
set rsap = con.execute(sqlquery)
sqlquery = “rsap”&t”
set nrsq = con.execute(sqlquery)
%>

<form action=“Rpt_LlamadosDeploDetalle.asp” method=“POST” target=“cuerpo”>
<INPUT type=“button” name=“cmdAtres” value=“< Atres” onclick=“js” onColor=“Atres()”>
<br>
<br>
</center>
<Table Border = “1”>
<br>
<tbl_struct>
<tbl_header>
| <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Reporte de Llamadas por Unidades de Negocio (Detalle Departamentos) </strong></center> </th> |
| --- |







| <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Reporte de Llamadas por Unidades de Negocio (Detalle Departamentos) </strong></center> </th>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <br> <br> <tbl_struct> <tbl_header> <tr> <th> &lt;font face=“Tahoma” size=“2” color=“NAVY”&gt; &lt;center&gt;&lt;strong&gt; Da: &lt;%=FechaNormal(Fini)%&gt; Hasta: &lt;%=FechaNormal(FFin)%&gt; &lt;/strong&gt;&lt;/center&gt; &lt;/th&gt; </th></tr> <tbl_info cols="1"></tbl_info> <tbl_r cells="1" ix="1" maxcspan="1" maxrspan="1" usedcols="1"></tbl_r> <tbl_r cells="1" ix="2" maxcspan="1" maxrspan="1" usedcols="1"></tbl_r> <tbl_r cells="1" ix="3" maxcspan="1" maxrspan="1" usedcols="1"></tbl_r> <tbl_r cells="1" ix="4" maxcspan="1" maxrspan="1" usedcols="1"></tbl_r> <tbl_r cells="1" ix="5" maxcspan="1" maxrspan="1" usedcols="1"></tbl_r> <table border="1"> <thead> <tr> <th> &lt;font face=“Tahoma” size=“2” color=“NAVY”&gt; &lt;center&gt;&lt;strong&gt; Da: &lt;%=FechaNormal(Fini)%&gt; Hasta: &lt;%=FechaNormal(FFin)%&gt; &lt;/strong&gt;&lt;/center&gt; &lt;/th&gt;</th></tr> </thead> <tbody> <tr> <td> &lt;br&gt; &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“RED”&gt; Clave UN &lt;/font&gt; &lt;/td&gt;</td></tr> <tr> <td> &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“RED”&gt; Unidad de Negocio &lt;/font&gt; &lt;/td&gt;</td></tr> <tr> <td> &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“RED”&gt; Clave Deplo. &lt;/font&gt; &lt;/td&gt;</td></tr> <tr> <td> &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“RED”&gt; Departamento &lt;/font&gt; &lt;/td&gt;</td></tr> <tr> <td> &lt;br&gt; &lt;%if not nrsq.EOF then %&gt; &lt;br&gt; &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“Black”&gt; &lt;%= nrsq(“UnegocioID”) %&gt; &lt;/font&gt; &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“Black”&gt; &lt;%= nrsq(“DescUN”) %&gt; &lt;/font&gt; &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“Black”&gt; &lt;%= nrsq(“DepartamentoID”) %&gt; &lt;/font&gt; &lt;br&gt; &lt;font face=“Tahoma” size=“1” color=“Black”&gt; &lt;%= nrsq(“DescDpt”) %&gt; &lt;/font&gt; &lt;br&gt; &lt;%end if %&gt; &lt;/td&gt;</td></tr> </tbody> </table> | <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Da: <%=FechaNormal(Fini)%> Hasta: <%=FechaNormal(FFin)%> </strong></center> </th> | <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Da: <%=FechaNormal(Fini)%> Hasta: <%=FechaNormal(FFin)%> </strong></center> </th> | <br> <br> <font face=“Tahoma” size=“1” color=“RED”> Clave UN </font> </td> | <br> <font face=“Tahoma” size=“1” color=“RED”> Unidad de Negocio </font> </td> | <br> <font face=“Tahoma” size=“1” color=“RED”> Clave Deplo. </font> </td> | <br> <font face=“Tahoma” size=“1” color=“RED”> Departamento </font> </td> | <br> <%if not nrsq.EOF then %> <br> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“UnegocioID”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DescUN”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DepartamentoID”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DescDpt”) %> </font> <br> <%end if %> </td> |
| <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Da: <%=FechaNormal(Fini)%> Hasta: <%=FechaNormal(FFin)%> </strong></center> </th>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <font face=“Tahoma” size=“2” color=“NAVY”> <center><strong> Da: <%=FechaNormal(Fini)%> Hasta: <%=FechaNormal(FFin)%> </strong></center> </th>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <br> <br> <font face=“Tahoma” size=“1” color=“RED”> Clave UN </font> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <br> <font face=“Tahoma” size=“1” color=“RED”> Unidad de Negocio </font> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <br> <font face=“Tahoma” size=“1” color=“RED”> Clave Deplo. </font> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <br> <font face=“Tahoma” size=“1” color=“RED”> Departamento </font> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |
| <br> <%if not nrsq.EOF then %> <br> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“UnegocioID”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DescUN”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DepartamentoID”) %> </font> <br> <font face=“Tahoma” size=“1” color=“Black”> <%= nrsq(“DescDpt”) %> </font> <br> <%end if %> </td>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                               |                                                                                                                                               |                                                                            |                                                                                |                                                                           |                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                     |


```


[illegible][illegible]

[illegible]

```

If EstMes(Document.myForm.Mini.Value) then
    If EstMes(Document.myForm.Mini.Value) and
        MesEntre(Document.myForm.Mini.Value) <= MesEntre(Document.myForm.MFin.Value) then
            Document.myForm.submit
        else
            MsgBox
    "El Mes inicial debe ser menor o igual que le final",vbExclamation,"Informaciyn"
    document.myForm.mini.focus
    else
        MsgBox,"Error en
        Call LimpiaFechas
        document.myForm.mini.focus
    end If
    If document.myForm.TipoFecha(3).checked then
        If EstBimestre(Document.myForm.Bini.Value)
            and EstBimestre(Document.myForm.BFin.Value) then
                BimestreEntre(Document.myForm.Bini.Value) <= BimestreEntre(Document.myForm.BFin.Value) then
                    Document.myForm.submit
                else
                    MsgBox
    "El Bimestre inicial debe ser menor o igual que le final",vbExclamation,"Informaciyn"
    document.myForm.bini.focus
    else
        MsgBox,"Error en
        Call LimpiaFechas
        document.myForm.bini.focus
    end If
    end If
End If

End Sub

Sub LimpiaFechas()
    Call LimpiaFechas
    Document.myForm.Fini.Value = FechaActual()
    Document.myForm.FFin.Value = FechaActual()
    Document.myForm.Sini.Value = ""
    Document.myForm.SFin.Value = ""
    Document.myForm.Mini.Value = MesActual()
    Document.myForm.MFin.Value = MesActual()
    Document.myForm.Bini.Value = BimestreActual()
    Document.myForm.BFin.Value = BimestreActual()
End Sub

Sub Limpia()
    Document.myForm.SocioID.Value = "0"
    Call LimpiaFechas
End Sub

Sub Registrar()
    top.location = "MENU.ASP"
End Sub

-->
</Script>

<Script Language="VBScript">
<--
    Sub focused()
    End Sub
-->
</Script>

<% onn.close %>

</body>
</html>
<%@ LANGUAGE="VBSCRIPT" %>
<% response.buffer = True %>
<--
***** Nombre del Autor: Hector Manuel Cerza Cerza.
*****
***** Nombre del Programa: Rpt_Tel%>Socio.asp *****
***** Fecha Creacion: 25/04/2000.
*****
***** Descripcion: Reporte de Llamadas a Telefonos no involucrados*****
***** Dependencia: Enc_Tel%>Socio.asp *****
-->

<% SegOptionID = 25 %>
<-- #include file = "seguridad.asp" -->
<-- #include file = "fechas.asp" -->
<-- #include file = "fechas2.asp" -->

<%
usuario = "sa"
password = "sa"
set con = server.createobject("ADODB.connection")
strcon = "DSN=Irtificador;UID=" & usuario & " & PWD=" & password & ";DATABASE=CORPORATIVA"
con.open strcon
%>

<%
SocioID = request.form("SocioID")
Fini = request.form("Fini")
FFin = request.form("FFin")
response.write UnagocioID & " & DepartamentoID & " & Fini & " & FFin
con.open strcon
%>

TipoFecha = request.form("TipoFecha")
Sini = request.form("Sini")
SFin = request.form("SFin")
Mini = request.form("Mini")
MFin = request.form("MFin")
Bini = request.form("Bini")
BFin = request.form("BFin")
Select Case TipoFecha
    Case "Sem"
        Fini = FechaSemana(Sini,true)
        FFin = FechaSemana(SFin,false)
    Case "Mes"
        Fini = FechaMes(Mini,true)
        FFin = FechaMes(MFin,false)
    Case "Bim"
        Fini = FechaBimestre(Bini,true)
        FFin = FechaBimestre(BFin,false)
    Case Else
        Fini = Fini
        FFin = FFin
End Select
%>

```



```

<%
sqquery = "declare @call varchar(255) exec sp_CreaCallPrimerio 16, 'SpRepTelbSocio' & chr(34) & SocioID
& chr(34) & ' ' & chr(34) & Fini & chr(34) & ' ' & chr(34) & FFin & chr(34) & ' '. @call out select substring(@call,
1, 150) 'r'
set rsp = cnn.execute(sqquery)
sqquery = 'rsp'
response.write sqquery
set nRpt = cnn.execute(sqquery)
%>

<form action="Rpt_TelbSocioDetalle.asp" method="POST" target="cuerpo">
<br>
<table border="1">
<br>
<td colspan="8">
<font face="Tahome" size="2" color="NAVY">
<center><strong>
Reporte de Teléscu, Jono No Involucrados con la Empresa por Socio
</strong></center>
<font>
<br>
<br>
<td colspan="8">
<font face="Tahome" size="2" color="NAVY">
<center><strong>
De: <%=Fini %> Hasta: <%=FFin %>
</strong></center>
<font>
<br>
<br>
<table border="1">
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Clave Socio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Socio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Tipo de Servicio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Teléscu, Jono
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
No. Llamadas
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Minutos
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Importe Total
<font>
<br>
<br>
<br>
<%
SOCIO = ""
NOMSOC = ""
TSERV = ""
%>

<% While not nRpt.EOF %>
<br>
<br>
COLOR="Black">
nRpt("SocioID")%>
nRpt("Nombre") %>
nRpt("DescTipOLL") %>
COLOR="Black">
nRpt("Telefono")%>
LigaAgregarTelefono nRpt("Telefono")%>

```

```

<td align="RIGHT">
COLOR="Black">
formalnumber(nRpt("Tiempo"),2)%>
COLOR="Black">
formalnumber(nRpt("Importe"),2)%>
<br>
<table border="1">
<br>
<td colspan="8">
<font face="Tahome" size="2" color="NAVY">
<center><strong>
Reporte de Teléscu, Jono No Involucrados con la Empresa por Socio (Detalle)
</strong></center>
<font>
<br>
<br>
<td colspan="8">
<font face="Tahome" size="2" color="NAVY">
<center><strong>
De: <%=FechaFormat(Fini) %> Hasta: <%=FechaFormat(FFin) %>
</strong></center>
<font>
<br>
<br>
<table border="1">
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Clave Socio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Socio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Tipo de Servicio
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Teléscu, Jono
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
No. Llamadas
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Minutos
<font>
<br>
<td colspan="8">
<font face="Tahome" size="1" color="RED">
Importe Total
<font>
<br>
<br>
<br>
<%
SOCIO = ""
NOMSOC = ""
TSERV = ""
%>

<% While not nRpt.EOF %>
<br>
<br>
COLOR="Black">
nRpt("SocioID")%>
nRpt("Nombre") %>
nRpt("DescTipOLL") %>
COLOR="Black">
nRpt("Telefono")%>
LigaAgregarTelefono nRpt("Telefono")%>

```

```

<FONT FACE="Tahoma" SIZE="11" COLOR="RED">
Tipo de Servicio
</FONT>
<AD>
<id colspan="2">
<FONT FACE="Tahoma" SIZE="11" COLOR="RED">
Teléfono de Inicio
</FONT>
<AD>
<AD>
<FONT FACE="Tahoma" SIZE="11" COLOR="RED">
Fecha
</FONT>
<AD>
<AD>
<FONT FACE="Tahoma" SIZE="11" COLOR="RED">
Minutos
</FONT>
<AD>
<AD>
<FONT FACE="Tahoma" SIZE="11" COLOR="RED">
Importe Total
</FONT>
<AD>
<AD>

<NTSERV = ""
TEL = ""
%

<% While not rsRpt.EOF %>

<AD>

SIZE="11" COLOR="White">
SIZE="11" COLOR="Black">

rsRpt("DescTipodLL") %>

COLOR="Black">

rsRpt("Telefono") %>

COLOR="Black">

rsRpt("Telefono") then %>
LigaAgregarTelefono rsRpt("Telefono") %>
rsRpt("Telefono") %>

<AD>
<id align="RIGHT">
COLOR="Black">
<AD>
<id align="RIGHT">
COLOR="Black">
formatnumber(rsRpt("Tiempo"),2)%>
<AD>
<id align="RIGHT">
COLOR="Black">
formatnumber(rsRpt("Importe"),2)%>
<AD>
<id align="RIGHT">
</id>
<rsRpt.MovedNext %>
</Wend %>
</Table>
</Center>
</>

<% ctm.close %>
<Script Language="VBScript">
<!--
Sub Alinea()
History.back
End Sub
-->
</Script>
</form>
</BODY>
</HTML>
<%@ LANGUAGE="VBSCRIPT" %>
<% response.buffer = True %>
<!--
***** Nombre del Autor: Hector Manuel Garza Gerza.
***** Nombre del Programa: Fria_TelbSocioGraf.asp
***** Fecha Creacion: 5/04/2000.
***** Descripción: Fries para reporte grafico de tele no involucrados.
***** Dependencia: Fria_TelbSocioGraf.asp
*****
-->
<%SegOptionID = 31 %>
<!-- @include file = "seguridad.asp" -->
<!-- @include file = "adobcs.inc" -->

<HTML>
<HEAD>
<TITLE>Reporte Grafico de Tele No Involucrados con la Empresa por Socio</TITLE>
</HEAD>

<frameset rows="1280,1000" border="1">
<frame src="Enc_TelbSocioGraf.asp" name="encabezado" scrolling="auto">
<frame src="Cuerpo.asp" name="cuerpo" scrolling="auto">

```

```

</frameset>
</asp>
<%@ LANGUAGE="VBSCRIPT" %>
<% response.buffer = True %>
<!--
***** Nombre del Autor: Hector Manuel Garza Gerza.
***** Nombre del Programa: Fria_TelbSocioGraf.asp
***** Fecha Creacion: 5/04/2000.
***** Descripción: Encabezado de reporte de tele no involucrados (grafico).
***** Dependencia: Fria_TelbSocioGraf.asp
*****
-->
<!-- @include file = "fria_combo.asp" -->
<!-- @include file = "fechas.asp" -->
<!-- @include file = "fechas2.asp" -->

<% SegOptionID = 31 %>
<!-- @include file = "seguridad.asp" -->

<%
usuario = "sa"
password = "sa"
set ctm = server.createObject("ADODB.Connection")
stron = "DSN=TelbSocioGraf;UID=" & usuario & ";PWD=" & password & ";DATABASE=CORPORATIVA"
ctm.open stron
%

<HTML>
<base target="cuerpo">
<BODY BGCOLOR="#333333" Script Language="VBScript" OnLoad = "Focused()">

<Form Name="MyForm" Action="Rpt_TelbSocioGraf.asp" method="post">
<Center?>
<TABLE border="0">
<tr>
<td>

<td colspan="2">
<div width="30">
<div colspan="1">
<Font Face = "Tahoma" Size = "2" Color = "White">
Socio:
</Font>
<div colspan="3">
<%
sqlquery = "declare @call varchar(255) exec spCreaCallPrimerio 14,SpRepSocioUsuario
" & chr(34) & request.cookies("TelUsuario") & chr(34) & " ", @call out select @call " "
set nssp = ctm.execute(sqlquery)
sqlquery = nssp("sql")
set nSocio = ctm.execute(sqlquery)
%"
<SELECT Name="SocioID">
<OPTION VALUE="0">-- Socio --
<%if not rsSocio.EOF then %>
<%if rsSocio("Soc") = "TODOS" then %>
<OPTION
VALUE="ALLSOC">" Todos "
<%End if %>
<%While not rsSocio.EOF %>
<OPTION
VALUE="&rsSocio("SocioID")&">&rsSocio("Apellido") & " " & rsSocio("Apellido") & " " &
rsSocio("Nombre") %>
<%rsSocio.MovedNext%>
</SELECT>
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
Tipo de Servicio:
</Font>
<div colspan="2">
<% Call Llena_Combo ("SpAtTipodLlamada","TipoLlamadaID","Tipo de Servicio") %>
<div colspan="1">
<Font Face = "Tahoma" Size = "2" Color = "White">
<INPUT type="radio" value="1" name="Valores" CHECKED>Llamadas
<INPUT type="radio" value="2" name="Valores">Tiempos
<INPUT type="radio" value="3" name="Valores">Pasos
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
Fecha Inicial:
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
<input type="text" name="Fini" Size = "10" Maxlength = "10"
value="&rsFechaActual%&" OnClick="document.myform.TipoFecha(0).checked=true"
onKeyPress="FechaKeyPress(1)">
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
Fecha Final:
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
<input type="text" name="FFini" Size = "10" Maxlength = "10"
Value="&rsFechaActual%&" OnClick="document.myform.TipoFecha(0).checked=true"
onKeyPress="FechaKeyPress(1)">
</td>
<td>
<td>
<td>
<td>
<Font Face = "Tahoma" Size = "2" Color = "White">
<input type="radio" name="TipoFecha" value="Sem" Semana Fiscal
</td>

```


[illegible]

```

and if
and if
If document.myform.TipoFecha(2) checked then
If EsMes(Document.myForm.Mini.Value) and
End if
End if
End Sub

Sub LimpierFechas
Call LimpierFechas
document.myForm.mini.focus
End Sub

Sub Regresar()
top location = "MENU.ASP"
End Sub

</Script>

<Script Language="VBScript">
<--
Sub focused()
End Sub
-->
</Script>

<% onn.close %>

</body>
</html>

<%@ LANGUAGE="VBSCRIPT" %>
<% response.buffer = True %>
<--
***** Nombre del Autor: Hector Manuel Garza Garza.
*****
***** Nombre del Programa: Rpt_TopoUn.asp *****
***** Fecha Creacion: 25/04/2000. *****
***** Descripción: Reporte de Top por Unidad de Negocio *****
***** Dependencia: Eric_TopoUn.asp *****
-->

<% SegOptionID = 36 %>
<-- #include file = "seguridad.asp" -->
<-- #include file = "fechas.asp" -->
<-- #include file = "fechas2.asp" -->

<%
usuario = "sa"
password = "sa"
set cnn = server.createObject("ADODB.Connection")
cnn.Open "DSN=Trinitador;UID=" & usuario & " & PWD=" & password & " & DATABASE= CORPORATIVA"
on error resume next

<%
UnNegocioID = request.form("UnNegocioID")
Valores = request.form("Valores")
TipoLamadaID = request.form("TipoLamadaID")
Fini = request.form("Fini")
FFin = request.form("FFin")

TipoFecha = request.form("TipoFecha")
Sini = request.form("Sini")
SFin = request.form("SFin")
Mini = request.form("Mini")
MFin = request.form("MFin")
Bini = request.form("Bini")
BFin = request.form("BFin")
Select Case TipoFecha
Case "Sem"
Fini = FechaSemana(Sini,true)
FFin = FechaSemana(SFin,false)
Case "Mes"
Fini = FechaMes(Mini,true)
FFin = FechaMes(MFin,false)
Case "Bim"
Fini = FechaBimestre(Bini,true)
FFin = FechaBimestre(BFin,false)

```

[illegible]

```
%@ LANGUAGE=VBSCRIPT %>
% response buffer = True %>
<!--
***** Nombre del Autor: Hector Manuel Garza Garza. *****
***** Nombre del Programa: Rpt_TopsLNDetalle.asp *****
***** Fecha Creacion: 25/04/2000. *****

***** Descripción: Reporte de Top por Unidad de Negocio (Detalle) *****
***** Dependencia: Rpt_TopsLN.asp *****

-->

<% SetOptionID = 36 %>
<!-- #include file = "seguridad.asp" -->
<!-- #include file = "fechas.asp" -->
<!-- #include file = "fechas2.asp" -->

<%
usuario = "sa"
password = "sa"
set cnn = server.createObject("ADODB.Connection")
strcon = "DSN=Trilitecador;UID=" & usuario & ";PWD=" & password & ";DATABASE=CORPORATIVA"
cnn.open strcon
%>

<%
UnNegocioID = request.querystring("UNID")
TipoLlamadaID = request.querystring("TLL")
Valores = request.querystring("ORD")
Fini = request.querystring("Fin")
FFin = request.querystring("FFin")
%>

<%
sqlquery = "declare @cat varchar(255) exec sp_CreCatPrimario 18, 'SpRepTopsLNNegocioDetalle '" & chr(34) & UnNegocioID & chr(34) & "','" & TipoLlamadaID & "','" & chr(34) & Fini & chr(34) & "','" & chr(34) & FFin & chr(34) & "','" & Valores & "','" & chr(34) & cat & chr(34) & "'"
set rsap = cnn.execute(sqlquery)
sqlquery = rsap("rs")
set nrRpt = cnn.execute(sqlquery)
%>

<form action="Rpt_LlamedasDeptoDetalle.asp" method="POST" target="cuerpo">
<INPUT type="button" name="cmdAtiras" value="<< Atir &ecute,s" onClick="Atiras()"/>
<br>
<Center>
<Table border = "1">
<br>
<td colspan="1">
<font face="Tahoma" size="2" color="NAVY">
<center><strong>
Reporte de Top por Unidades de Negocio (Detalle)
</strong></center>
</font>
<br>
<br>
<td colspan="1">
<font face="Tahoma" size="2" color="NAVY">
<center><strong>
Dev: <%=FechaNormal(Fini) %> Hasta: <%=FechaNormal(FFin) %>
</strong></center>
</font>
<br>
<br>
<td colspan="1">
<font face="Tahoma" size="2" color="NAVY">
<center><strong>
Tipo de Servicio <%=nrRpt("DescTipoLL") %>
</strong></center>
</font>
<br>
<br>
<td colspan="1">
<font face="Tahoma" size="1" color="RED">
Clave UN
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
Unidad de Negocio
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
Clave Depto.
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
Departamento
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
No. Llamadas
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
Minutos
</font>
<br>
<br>
<font face="Tahoma" size="1" color="RED">
Importe Total
</font>
<br>
<br>
<tblinfo=""
DESC = ""
DPT = ""
DESCOPT = ""
%>

<% While not nrRpt.EOF %>
<br>
<br>
<font face="Tahoma" size="1"
color="Black">
<br>
<br>
<tblinfo=""
DESC = ""
DPT = ""
DESCOPT = ""
%>
```

```
<!-->  
COLOR=Black>  
  
%  
  
COLOR=Black>  
  
nre=Rpt_TopDepctoDetalle.asp?UNID=%&nRpt(LUnegocioID)%&DPT=%&nRpt(DepartamentoID)%&FFin=&%Fini-%&FFIn-&%Fin%&tLL=%&TipoLlamadoID%&ORD=%&Valores%>  
nRpt("DepartamentoID"%)  
  
nRpt("DepartamentoID"%)  
  
%OPT =  
  
%END IF %>  
  
%</p><br/>  
%</d align="RIGHT">  
COLOR=Black>  
  
%Font Face="Tahoma" SIZE="1"  
%</font>  
%NMT DESC OPT < nRpt("DESCOPT") then %>  
%<nRpt("DescOpt"%)>  
%NODESCOPT =  
  
%END IF %>  
%</font>  
%<d align="RIGHT">  
COLOR=Black>  
  
%Font Face="Tahoma" SIZE="1"  
%</font>  
%FORMNUMBER(nRpt(Tiempo),2)%>  
  
%</d align="RIGHT">  
COLOR=Black>  
  
%Font Face="Tahoma" SIZE="1"  
%</font>  
formnumber(nRpt(Importe),2)%>  
  
%</d align="RIGHT">  
COLOR=Black>  
  
%Font Face="Tahoma" SIZE="1"  
%</font>  
formnumber(nRpt(Importe),2)%>  
  
%</d align="RIGHT">  
%</p>  
%<nRpt.MoveNext %>  
  
%</wend %>  
  
%</Table>  
%<Center>  
%<br>  
  
%<pnn.close %>  
  
%<Script Language=VBScript>  
%<- Sub Area()  
history.back  
End Sub  
-->  
%</Script>  
  
%</Form>  
%</BODY>  
%</HTML>  
%<BEG LANGUAGE=VBSOCPRT %>  
%<response.buffer = True %>  
%<->  
***** Nombre del Autor: Hector Manuel Gerza Gerza.  
*****  
***** Nombre del Programa: Rpt_TopDepctoDetalle.asp  
*****  
***** Fecha Creacion: 25/04/2000.  
*****  
***** Descripcion: Reporte de Top por Departamento (Detalle) *****  
***** Dependencia: Rpt_TopUndatale.asp *****  
-->  
  
%SegOptionID = 36 %>  
%<#include file = "seguridad.asp"-->  
%<#include file = "techas.asp"-->  
%<#include file = "teches2.asp"-->  
  
%<usuario = "sa"  
password = "sa"  
set cnn = server.createObject("ADODB.Connection")  
set cnn.Provider="Microsoft.Jet.OLEDB.4.0"; user="sa;" password="" &"DATABASE=CORPORATIVA"  
cnn.open set cnn  
%>  
  
%<TipoLlamadoID = request.querystring("TLL")  
Valores = request.querystring("ORD")  
UnegocioID = request.querystring("UNID")  
DepartamentoID = request.querystring("DPT")  
Fini = request.querystring("Fini")  
FFin = request.querystring("FFin")  
response.write UnegocioID & "" & DepartamentoID & "" & Fini & "" & Ffin  
%>  
  
%<squery = "declare @call varchar(255) exec sp_CreaCall@primario 16, 'SprTopDepctoDetalle' & chr(34) &  
chr(34) & chr(34) & "" & chr(34) & DepartamentoID & chr(34) & "" & TipoLlamadoID & "" & chr(34) & Fini  
& chr(34) & "" & chr(34) & FFin & chr(34) & "" & Valores & "" , @call out select substring(@call, 1, 150)" if  
set rssp = cnn.execute(squery)  
squery = rssp("a")  
set nRpt = cnn.execute(squery)  
%>  
  
%<font action="Rpt_LlamadosDepctoDetalle.asp" method="POST" target="/cuerpo">  
%<INPUT type="button" name="cmdArea" value=""< A&Escule,s" onclick="Area()>  
%<br>  
%<center>  
%<table border = "1">  
%<br>  
%<d collapse="4">  
%<font face="Tahoma" size="7" color="NAVY">  
%<script><br/>
```

```
Reporte de Top por Unidades de Negocio (Detalle Departamentos)
<strong></center>
</FONT>

</Ad>
</P>
</?>
<id collapse="4">
    <FONT FACE="Tahoma" SIZE="2" COLOR="NAVY">
        <center><strong>
            De: <%=FechaNormal(Fin)> %> Hasta: <%=FechaNormal(FFin)> %>
        </strong></center>
    </FONT>

</Ad>
</P>
</?>
<id collapse="4">
    <FONT FACE="Tahoma" SIZE="2" COLOR="NAVY">
        <center><strong>
            Tipo de Servicio: <%=nrRpt("DescTipoLL")%>
        </strong></center>
    </FONT>

</Ad>
</P>
</?>
<id>
    <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
        Clave UN
    </FONT>
</Ad>
</P>
</?>
<id>
    <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
        Unidad de Negocio
    </FONT>
</Ad>
</P>
</?>
<id>
    <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
        Clave Despl.
    </FONT>
</Ad>
</P>
</?>
<id>
    <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
        Departamento
    </FONT>
</Ad>
</P>
</?>
<if not nrPLEOF then %>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="Black">
            <%= nrRpt("UnegocioID")%>
        </font>
    </Ad>
    </P>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="Black">
            <%= nrRpt("DescUN")%>
        </font>
    </Ad>
    </P>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="Black">
            <%= nrRpt("DescOpt")%>
        </font>
    </Ad>
    </P>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="Black">
            <%= nrRpt("DepartamentoID")%>
        </font>
    </Ad>
    </P>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="Black">
            <%= nrRpt("DescOpt")%>
        </font>
    </Ad>
    </P>
    </?>
    </if>
</Table>
</?>

<Table Border="1">
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
            Clave Proprietario
        </FONT>
    </Ad>
    </P>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
            Propietario
        </FONT>
    </Ad>
    </P>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
            No. Llamadas
        </FONT>
    </Ad>
    </P>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
            Minutos
        </FONT>
    </Ad>
    </P>
    <id>
        <FONT FACE="Tahoma" SIZE="1" COLOR="RED">
            Importe Total
        </FONT>
    </Ad>
    </P>
    </?>

<%CLAVE = ""
DESCOCC = ""
%>

<% While not nrPLEOF %>

    <id>
        <FONT FACE="Tahoma" SIZE="1"
        <br/>
        <%If CLAVE <> nrRpt("Clave") then %>
            <br/>
            href="Top_TopDetDetails.asp?UNIT=<%= nrRpt("UnegocioID")>&OPT=<%=nrRpt("DepartamentoID")>
            &TEXT=<%=nrRpt("TipoCategori")> &MOVE=<%=nrRpt("Clave")> &ILL=<%=nrRpt("LlamadasID")>
            &FIR=<%=Fin)>&FIN=<%=Fin)>&ORD=<%=Valores)>
        <br/>
            nrRpt("Clave")%>
        </if>
        <br/>
        <%CLAVE = nrRpt("Clave")
    </Ad>
    </P>
    </?>
    <id>
        <%End if %>
    </font>
    </Ad>
    </P>
    </?>
    <id>
        <FONT FACE="Tahoma" SIZE="1"
        <br/>
        <%If DESCOCC <> nrRpt("Description") then %>
            <br/>
            nrRpt("Description") %>
        </if>
        <br/>
        <%DESCOCC =
    </Ad>
    </P>
    </?>
    <id align="RIGHT">
        <FONT FACE="Tahoma" SIZE="1"
        <br/>
        <%= nrRpt("Cantidad")>
    </Ad>
    </P>
    </?>
</Table>
```


270

```

<input Type = "Text" Name = "ChNombreDisplay" Size="20" MaxLength="20"
value="<%rsSol("DisplayNuevo")%>"
</input>
<div>
(Nombre de gaseite y primer letra de apellido patrono)
</div>
<Script Language = "VBScript">
<!--
If document.forms.ChNombreDisplay.value > "" then
document.forms.CO-HonDisp.checked = true
else
document.forms.CO-HonDisp.checked = false
end if
-->
</Script>
</Form>

</div>
</div>
</div>

<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "ACHCodAut">
OnClick="Desactiva_ACHCodAut"><STRONG>Codigo de Autorizaci</STRONG>
</div>
<div collapse="2">
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "ACSLocal">Local
<input Type = "Checkbox" Name = "ACSNac">Nacional
<input Type = "Checkbox" Name = "ACSInt">Internacional
</div>
<input Type = "HIDDEN" Name = "HAClaseServ" value="<%rsSol("CodigoAut")%>"
<Script Language = "VBScript">
<!--
Select Case document.forms.HAClaseServ.value
Case "1"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = true
document.forms.ACSNac.checked = false
document.forms.ACSInt.checked = false
Case "2"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = false
document.forms.ACSNac.checked = true
document.forms.ACSInt.checked = false
Case "3"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = false
document.forms.ACSNac.checked = false
document.forms.ACSInt.checked = true
Case "4"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = true
document.forms.ACSNac.checked = true
document.forms.ACSInt.checked = false
Case "5"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = true
document.forms.ACSNac.checked = false
document.forms.ACSInt.checked = true
Case "6"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = false
document.forms.ACSNac.checked = true
document.forms.ACSInt.checked = true
Case "7"

document.forms.ACHCodAut.checked = true
document.forms.ACSLocal.checked = true
document.forms.ACSNac.checked = true
document.forms.ACSInt.checked = true
Case else

document.forms.ACHCodAut.checked = false
document.forms.ACSLocal.checked = false
document.forms.ACSNac.checked = false
document.forms.ACSInt.checked = false
end Select
-->
</Script>
</Form>

</div>
</div>

<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "COHRepApe">
OnClick="Desactiva_COHRepApe"><STRONG>Respo</STRONG>
</div>
<div>
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "CApato" VALUE="Ana">Anal</div>
<div>
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Radio" Name = "CApato" VALUE="Dig">Digit</div>
<input Type = "HIDDEN" Name = "HCApato" value="<%rsSol("NuevoAparato")%>"
<Script Language = "VBScript">
<!--
If document.forms.HCApato.value = "1" then
document.forms.CAparato(0).checked = true
document.forms.CO-HRepApe.checked = true
else
-->

```

```

If document.forms.HCApato.value = "2" then
document.forms.CAparato(1).checked = true
document.forms.CO-HRepApe.checked = true
else
document.forms.CO-HRepApe.checked = false
end if
-->
</Script>
</Form>

</div>
</div>

<Font Face = "Tahoma" Size = "2" Color = "Black">
<STRONG>Fecha de Inicio de Labores</STRONG>
</div>
<div collapse="2">
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Text" Name = "AFechaInic" Size="10" MaxLength="10"
value="<%rsSol("Fecha_Inicio")%>"(nuevo ingreso)
</div>
<div>
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "COHModExt">
OnClick="Desactiva_COHModExt"><STRONG>Mover de Lugar una Extensi</STRONG>
</div>
<div>
<Font Face = "Tahoma" Size = "2" Color = "Black">
Nodo Nuevo (azul)
</div>
<div>
<input Type = "Text" Name = "ChNumNodo" Size = "15" MaxLength = "15"
value="<%rsSol("NuevoNodo")%>"
<Script Language = "VBScript">
<!--
If document.forms.ChNumNodo.value > "" then
document.forms.CO-HModExt.checked = true
else
document.forms.CO-HModExt.checked = false
end if
-->
</Script>
</Form>

</div>
</div>

<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "COHDirExt">
OnClick="Desactiva_COHDirExt"><STRONG>Cambio de Dire</STRONG>
</div>
<div>
<Font Face = "Tahoma" Size = "2" Color = "Black">
Numero Actual
</div>
<div>
<input Type = "Text" Name = "ChNumDirExt" Size = "15" MaxLength = "15"
value="<%rsSol("NumDirExt")%>"
<Script Language = "VBScript">
<!--
If document.forms.ChNumDirExt.value > "" then
document.forms.CO-HDirExt.checked = true
else
document.forms.CO-HDirExt.checked = false
end if
-->
</Script>
</Form>

</div>
</div>

<div collapse="3">
<div collapse="3">
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "COHExtDir"><STRONG>Cambio de Extensi</STRONG>
</div>
<div>
<input Type = "HIDDEN" Name = "HCOHExtDir" value="<%rsSol("Ext_directo")%>"
<Script Language = "VBScript">
<!--
If document.forms.HCOHExtDir.value = "1" then
document.forms.COHExtDir.checked = true
else
document.forms.COHExtDir.checked = false
end if
-->
</Script>
</div>
</div>
</div>

<div collapse="3">
<div collapse="3">
<Font Face = "Tahoma" Size = "2" Color = "Black">
<input Type = "Checkbox" Name = "COHBloqLD"><STRONG>Bloquear Larga Distanci</STRONG>
</div>
<div>
<input Type = "HIDDEN" Name = "HCOHBloqLD" value="<%rsSol("BloqueadoLD")%>"
<Script Language = "VBScript">
<!--
If document.forms.HCOHBloqLD.value = "1" then
document.forms.COHBloqLD.checked = true
else
document.forms.COHBloqLD.checked = false
end if
If document.forms.AC(0).checked then
Document.Forms.ACHLinTel.disabled = false
Document.Forms.ACHCodAut.disabled = false
If Document.Forms.ACHLinTel.checked = false
-->
Document.Forms.ALinTel(0).disabled = true
Document.Forms.ALinTel(1).disabled = true
Document.Forms.AIsoNum(0).disabled = true
Document.Forms.AIsoNum(1).disabled = true
-->

```

```

Document.Forme.ANombresDisplay.disabled = true
and if
if Document.Forme.ACHCodAut.checked = false
then
Document.Forme.ACSLocal.disabled = true
Document.Forme.ACSNec.disabled = true
Document.Forme.ACSInt.disabled = true
and if
Document.Forme.AFecheInt.disabled = false
Document.Forme.CCHCodAut.disabled = true
Document.Forme.CCHNonDisp.disabled = true
Document.Forme.CCHRepApe.disabled = true
Document.Forme.CCHMovExt.disabled = true
Document.Forme.CCHDirExt.disabled = true
Document.Forme.CCHBloqLD.disabled = true
Document.Forme.CCodArt.disabled = true
Document.Forme.CCSLocal.disabled = true
Document.Forme.CCSNec.disabled = true
Document.Forme.CCSInt.disabled = true
Document.Forme.CNombresDisplay.disabled = true
Document.Forme.CAparato(0).disabled = true
Document.Forme.CAparato(1).disabled = true
Document.Forme.CNumModo.disabled = true
Document.Forme.CNumAcID.disabled = true
else
Document.Forme.ACHIntTel.disabled = true
Document.Forme.ACHCodAut.disabled = true
Document.Forme.ACHIntTel(0).disabled = true
Document.Forme.ACHIntTel(1).disabled = true
Document.Forme.AlpoNum(0).disabled = true
Document.Forme.AlpoNum(1).disabled = true
Document.Forme.ANombresDisplay.disabled = true
Document.Forme.ACSLocal.disabled = true
Document.Forme.ACSNec.disabled = true
Document.Forme.AFecheInt.disabled = true
Document.Forme.CCHCodAut.disabled = false
Document.Forme.CCHNonDisp.disabled = false
Document.Forme.CCHRepApe.disabled = false
Document.Forme.CCHMovExt.disabled = false
Document.Forme.CCHDirExt.disabled = false
Document.Forme.CCHBloqLD.disabled = false
Document.Forme.CCHCodAut.checked = false
if Document.Forme.CCHCodAut.checked = false
then
Document.Forme.CCodArt.disabled = true
Document.Forme.CCSLocal.disabled = true
Document.Forme.CCSNec.disabled = true
Document.Forme.CCSInt.disabled = true
and if
if Document.Forme.CCHNonDisp.checked = false
then
Document.Forme.CNombresDisplay.disabled = true
and if
if Document.Forme.CCHRepApe.checked = false
then
Document.Forme.CAparato(0).disabled = true
Document.Forme.CAparato(1).disabled = true
and if
if Document.Forme.CCHMovExt.checked = false
then
Document.Forme.CNumModo.disabled = true
and if
if Document.Forme.CCHDirExt.checked = true
then
Document.Forme.CNumAcID.disabled = true
and if
and if
Document.Forme.AC(0).disabled = true
Document.Forme.AC(1).disabled = true
--
</Script>
</>
</Table>
</Center>
<Input Type = "hidden" Name = "Valor">
<br>
</Center>
<Table Border = "1">
<tr>
  |
```

```

valida_datos = false
if Document.Forme.SocioID.value <> "" and
Document.Forme.NumModo.value <> "" then
if Document.Forme.AC(0).checked then
not(Document.Forme.ACHIntTel.checked) and not(Document.Forme.ACHCodAut.checked) then
valida_datos = false
Exit
Function
Document.Forme.ACHIntTel.checked then
and if
if
Document.Forme.ANombresDisplay.value = "" or not(Document.Forme.ACHIntTel(0).checked) and
not(Document.Forme.ACHIntTel(1).checked) or not(Document.Forme.AlpoNum(0).checked) and
not(Document.Forme.AlpoNum(1).checked) then
valida_datos = false
Exit Function
Document.Forme.ACHCodAut.checked then
and if
if
not(Document.Forme.ACSLocal.checked) and not(Document.Forme.ACSNec.checked) and
not(Document.Forme.ACSInt.checked) then
valida_datos = false
Exit Function
Document.Forme.Extension.value = "" and IsNumeric(Document.Forme.Extension.value) then
valida_datos = false
Exit
Function
Document.Forme.CCHCodAut.checked then
and if
if
document.Forme.CCodArt.value = "" or not(Document.Forme.CCSLocal.checked) and
not(Document.Forme.CCSNec.checked) and not(Document.Forme.CCSInt.checked) then
valida_datos = false
Exit Function
Document.Forme.CCHNonDisp.checked then
and if
if
Document.Forme.CNombresDisplay.value = "" then
valida_datos = false
Exit Function
Document.Forme.CCHRepApe.checked then
and if
if
not(Document.Forme.CAparato(0).checked) and not(Document.Forme.CAparato(1).checked) then
valida_datos = false
Exit Function
Document.Forme.CCHMovExt.checked then
and if
if
Document.Forme.CNumModo.value = "" then
valida_datos = false
Exit Function
Document.Forme.CCHDirExt.checked then
and if
if
Document.Forme.CNumAcID.value = "" then
valida_datos = false
Exit Function
End Function
and if
valida_datos = true
Sub Actualizar()
Document.Forme.Solid.disabled = false
Document.Forme.AC(0).disabled = false
Document.Forme.AC(1).disabled = false
if (Document.Forme.AC(1).checked or
Document.Forme.AFecheInt.value) and Valida_datos() then
Document.Forme.ACHIntTel.disabled = false
Document.Forme.ACHCodAut.disabled = false
Document.Forme.ACHIntTel(0).disabled = false
Document.Forme.ACHIntTel(1).disabled = false
Document.Forme.AlpoNum(0).disabled = false
Document.Forme.AlpoNum(1).disabled = false
Document.Forme.ANombresDisplay.disabled =
false
Document.Forme.ACSLocal.disabled = false
Document.Forme.ACSNec.disabled = false
Document.Forme.ACSInt.disabled = false
Document.Forme.AFecheInt.disabled = false
Document.Forme.CCHRepApe.disabled = false
Document.Forme.CCHNonDisp.disabled = false
Document.Forme.CCHMovExt.disabled = false
Document.Forme.CCHCodAut.disabled = false
Document.Forme.CCHDirExt.disabled = false
Document.Forme.CCHBloqLD.disabled = false
Document.Forme.CCodArt.disabled = false
Document.Forme.CCSLocal.disabled = false
Document.Forme.CCSNec.disabled = false
Document.Forme.CCSInt.disabled = false

```

```

false
Document.Forme.ChombreDisplay.disabled =
.
Document.Forme.CAperato(0).disabled = false
Document.Forme.CAperato(1).disabled = false
Document.Forme.ChumNode.disabled = false
Document.Forme.ChumAcid.disabled = false

Document.Forme.Volr.Value = 1
Document.Forme.submit

Else
MegBox "Ingrese dato",32,"Informachy"
Document.Forme.Solid.Focus

End Sub
End If

Sub Regresar()
History.back
End Sub

Sub Desactiva_ACHLinTel()
If Document.Forme.ACHLinTel.checked then
Document.Forme.ALinTel(0).disabled = false
Document.Forme.ALinTel(1).disabled = false
Document.Forme.AlpoNum(0).disabled = false
Document.Forme.AlpoNum(1).disabled = false
Document.Forme.ANombreDisplay.disabled =

false
else
Document.Forme.ALinTel(0).disabled = true
Document.Forme.ALinTel(1).disabled = true
Document.Forme.AlpoNum(0).disabled = true
Document.Forme.AlpoNum(1).disabled = true
Document.Forme.ANombreDisplay.disabled = true

end if

End Sub

sub Desactiva_ACHCodAut()
If Document.Forme.ACHCodAut.checked then
Document.Forme.ACSLocal.disabled = false
Document.Forme.ACSNec.disabled = false
Document.Forme.ACSInt.disabled = false

else
Document.Forme.ACSLocal.disabled = true
Document.Forme.ACSNec.disabled = true
Document.Forme.ACSInt.disabled = true

end if

End Sub

Sub Desactiva_CCHCodAut()
If Document.Forme.CCHCodAut.checked then
Document.Forme.CCSLocal.disabled = false
Document.Forme.CCSNec.disabled = false
Document.Forme.CCSInt.disabled = false

else
Document.Forme.CCSLocal.disabled = true
Document.Forme.CCSNec.disabled = true
Document.Forme.CCSInt.disabled = true

end if

End Sub

sub Desactiva_CCHNonDisp()
If Document.Forme.CCHNonDisp.checked then
Document.Forme.ChombreDisplay.disabled =

false
else
Document.Forme.ChombreDisplay.disabled = true

end if

End Sub

Sub Desactiva_CCHRepApe()
If Document.Forme.CCHRepApe.checked then
Document.Forme.CAperato(0).disabled = false
Document.Forme.CAperato(1).disabled = false

else
Document.Forme.CAperato(0).disabled = true
Document.Forme.CAperato(1).disabled = true

end if

End Sub

sub Desactiva_CCHMovExt()
If Document.Forme.CCHMovExt.checked then
Document.Forme.ChumNode.disabled = false

else
Document.Forme.ChumNode.disabled = true

end if

End Sub

sub Desactiva_CCHDirExt()
If Document.Forme.CCHDirExt.checked then
Document.Forme.ChumAcid.disabled = false

else
Document.Forme.ChumAcid.disabled = true

end if

End Sub

-->
</Script>

<%End If%>

<%If Valr = 3 then%>

<%
usuario = "sa"
password = "sa"
set cnn2 = server.createObject("ADODB.Connection")
stron = "DSN=Tarificador;UID=" & usuario & ";PWD=" & password & ";DATABASE=CORPORATIVA"
cnn2.open stron
%>

sqlquery = "declare @call varchar(255) exec spCreaCallPrimario
14,SpSelComulacionSocio" & chr(34) & SocioID & chr(34) & ", @call out select @call 'r' "
set rsap = cnn2.execute(sqlquery)
sqlquery = rsap("r")
set rsSol = cnn2.execute(sqlquery)

%>
<%
If not(rsCon.EOF) or not(rsCon.BOF) then
ComuladorID = rsCon("ComuladorID")
End If
cnn2.close

%>

<%
If AC = "Act" then
TipoSol = 1
else
TipoSol = 2
end if
If Extension = "" then
Extension = "NULL"
End If
If CCosto = "" then
CCosto = "NULL"
else

```

```

CCosto = chr(34) & CCosto & chr(34)

End If

sqlquery = "declare @call varchar(255) exec spCreaCallPrimario 16,SpSelSolcitud" &
TipoSol & " & ComuladorID & " & Extension & " & Chr(34) & SocioID & Chr(34) & " & Chr(34) &
NumNode & Chr(34) & " & CCosto & " & @call out select @call 'r' "
set rsap = cnn.execute(sqlquery)
sqlquery = rsap("r")
response.write sqlquery
set rsSol = cnn.execute(sqlquery)

%>
<% If rsSol("RESULTADO") = "0" Then %>
<Script Language = "VBScript">
<--
megbox "Le solcitud ya existe"
History.back

-->
</Script>
<% Else
Solid = rsSol("Solid")
If Solid = "" then
Molvo = "NULL"
else
Molvo = chr(34) & Molvo & chr(34)

end if
sqlquery = "declare @call varchar(255) exec spCreaCallPrimario
16,SpSelSolcitud_Molvo" & Solid & " & Molvo & " & @call out select @call 'r' "
set rsap = cnn.execute(sqlquery)
sqlquery = rsap("r")
response.write sqlquery
set rsSol = cnn.execute(sqlquery)
If ACHLinTel = "on" then
If ALinTel = "Anr" then
Tipolinea = 1
Else
Tipolinea = 2
end if
If ATipokum = "Dir" then
Tipokumero = 1
Else
Tipokumero = 2
end if
Display = chr(34) & Anombredisplay & chr(34)

else
Tipolinea = "NULL"
Tipokumero = "NULL"
Display = "NULL"

end if

If ACHCodAut = "on" then
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 1
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 2
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 3
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 4
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 5
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 6
end if
If ACSLocal = "on" and ACSNec = "on" and ACSInt = "on" then
CodgoAut = 7
end if

else
CodgoAut = "NULL"
end if

If AFechalrio = "on" then
Fechalrio = chr(34) & AFechalrio & chr(34)
else
Fechalrio = "NULL"
end if

If CCHCodAut = "on" then
CodgoAnterior = CCostoAnt
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 1
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 2
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 3
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 4
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 5
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 6
end if
If CCSLocal = "on" and CCSNec = "on" and CCSInt = "on" then
CodgoAut = 7
end if

else
CodgoAnterior = "NULL"
CodgoAut = "NULL"
end if

If CCHNonDisp = "on" then
DisplayNuevo = chr(34) & Chombredisplay & chr(34)
else
DisplayNuevo = "NULL"
end if

If CCHRepApe = "on" then
If CAperato = "Anr" then
NuevoAperato = 1
Else
NuevoAperato = 2
end if

else
NuevoAperato = "NULL"
end if

If CCHMovExt = "on" then
NuevoNode = chr(34) & ChumNode & chr(34)
else
NuevoNode = "NULL"
end if

If CCHDirExt = "on" then
NumDirectoAnt = ChumAcid
else
NumDirectoAnt = "NULL"
end if

If CCHExtDir = "on" then
ExtDirecto = 1
else
ExtDirecto = "NULL"
end if

```



```

If COBLogD = "on" then
    BloqueoLD = 1
else
    BloqueoLD = "NULL"
end if

If rsSol("RESULTADO") = 1 Then
    If TipoSol = 1 Then
        sqquery = "declare @call
        verchar(255) exec spCreaCalPrimario 16, SpineSolitudCamb & " & SolID & " & TipoSol & " & TipoLinea & " &
        Tipohumero & " & Display & " & CodgoAut & " & Fechalicio & " & @call out select @call 'r'
        set nrap =
        cnn.execute(sqquery)
        sqquery = nrap('r')
        set rsSol =
        cnn.execute(sqquery)

        Else
            sqquery = "declare @call
            verchar(255) exec spCreaCalPrimario 16, SpineSolitudCamb & " & SolID & " & TipoSol & " & TipoLinea & " &
            CodgoAnterior & " & DisplayNumero & " & NumeroOpero & " & NumeroNodo & " & NumDirctoAut & " &
            ExtDircto & " & BloqueoLD & " & @call out select @call 'r'
            set nrap =
            cnn.execute(sqquery)
            sqquery = nrap('r')
            set rsSol =
            cnn.execute(sqquery)

            End If
            If rsSol("Resultado") = 1 then
                sqquery = "declare @call
                verchar(255) exec spCreaCalPrimario 14, SpSeSocio & " & chr(34) & SocioID & chr(34) & " & @call out select
                @call 'r'
                set nrap =
                cnn.execute(sqquery)
                sqquery = nrap('r')
                If not isNull(rsSocio("Mail"))
                    usuario =
                    password =
                    set cnn =
                    server.createobject("ADODB.connection")
                    strcn =
                    "DSN=Tranficador;UID=" & usuario & "PWD=" & password & "DATABASE=CORPORATIVA"
                    cnn2.open strcn
                    sqquery =
                    "des201001.master_xp_sendmail @recipients=" & rsSocio("Mail") & " @message=Se ha creado una
                    solitud de servicio con el toto " & SolID & " @subject=Nueva Solitud de Servicio"
                    response.write sqquery
                    set rsMail =
                    cnn2.execute(sqquery)
                    cnn2.close
                    end if
                    %>
                    <Script Language =
                    <--
                    MsgBox
                    VBScript>
                    "Insertado con Exito"
                    history.back
                    </Script>
                    <Else
                        sqquery = "declare @call
                        verchar(255) exec spCreaCalPrimario 16, SpDeSolitud & " & SolID & " & @call out select @call 'r'
                        set nrap =
                        cnn.execute(sqquery)
                        sqquery = nrap('r')
                        set rsSol =
                        cnn.execute(sqquery)
                        %>
                        <Script Language =
                        <--
                        MsgBox
                        VBScript>
                        "Error al Insertar"
                        history.back
                        </Script>
                    </End If
                    <End If %>
                    <End If %>
                    <End If %>
                    <Norm.close%>
                    <Body>
                    </Body>
                    </html>

                    <%@ Language=VBScript %>
                    <% Response.Buffer = True %>
                    <!--
                    ***** Nombre del Autor: Hector Manuel Garza Garza.
                    ***** Nombre del Programa: Actualiza_Sol.asp
                    ***** Fecha Creacion: 03/04/2000.
                    ***** Descripcion: Graba cambios o elimina el registro.
                    ***** Dependencia: Graba_Sol.asp
                    -->
                    <HTML>
                    <HEAD>
                    <Title>Solitud de Servicios TeleLocales/locos/Title>
                    <META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
                    <HEAD>
                    <BODY BGCOLOR="#e7e7e7" Script Language = "VBScript">
                    <!-- include file = "adobis.inc" -->
                    <!--%Option Explicit -->
                    <!-- include file = "seguridad.asp" -->

                    <%
                    SolID = request.form("SolID")
                    SocioID = request.form("SocioID")
                    Extension = request.form("Extension")
                    NumNodo = request.form("NumNodo")
                    CCosto = request.form("CCosto")
                    Motivo = request.form("Motivo")
                    AC = request.form("AC")
                    ACHIntTel = request.form("ACHIntTel")
                    ACHCodAut = request.form("ACHCodAut")
                    AlnTel = request.form("AlnTel")
                    ATipohum = request.form("ATipohum")
                    AnombrDisplay = request.form("AnombrDisplay")

```

```

ACSLocal = request.form("ACSLocal")
ACSHec = request.form("ACSHec")
ACSInt = request.form("ACSInt")
AFechalio = request.form("AFechalio")
COHCodAut = request.form("COHCodAut")
COHNonDrap = request.form("COHNonDrap")
COHResApas = request.form("COHResApas")
COHModExt = request.form("COHModExt")
COHCodAut = request.form("COHCodAut")
COHDirExt = request.form("COHDirExt")
COHExtDir = request.form("COHExtDir")
COBLogD = request.form("COBLogD")
CCodAut = request.form("CCodAut")
CCSLocal = request.form("CCSLocal")
CCSHec = request.form("CCSHec")
CCSInt = request.form("CCSInt")
ChombrDisplay = request.form("ChombrDisplay")
COperato = request.form("COperato")
ChumNodo = request.form("ChumNodo")
ChumMod = request.form("ChumMod")
Valor = request.form("Valor")

<%
usuario = "sa"
password = "sa"
set cnn = server.createobject("ADODB.connection")
strcn = "DSN=Tranficador;UID=" & usuario & "PWD=" & password & "DATABASE=CORPORATIVA"
cnn.open strcn
%>

<% If Valor = 1 Then %>
<%
sqquery = "declare @call verchar(255) exec spCreaCalPrimario
14, SpSeComunidadesSocio & " & chr(34) & SocioID & chr(34) & " & @call out select @call 'r'
set nrap = cnn.execute(sqquery)
sqquery = nrap('r')
set rsCon = cnn.execute(sqquery)
If not(rsCon.EOF) or not(rsCon.BOF) then
    ComunadorID = rsCon("ComunadorID")
End if

If Extension = "" then
    Extension = "NULL"
end if

If NumNodo = "" then
    NumNodo = "NULL"
else
    NumNodo = chr(34) & NumNodo & chr(34)
end if

If CCosto = "" then
    CCosto = "NULL"
else
    CCosto = chr(34) & CCosto & chr(34)
end if

If Motivo = "" then
    Motivo = "NULL"
else
    Motivo = chr(34) & CCosto & chr(34)
end if

If AC = "Ac" then
    TipoSol = 1
else
    TipoSol = 2
end if

If ACHIntTel = "on" then
    If ALIntTel = "An" then
        TipoLinea = 1
    Else
        TipoLinea = 2
    end if
    If ATipohum = "Dir" then
        Tipohumero = 1
    Else
        Tipohumero = 2
    end if
    Display = chr(34) & AnombrDisplay & chr(34)
else
    TipoLinea = "NULL"
    Tipohumero = "NULL"
    Display = "NULL"
end if

If ACHCodAut = "on" then
    If ACSLocal = "on" and ACSHec <> "on" and ACSInt <> "on" then
        CodgoAut = 1
    end if
    If ACSLocal <> "on" and ACSHec = "on" and ACSInt <> "on" then
        CodgoAut = 2
    end if
    If ACSLocal <> "on" and ACSHec <> "on" and ACSInt = "on" then
        CodgoAut = 3
    end if
    If ACSLocal = "on" and ACSHec = "on" and ACSInt <> "on" then
        CodgoAut = 4
    end if
    If ACSLocal = "on" and ACSHec <> "on" and ACSInt = "on" then
        CodgoAut = 5
    end if
    If ACSLocal <> "on" and ACSHec = "on" and ACSInt = "on" then
        CodgoAut = 6
    end if
    If ACSLocal = "on" and ACSHec = "on" and ACSInt = "on" then
        CodgoAut = 7
    end if
else
    CodgoAut = "NULL"
end if

If AFechalio <> "" then
    Fechalicio = chr(34) & AFechalio & chr(34)
else
    Fechalicio = "NULL"
end if

If COHCodAut = "on" then
    CodgoAnterior = CCodAut
    If CCSLocal = "on" and CCSHec <> "on" and CCSInt <> "on" then
        CodgoAut = 1
    end if
    If CCSLocal <> "on" and CCSHec = "on" and CCSInt <> "on" then
        CodgoAut = 2
    end if
    If CCSLocal <> "on" and CCSHec <> "on" and CCSInt = "on" then
        CodgoAut = 3
    end if
    If CCSLocal = "on" and CCSHec = "on" and CCSInt <> "on" then
        CodgoAut = 4
    end if
    If CCSLocal = "on" and CCSHec <> "on" and CCSInt = "on" then
        CodgoAut = 5
    end if
    If CCSLocal <> "on" and CCSHec = "on" and CCSInt = "on" then
        CodgoAut = 6
    end if
    If CCSLocal = "on" and CCSHec = "on" and CCSInt = "on" then
        CodgoAut = 7
    end if
end if

```

```

and If
else
CodigoAnterior = "NULL"
CodigoAnt = "NULL"
end If
If CCHNonDisp = "on" then
DisplayNuevo = chr(34) & Chombredisplay & chr(34)
else
DisplayNuevo = "NULL"
end If
If CCHRepAps = "on" then
If CAperato = "Ans" then
NuevoAperato = 1
Else
NuevoAperato = 2
end If
else
NuevoAperato = "NULL"
end If
If CCHModExt = "on" then
NuevoModo = chr(34) & CNumModo & Chr(34)
else
NuevoModo = "NULL"
end If
If CCHDirExt = "on" then
NumDireccionAnt = CNumAdd
else
NumDireccionAnt = "NULL"
end If
If CCHExtDir = "on" then
ExtDireccion = 1
else
ExtDireccion = "NULL"
end If
If CCHBloqLD = "on" then
BloqueoLD = 1
else
BloqueoLD = "NULL"
end If
sqlquery = "declare @call varchar(255) exec spCreeCallPrimario 16,SpUpdSolcitud " &
Solid & " " & TipoSol & " " & ComutadorID & " " & Extension & " " & chr(34) & SocioID & chr(34) & " " &
NumModo & " " & CCocto & " ", @call out select @call " "
set rsap = crm.execute(sqlquery)
sqlquery = rsp("a")
response.write sqlquery
set rsSol = crm.execute(sqlquery)
If rsSol("Resultado") = 0 Then %>
<Script Language="VBScript">
<!--
msgbox "No es posible actualizar el registro",32,"Informeciyn"
document.location.href = "Solcitud.asp"
-->
</Script>
%< Else
sqlquery = "declare @call varchar(255) exec spCreeCallPrimario 16,
'SpUpdSolcitud_Motivo " & Solid & " " & Chr(34) & Motivo & Chr(34) & " ", @call out select @call " "
set rsap = crm.execute(sqlquery)
sqlquery = rsp("a")
set rsSol = crm.execute(sqlquery)
If rsSol("RESULTADO") = 1 Then
If TipoSol = 1 Then
sqlquery = "declare @call
varchar(255) exec spCreeCallPrimario 16, 'SpUpdSolcitudAct " & Solid & " " & TipoSol & " " & TipoLime & " " &
TipoNumero & " " & Display & " " & CodigoAut & " " & FechaInicio & " ", @call out select @call " "
set rsap =
crm.execute(sqlquery)
sqlquery = rsp("a")
set rsSol =
Else
sqlquery = "declare @call
varchar(255) exec spCreeCallPrimario 16, 'SpUpdSolcitudComb " & Solid & " " & TipoSol & " " & CodigoAut & " " & CodigoAnterior & " " & DisplayNuevo & " " & NuevoAperato & " " & NuevoModo & " " & NumDireccionAnt &
& ExtDireccion & " " & BloqueoLD & " ", @call out select @call " "
set rsap =
crm.execute(sqlquery)
sqlquery = rsp("a")
set rsSol =
End If
If rsSol("Resultado") = 1 then
sqlquery = "declare @call
varchar(255) exec spCreeCallPrimario 14,SpSeSocio " & chr(34) & SocioID & chr(34) & " ", @call out select
@call " "
set rsap =
crm.execute(sqlquery)
sqlquery = rsp("a")
set rsSocio =
If not isNull(rsSocio("Mail"))
and rsSocio("Mail") <> "" then
usuario
"defi"
password
"defi"
set crm2
= server.createObject("ADODB.connection")
strcon =
"DSN=+Tificador;UID="+ usuario & " & "PWD="+ password & ";DATABASE= CORPORATIVA"
sqlquery
crm2.open strcon
"des201001_master_ap_sendmail @recipients=" & rsSocio("Mail") & " ",@message=Se han registrado los
cambios en la solicitud de servicio con el folio " & Solid & " " & @Subject=Cambios en Solicitud de Servicio"
response.write sqlquery
set rsap
= crm2.Execute(sqlquery)
crm2.close
end If
%>
<Script Language =
"VBScript">
<!--
msgbox
"Actualizado con Exito"
document.location.href = "Solcitud.asp"
-->
</Script>
%< Else %>
<Script Language =
"VBScript">
<!--
msgbox
"Error al Actualizar"
document.location.href = "Solcitud.asp"
-->
</Script>
%< End If %>

```

```

<End If%>
<End If%>
<End If%>

% If Valor = 2 Then %>
%
    sqquery = "declare @call varchar(255) exec sp_CreaCallPrimario 16,SpOpcSolitud " &
    & " & ", @call out select @call " & "
    set rsp = con.execute(sqquery)
    sqquery = resp("r")
    set nUsuario = con.execute(sqquery)

    If nUsuario("Resultado") = 0 Then %>
    <Script Language="VBScript">
    <--
    msgbox "No es posible eliminar el registro",32,"Informaciyn"
    document.location.href = "Solitud.asp"
    -->
    </Script>
    <% Else %>
    <Script Language="VBScript">
    <--
    msgbox "El registro se elimino con éxito",32,"Informaciyn"
    document.location.href = "Solitud.asp"
    -->
    </Script>
    <End If%>

<End If%>

<% con.Close %>

<BODY>
<HTML>

<--
***** Nombre del Autor: Hector Manuel Garza Garza.
*****
***** Nombre del Programa: Seguridad.asp
*****
***** Fecha Creacion: 25/04/2000.
*****
***** Descripción: Incluido para detectar el acceso del usuario
*****
***** Dependencia: Varios
*****

-->
<%
If request.cookies("TarUsuario")("UID") = "" then
    response.redirect "usuario_invalido.asp"
else
    If SegOpcionID <> 0 Then
        usuario = "sa"
        password = "sa"
        set conUsuario = server.createObject("ADODB.connection")
        strcon = "DSN=Tratificador;UID=" & usuario & ";PWD=" & password
        conUsuario.open strcon

        sqquery = "declare @call varchar(255) exec sp_CreaCallPrimario
        16,SpSeguridadOpcionPerit & SegOpcionID & ", " & request.cookies("TarUsuario")("Perit") & ", " & @call out select @call " & "
        set rsp = conUsuario.execute(sqquery)
        sqquery = resp("r")
        set nUsuario = conUsuario.execute(sqquery)

        If nUsuario("Resultado") = 0 then
            response.redirect "usuario_invalido.asp"
        End If
        conUsuario.close

    end if

End If

Sub LipaAgregarTelefono(Telefono)
    usuario = "sa"
    password = "sa"
    set conUsuario = server.createObject("ADODB.connection")
    strcon = "DSN=Tratificador;UID=" & usuario & ";PWD=" & password
    conUsuario.open strcon

    sqquery = "declare @call varchar(255) exec sp_CreaCallPrimario
    16,SpSeguridadOpcionPerit & 33 & ", " & request.cookies("TarUsuario")("Perit") & ", " & @call out select @call " & "
    set rsp = conUsuario.execute(sqquery)
    sqquery = resp("r")
    set nUsuario = conUsuario.execute(sqquery)

    If nUsuario("Resultado") = 1 then
        sqquery = "declare @call varchar(255) exec
        sp_CreaCallPrimario 16,SpSetTel_Permisido " & chr(34) & Telefono & chr(34) & ", " & @call out select @call " & "
        set rsp = conUsuario.execute(sqquery)
        sqquery = resp("r")
        set nTel = conUsuario.execute(sqquery)
        If nTel = conUsuario.execute(sqquery)
            response.write "<a href=" &
            chr(34) & "AgregaTelefono.asp?TEL=" & Telefono & chr(34) & ">Agregar</a>" &
            & " if
        End If
        conUsuario.close

    End sub

%>
<--
***** Nombre del Autor: Hector Manuel Garza Garza.
*****
***** Nombre del Programa: Usuario_invalido.asp
*****
***** Fecha Creacion: 20/04/2000.
*****
***** Descripción: Parcial para el usuario invalido
*****
***** Dependencia: Varios
*****

-->
<HTML>
<HEAD>

<TITLE>ACCESO INVALIDO</TITLE>

</HEAD>

<Body BGColor = "Orange">

<BODY>
<br>
<br>
<br>
<center>
<font Face = "Arial" Size = "8" Color = "BLUE">
<strong>Sistema de Tratificación de la Red de Voz de HEB México</strong>
</font>
</center>
<br>
<br>
<center>
<font Face = "Arial" Size = "8" Color = "RED">
<strong>ADVERTENCIA</strong>
</font>
</center>

```

```
</Font>
</center>
<br>
<br>
<center>
<Font Face = "Arial" Size = "5" Color = "WHITE">
<strong>Usad No tiene privilegios para acceder esta página</strong>
</Font>
<br>
<br>
<br>
<br>
<Font Face = "Arial, Tahoma" Size = "3" Color = "Blue">
```

```
<A HREF="default.asp" TARGET="_top">Haga Click Aqui</A> para entrar al Sistema.
</end>
<br>
<br>
<br>
<br>
<br>
<BODY>
</HTML>
```

Visual Basic

```

VERSION 5.00
Object = "848A5603-2CBE-101B-8296-000000000014":#1,160; "MSCOMCTL.OCX"
Object = "831FDD16-8C5C-11D2-8BFC-0000F0754DA1":#2,060; "MSCOMCTL.OCX"
Begin VB.Form frm_principal
    BorderStyle = 4 'Fixed ToolWindow
    Caption = "Tarificador"
    ClientHeight = 630
    ClientLeft = 645
    ClientTop = 2910
    ClientWidth = 7920
    LinkTopic = "Form1"
    MaxButton = 0 'False
    MinButton = 0 'False
    ScaleHeight = 630
    ScaleWidth = 7920
    ShowInTaskbar = 0 'False
    StartUpPosition = 2 'CenterScreen
    Begin MSComctlLib.StatusBar StatusBar1
        Align = 2 'Align Bottom
        Height = 300
        Left = 0
        TabIndex = 2
        Top = 630
        Width = 7920
        _ExtentX = 13470
        _ExtentY = 529
        _Version = 393216
        BeginProperty Panels (RE3867AB-8586-11D1-816A-00C0F0263626)
            NumPanels = 1
            BeginProperty Panel1 (RE3867AB-8586-11D1-816A-00C0F0263626)
                Object.Width = 12700
                MinWidth = 12700
            EndProperty
        EndProperty
    End
    Begin MSComctlLib.MSComm ComPuerto
        Left = 1560
        Top = 480
        _ExtentX = 1005
        _ExtentY = 1005
        _Version = 327681
        DTREnable = -1 'True
        InputLen = 1
        RTThreshold = 1
        RTSEnable = -1 'True
        BaudRate = 2400
    End
    Begin VB.CommandButton Command1
        Caption = "Procesar"
        Height = 375
        Left = 120
        TabIndex = 1
        Top = 480
        Visible = 0 'False
        Width = 1335
    End
    Begin VB.TextBox TxtLlamada
        Height = 285
        Left = 120
        TabIndex = 0
        Top = 120
        Width = 7695
    End
End
Attribute VB_Name = "frm_principal"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Attribute VB_Name = "frm_principal"
Title: Form Principal de Tarificación
Module: frm_principal.frm
Version: 1.0
Fecha: 15/03/2000
'Autor: Hector Manuel Garza Garza

Private Sub Command1_Click()
    On Error GoTo eiq_error

    Mensaje "Procesando Llamada"
    ProcessLlamada TxtLlamada.Text
    Mensaje ""

Exit Sub
eiq_error:
    Envia_mail "Command1_click", Err.Number, Err.Description
    Resume Next
End Sub

Private Sub ComPuerto_OnComm()
    Function: ComPuerto_OnComm
    Descripción: Recepción de Datos del Computador

    Dim VntCuenta As Variant
    Dim VntBuffer As Variant

    On Error GoTo eiq_error
    Mensaje ComPuerto.Input

    Do While ComPuerto.InBufferCount > 0
        VntCuenta = VntCuenta + 1
        If ComPuerto.InBufferCount >= 113 Then
            VntBuffer = ComPuerto.Input
            TxtLlamada.Text = VntBuffer
            If Len(VntBuffer) >= 113 Then
                ProcessLlamada VntBuffer
            End If
        ElseIf VntCuenta > 30000 Then
            VntBuffer = ComPuerto.Input
            Exit Do
        End If
    Loop
Exit Sub
eiq_error:
    Envia_mail "ComPuerto_OnComm", Err.Number, Err.Description
    Resume Next
End Sub

Private Sub Form_DblClick()
    Command1_Click
End Sub

Private Sub Form_Load()
    Function: Form_Load
    Descripción: Inicia el proceso y abre el puerto serial

    Dim sDummy As String
    On Error GoTo eiq_error

```

```

    Mensaje "Abriendo Puerto"
    ComPuerto.InputLen = 113
    ComPuerto.RThreshold = 5
    ComPuerto.CommPort = sPUERTO
    ComPuerto.Settings = sCONFIGURACION
    ComPuerto.PortOpen = True

    sDummy = ComPuerto.Input
    Mensaje ""

Exit Sub
eiq_error:
    Envia_mail "Form_Load", Err.Number, Err.Description
    Mensaje "ERROR Abriendo Puerto"
    Resume Next
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Function: Form_QueryUnload
    Descripción: Verifica si se quiere salir del sistema y cierra el puerto serial

    Dim VntOpcion As Variant
    On Error GoTo eiq_error

    Mensaje "Saliendo"
    VntOpcion = MsgBox("Seguro que desea salir del Tarificador", vbYesNo + vbExclamation, "Precaución")
    If VntOpcion = vbYes Then
        ComPuerto.PortOpen = False
        Exit
    Else
        Cancel = True
        Exit If
    End If
    Mensaje ""

Exit Sub
eiq_error:
    Envia_mail "Form_QueryUnload", Err.Number, Err.Description
    Resume Next
End Sub

Public Sub Mensaje(VntMensaje As Variant)
    On Error GoTo eiq_error
    StrMessages Panels(1).Text = VntMensaje
Exit Sub
eiq_error:
    Resume Next
End Sub

Attribute VB_Name = "modPrincipal"
Title: Modulo Principal
Module: modPrincipal
Version: 1.0
Fecha: 15/03/2000
'Autor: Hector Manuel Garza Garza

Global gConn As ADODB.Connection

Global sPATH_APLICACION As String
Global sEMAIL As String
Global sPATHINI As String
Global sPROFILE As String
Global sCONMUTADORIO As String
Global sLONGTUD As String
Global sUSUARIO As String
Global sPWD As String
Global sCSDI As String
Global sPUERTO As String
Global sCONFIGURACION As String
Global sCENTRO As String
Global sCENTROID As String

Global bBanderaConexion As Boolean

Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal ApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long
Public Sub Main()
    Function: Main
    Descripción: Inicio del Proceso

    Dim x As Date

    On Error GoTo eiq_error

    AssignValuesIniciates
    Set gConn = CreateObject("ADODB.Connection")
    frm_principal.Show

Exit Sub
eiq_error:
    Envia_mail "Main", Err.Number, Err.Description
    Resume Next
End Sub

Public Sub Descomponer_Registro(Vnt_registro As Variant, _
    VntR_Fecha As Variant, _
    VntR_Commutador As Variant, _
    VntR_Extension As Variant, _
    VntR_Duracion As Variant, _
    VntR_Codigo As Variant, _
    VntR_Telefono As Variant)

    Function: Descomponer_Registro
    Descripción: Desglosa el registro enviado por el computador

    User: Descomponer_Registro llamada, VntFecha1, VntConn, VntExt, VntDur, VntCodigo, VntTel

    Dim Vnt_Auxiliar As Variant
    On Error GoTo eiq_error

    frm_principal.Mensaje "Descomponiendo Registro"

    VntR_Fecha = Mid$(Vnt_registro, 4, 8)

    If sLONGTUD = "N" Then
        '---para 4 digitos en ID computador
        If sCENTRO = "CON_ID" Then
            VntR_Commutador = Mid$(Vnt_registro, 50, 4)
            VntR_Extension = Mid$(Vnt_registro, 83, 4)
        Else
            Vnt_Auxiliar = Mid$(Vnt_registro, 50, 8)
            VntR_Commutador = Int(Len(Trim(Vnt_Auxiliar))) > 4, Mid$(Vnt_Auxiliar, 1, 4), sCENTROID)
            VntR_Extension = Int(Len(Trim(Vnt_Auxiliar))) > 4, Mid$(Vnt_Auxiliar, 5, 4), Vnt_Auxiliar)
        End If
    Else
        '---para 2 digitos en ID computador
        If sCENTRO = "CON_ID" Then
            VntR_Commutador = Mid$(Vnt_registro, 50, 2)

```

```

Vnr_Extension = Mid$(Vnr_registro, 61, 4)
Else
    Vnr_Auxiliar = Mid$(Vnr_registro, 56, 6)
    Vnr_Conmutador = IIf(Len(Trim(Vnr_Auxiliar)) > 4, Mid$(Vnr_Auxiliar, 1, 2), sCENTROID)
    Vnr_Extension = IIf(Len(Trim(Vnr_Auxiliar)) > 4, Mid$(Vnr_Auxiliar, 3, 4), Vnr_Auxiliar)
End If
End If

Vnr_Duracion = Mid$(Vnr_registro, 13, 4)
Vnr_Codigo = Mid$(Vnr_registro, 86, 6)
Vnr_Telefono = Mid$(Vnr_registro, 38, 20)

Im_principal Mensaje ""
Exit Sub
eqq_error:
    Envia_mail "Descompone Registro", Err.Number, Err.Description
    Im_principal Mensaje "ERROR descomponiendo el registro"
    Resume Next

End Sub

Public Sub Envia_mail(Vnr_rutina As Variant, _
    Vnr_num_error As Variant, _
    Vnr_descripcion As Variant)

    Dim MpsSession As Object
    Dim Mpmmessage As Object
    Dim ObjRecipients As Object

    On Error GoTo eqq_error
    Im_principal Mensaje "Enviando Mail"

    Set MpsSession = CreateObject("mspl session")
    MpsSession.Login profileName:=sPROFILE
    Set Mpmmessage = MpsSession.Outbox.Messages.Add

    Mpmmessage.subject = "Error en Tarificador"
    Mpmmessage.Text = Format$(Now, "dd/MM/yyyy hh:mm:ss") & " Error en rutina: " & Vnr_rutina & " (" & _
    Vnr_num_error & ", " & Vnr_descripcion & ")"
    Set ObjRecipients = Mpmmessage.Recipients.Add

    ObjRecipients.Name = sEMAIL
    ObjRecipients.Resolve
    Mpmmessage.Send showDialog:=0
    MpsSession.Logoff
    Im_principal Mensaje ""

Exit Sub
eqq_error:
    Escribe_log Vnr_rutina, Vnr_num_error, Vnr_descripcion
    Im_principal Mensaje "ERROR Enviando Mail"
    Resume Next

End Sub

Public Sub Escribe_log(Vnr_rutina As Variant, _
    Vnr_num_error As Variant, _
    Vnr_descripcion As Variant)

    On Error GoTo eqq_error
    Im_principal Mensaje "Escribiendo en Log"

    Open sPATH_APLICACION & "tarif.log" For Append Access Read Write Shared As #1
    Write #1, Format$(Now, "dd/MM/yyyy hh:mm:ss") & " Error en rutina: " & Vnr_rutina & " (" & Vnr_num_error & _
    ", " & Vnr_descripcion & ")"
    Close #1

    Im_principal Mensaje ""
Exit Sub
eqq_error:
    Resume Next
    Im_principal Mensaje "ERROR Escribiendo en LOG"

End Sub

Public Sub AsigneValoresIniciales()
    Dim showstate As String
    Dim Lerror As Long
    On Error GoTo eqq_error

    sPATHINI = App.Path & "\TARIFICA.INI"

    sPATH_APLICACION = FrBuscaValorIni(sPATHINI, "PATHS", "APLICACION")
    sEMAIL = FrBuscaValorIni(sPATHINI, "MAIL", "DIRECCION")
    sPROFILE = FrBuscaValorIni(sPATHINI, "MAIL", "PROFILE")
    sCONMUTADOR = FrBuscaValorIni(sPATHINI, "COMUTADOR", "ID")
    sLONGITUD = FrBuscaValorIni(sPATHINI, "COMUTADOR", "LONGITUD")
    sUSUARIO = FrBuscaValorIni(sPATHINI, "CONECTIVIDAD", "USUARIO")
    sPWD = FrBuscaValorIni(sPATHINI, "CONECTIVIDAD", "PASSWORD")
    sPWD = IIf(sPWD <> "NULL", sPWD, "")
    sDSN = FrBuscaValorIni(sPATHINI, "CONECTIVIDAD", "DSN")
    sPUERTO = FrBuscaValorIni(sPATHINI, "SERIAL", "PUERTO")
    sCONFIGURACION = FrBuscaValorIni(sPATHINI, "SERIAL", "CONFIGURACION")
    sCENTRO = FrBuscaValorIni(sPATHINI, "COMUTADOR", "CENTROAPOYOT")
    sCENTROID = FrBuscaValorIni(sPATHINI, "COMUTADOR", "CENTROAPOYOT")

Exit Sub
eqq_error:
    Envia_mail "AsigneValoresIniciales", Err.Number, Err.Description
    Resume Next

End Sub

Function FrBuscaValorIni(sArchivoINI As String, sSeccion As String, sEntrada As String) As String
    Dim showstate As String
    Dim sCadena As String * 80
    Dim iTemafno As Integer
    Dim sResultado As Integer
    Dim sResultado As String
    On Error GoTo eqq_error
    sCadena = ""
    iTemafno = 80
    If Dir(sArchivoINI, 0) <> "" Then
        Resultado = GetPrivateProfileString(sSeccion, sEntrada, showstate, sCadena, iTemafno, sArchivoINI)
        If (Not (IsEmpty(sCadena))) And (sCadena <> "") Then
            sResultado = Trim$(Mid$(sCadena, 1, Resultado))
        End If
        If sResultado = showstate Then
            MsgBox "No se encontró la sección [" & sSeccion & "] y entrada " & sEntrada & " en el archivo " & _
            sArchivoINI & ", verifique."
            Shell "notepad.exe " & sArchivoINI, vbNormalFocus
        End If
    End If
    Else
        MsgBox "No se encontró el archivo " & sArchivoINI & ", verifique."
    End If
    End If
    Resultado = FrBuscaValorIni(sResultado)
Exit Function
eqq_error:
    Envia_mail "FrBuscaValorIni", Err.Number, Err.Description
    FrBuscaValorIni = ""
    Resume Next
End Function

Public Function FrConectaBO() As Boolean
    On Error GoTo eqq_error
    Im_principal Mensaje "Conectando a BO"
    FrConectaBO = True

```

```

If gConn.State = 0 Then
    gConn.Open "UID=" & sUSUARIO & ";pwd=" & sPWD & ";dsn=" & sDSN
    If gConn.State = 1 Then
        FrConectaBO = True
        bBanderaConexion = False
    Else
        FrConectaBO = False
    End If
End If

Im_principal Mensaje ""
Exit Function
eqq_error:
    FrConectaBO = False
    If bBanderaConexion = False Then
        Envia_mail "Se perdió conexión con la BO", Err.Number, Err.Description
        bBanderaConexion = True
    End If
    Im_principal Mensaje "ERROR Conectando BO"
    Resume Next
End Function

Public Function FrDesconectaBO() As Boolean
    On Error GoTo eqq_error
    Im_principal Mensaje "Desconectando BO"

    If IsObject(gConn) Then
        If gConn.State <> 0 Then
            gConn.Close
            FrDesconectaBO = True
        End If
    End If
    Im_principal Mensaje ""

Exit Function
eqq_error:
    FrDesconectaBO = False
    Envia_mail "FrDesconectaBO", Err.Number, Err.Description
    Im_principal Mensaje "ERROR Desconectando BO"
    Resume Next
End Function

Public Function FrHora(VnrFecha As Variant) As Variant
    Dim VnrHora As Variant
    On Error GoTo eqq_error

    VnrHora = Mid$(VnrFecha, 5, 2)

    If IsNumeric(VnrHora) Then
        FrHora = VnrHora
    Else
        FrHora = 0
    End If
    Exit Function
eqq_error:
    FrHora = 0
    Envia_mail "FrHora", Err.Number, Err.Description
    Im_principal Mensaje "ERROR Calculando Hora"
    Resume Next

End Function

Public Function FrMin(VnrFecha As Variant) As Variant
    Dim VnrMin As Variant
    On Error GoTo eqq_error

    VnrMin = Mid$(VnrFecha, 7, 2)

    If IsNumeric(VnrMin) Then
        FrMin = VnrMin
    Else
        FrMin = 0
    End If
    Exit Function
eqq_error:
    FrMin = 0
    Envia_mail "FrFecha", Err.Number, Err.Description
    Im_principal Mensaje "ERROR Calculando Minutos"
    Resume Next

End Function

Public Function FrFecha(VnrFecha As Variant) As Variant
    Dim VnrMes As Variant
    Dim VnrDia As Variant
    Dim VnrHora As Variant
    Dim VnrMin As Variant
    Dim VnrAnio As Variant
    On Error GoTo eqq_error

    VnrMes = Mid$(VnrFecha, 1, 2)
    VnrDia = Mid$(VnrFecha, 3, 2)
    VnrAnio = Year(Now)

    If IsNumeric(VnrMes) And IsNumeric(VnrDia) And IsNumeric(VnrAnio) Then

        If VnrMes = 12 And VnrDia = 31 Then
            If Month(Now) = 1 And Day(Now) = 1 Then
                VnrAnio = Year(Now) - 1
            End If
        End If

        FrFecha = Str(VnrAnio) & Format(Str(VnrMes), "00") & Format(Str(VnrDia), "00")
    Else
        FrFecha = "19000101"
    End If
    Exit Function
eqq_error:
    FrFecha = "19000101"
    Envia_mail "FrFecha", Err.Number, Err.Description
    Im_principal Mensaje "ERROR Calculando Fecha"
    Resume Next

End Function

Public Function FrDuracionSeg(VnrDuracion As Variant) As Variant
    Dim VnrHoras As Variant
    Dim VnrMin As Variant
    Dim VnrSeg As Variant

    On Error GoTo eqq_error
    If Len(VnrDuracion) >= 4 Then
        VnrHoras = Mid$(VnrDuracion, 1, 1)
        VnrMin = Mid$(VnrDuracion, 2, 2)
        VnrSeg = Mid$(VnrDuracion, 4, 1)
        If IsNumeric(VnrHoras) And IsNumeric(VnrMin) And IsNumeric(VnrSeg) Then
            FrDuracionSeg = VnrHoras * 3600 + VnrMin * 60 + VnrSeg * 6
        End If
    Else
        FrDuracionSeg = 0
    End If

Exit Function
eqq_error:
    FrDuracionSeg = 0
    Envia_mail "FrDuracionSeg", Err.Number, Err.Description
    Im_principal Mensaje "ERROR Calculando duración"
    Resume Next

```

```

End Function

Public Function FnTipoLlamada(VntTelefono As Variant) As Variant
Dim RResultado As Recordset

On Error GoTo eiq_error
Im_principal.Mensaje "Clasificando Tipo Llamada"

If gConn.State = 1 Then
Set RResultado = gConn.Execute("spClasificaTipoLlamada " & VntTelefono & " ")
FnTipoLlamada = RResultado.Fields("TipoLlamada")
Else
FnTipoLlamada = "NULL"
End If
Im_principal.Mensaje ""

Exit Function
eqi_error:
FnTipoLlamada = "NULL"
Envia_mail "FnTipoLlamada", Err.Number, Err.Description
Im_principal.Mensaje "ERROR Clasificando Tipo de Llamada"
Resume Next

End Function

Public Function FnCaveLD(VntTelefono As Variant, VntTipoLlamada As Variant) As Variant
Dim RResultado As Recordset

On Error GoTo eiq_error
Im_principal.Mensaje "Buscando Cave LD"

If gConn.State = 1 Then
Set RResultado = gConn.Execute("spBuscaCaveLD " & VntTelefono & " " & VntTipoLlamada)
FnCaveLD = RResultado.Fields("CaveLD")
Else
FnCaveLD = "NULL"
End If
Im_principal.Mensaje ""

Exit Function
eqi_error:
FnCaveLD = "NULL"
Envia_mail "FnCaveLD", Err.Number, Err.Description
Im_principal.Mensaje "ERROR en Búsqueda de Cave LD"
Resume Next

End Function

Public Sub InsertaLlamada(VntFecha As Variant, VntHora As Variant, VntMin As Variant, VntComunador As Variant, VntExtension As Variant, VntTipoLlamada As Variant, VntCaveLD As Variant, VntDuracion As Variant, VntCodigo As Variant, VntTelefono As Variant)
Dim RResultado As Recordset

On Error GoTo eiq_error
Im_principal.Mensaje "Insertando Llamada"

If gConn.State = 1 Then
Set RResultado = gConn.Execute("spInsLlamada " & VntFecha & " " & VntHora & " " & VntMin & " " & Trim(VntComunador) & " " & Trim(VntExtension) & " " & IIf(Trim(VntTipoLlamada) <> "", Trim(VntTipoLlamada), "NULL") & " " & IIf(Trim(VntCaveLD) <> "", Trim(VntCaveLD), "NULL") & " " & Trim(VntDuracion) & " " & IIf(Trim(VntCodigo) <> "", Trim(VntCodigo), "NULL") & " " & Trim(VntTelefono) & " ")
End If
Im_principal.Mensaje ""

Exit Sub
eqi_error:
Envia_mail "InsertaLlamada", Err.Number, Err.Description
Im_principal.Mensaje "ERROR Insertando Llamada"
Resume Next

End Sub

Public Sub AlmacenaLlamadaBO(lemada As Variant)
Dim VntFecha As Variant
Dim VntHora As Variant
Dim VntMin As Variant
Dim VntCom As Variant
Dim VntExt As Variant
Dim VntDur As Variant
Dim VntCodigo As Variant
Dim VntTel As Variant
Dim VntTipo As Variant
Dim VntCave As Variant
On Error GoTo eiq_error

Im_principal.Mensaje "Almacenando en BO"

Descompones_RegistroLlamada VntFecha, VntCom, VntExt, VntDur, VntCodigo, VntTel
VntFecha = FnFecha(VntFecha)
VntHora = FnHora(VntFecha)
VntMin = FnMin(VntFecha)
VntTipo = FnTipoLlamada(VntTel)
VntCave = FnCaveLD(VntTel, VntTipo)
VntDur = FnDuracionSeg(VntDur)
InsertaLlamada VntFecha, VntHora, VntMin, VntCom, VntExt, VntTipo, VntCave, VntDur, VntCodigo, VntTel

Im_principal.Mensaje ""

Exit Sub
eqi_error:
Envia_mail "AlmacenaLlamadaBO", Err.Number, Err.Description
Im_principal.Mensaje "ERROR Almacenando en BO"
Resume Next

End Sub

Public Sub AlmacenaLlamadaArchivo(VntLlamada As Variant)
On Error GoTo eiq_error
Im_principal.Mensaje "Almacenando Localmente"

Open sPATH_APLICACION & "TarifaM.TXT" For Append Access Read Write Shared As #1
Print #1, VntLlamada
Close #1

Im_principal.Mensaje ""

Exit Sub
eqi_error:
Envia_mail "AlmacenaLlamadaArchivo", Err.Number, Err.Description
Im_principal.Mensaje "ERROR Almacenando localmente"
Resume Next

End Sub

Public Sub ProcesarLlamadasArchivo()
Dim VntLlamada As Variant
On Error GoTo eiq_error

Im_principal.Mensaje "Procesando Llamadas del Almacenamiento Local"

If Dir(sPATH_APLICACION & "TarifaM.TXT") <> "" Then
Open sPATH_APLICACION & "TarifaM.TXT" For Input Access Read Shared As #1

While Not EOF(1)
Line Input #1, VntLlamada
AlmacenaLlamadaBO(VntLlamada)
Wend

Close #1

```

```

K8 sPATH_APLICACION & "TarifaM.TXT"
End If
Im_principal.Mensaje ""

Exit Sub
eqi_error:
Close #1
Envia_mail "ProcesarLlamadasArchivo", Err.Number, Err.Description
Im_principal.Mensaje "ERROR Procesando Llamadas de Almacenamiento Local"
Resume Next

End Sub

Public Sub ProcesarLlamada(VntLlamada As Variant)
Function ProcesarLlamada
Description: Procesa el registro enviado por el comunador
User: ProcesarLlamada VntLlamada

On Error GoTo eiq_error
Im_principal.Mensaje "Procesando Llamada"

If FnConectaBO Then
ProcesarLlamadaArchivo
AlmacenaLlamadaBO VntLlamada
FnDesconectaBO
Else
AlmacenaLlamadaArchivo VntLlamada
End If

DoEvents
Im_principal.Mensaje ""

Exit Sub
eqi_error:
Envia_mail "ProcesarLlamada", Err.Number, Err.Description
Im_principal.Mensaje "Error Procesando Llamada"
Resume Next

End Sub

VERSION 5.00
Begin VB.Form frmActivarExt
Caption = "Activar Extensión"
ClientHeight = 4230
ClientLeft = 3180
ClientTop = 2745
ClientWidth = 5535
LinkTopic = "Form1"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 4230
ScaleWidth = 5535
StartupPosition = 2 'CenterScreen
Begin VB.Frame frmExt
Caption = "SGU's Digitales"
Height = 2175
Left = 3720
TabIndex = 14
Top = 120
Width = 1695
Begin VB.ListBox lstEqExt
Enabled = 0 'False
Height = 1815
Left = 120
TabIndex = 15
Top = 240
Width = 1455
End
End
Begin VB.CommandButton btnCancelar
Caption = "Cancelar"
Height = 360
Left = 4200
TabIndex = 13
Top = 3840
Width = 1230
End
Begin VB.CommandButton btnAceptar
Caption = "Aceptar"
Enabled = 0 'False
Height = 360
Left = 2880
TabIndex = 12
Top = 3840
Width = 1230
End
End
Begin VB.Frame frmFotos
Height = 1455
Left = 120
TabIndex = 2
Top = 2280
Width = 5295
Begin VB.PictureBox PictureBox1
Appearance = 0 'Flat
BackColor = &H00000004
BorderStyle = 0 'None
ForeColor = &H80000000
Height = 375
Left = 960
ScaleHeight = 375
ScaleWidth = 3135
TabIndex = 7
Top = 600
Width = 1335
Begin VB.OptionButton optTelM
Caption = "Tel. Normal"
Enabled = 0 'False
Height = 255
Left = 0
TabIndex = 9
Top = 0
Value = -1 'True
Width = 1575
End
Begin VB.OptionButton optTelS
Caption = "Tel. Ejecutivo"
Enabled = 0 'False
Height = 255
Left = 1800
TabIndex = 8
Top = 0
Width = 1335
End
End
Begin VB.OptionButton optLinea
Caption = "Linea Analógica"
Height = 255
Left = 960
TabIndex = 6
Top = 240
Value = -1 'True
Width = 1575
End
Begin VB.OptionButton optLineaD
Caption = "Linea Digital"
Height = 255
Left = 2760
TabIndex = 5
Top = 240
Width = 1215
End
End

```

```

Begin VS.ComboBox cboCategoria
    Height = 315
    ItemData = "frmActivarExt.frm":0000
    Left = 1560
    List = "frmActivarExt.frm":0010
    Style = 2 'Dropdown List
    TabIndex = 3
    Top = 960
    Width = 2655
End
Begin VS.Label lblCategoria
    AutoSize = -1 'True
    Caption = "Categoría:"
    Height = 195
    Left = 690
    TabIndex = 4
    Top = 1020
    Width = 750
End
Begin VS.Frame frmEQ
    Caption = "EQ's Analógicas"
    Height = 2175
    Left = 1920
    TabIndex = 1
    Top = 120
    Width = 1695
Begin VS.ListBox lstEQ
    Height = 1815
    Left = 120
    TabIndex = 11
    Top = 240
    Width = 1455
End
End
Begin VS.Frame frmExt
    Caption = "Extensiones Libres"
    Height = 2175
    Left = 120
    TabIndex = 0
    Top = 120
    Width = 1695
Begin VS.ListBox lstExtensiones
    Height = 1815
    Left = 120
    TabIndex = 10
    Top = 240
    Width = 1455
End
End
Attribute VB_Name = "frmActivarExt"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Activar Extension
'Modulo: frmActivarExt.frm
'Version: 1.0
'Fecha: 22/04/2000
'Autor: Hector Manuel Garza Garza
Option Explicit

Private Sub btnAceptar_Click()
    Dim iRespuesta As VBMsgBoxResult
    Dim sExtension As String
    Dim sComando As String

    iRespuesta = MsgBox("¿ Desea activar la extensión ?, vbExclamation + vbYesNo,
    "Precaución")
    If iRespuesta = vbNo Then Exit Sub

    MDIPrincipal.MousePointer = 11

    DoEvents
    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
    frmTelnet.txtPantalla.Find("PASSWORD") <> -1
        DoEvents
    End While
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
        sComando = IIf(optAna.Value, "EXTEN:", "RESEI:")
        sComando = sComando & "DIR=" &
        Trim(lstExtensiones.List(lstExtensiones.ListIndex)) & ", "
        sComando = sComando & "EQU=" & IIf(optEQ.Value,
        Trim(lstEQ.List(lstEQ.ListIndex)), Trim(lstEQUD.List(lstEQUD.ListIndex))) &
        ", "
        sComando = sComando & IIf(optLan.Value, "TYPE=EL6", IIf(optTelM.Value,
        "ITYPE=19", "ITYPE=20")) & ", "
        sComando = sComando & "CAT=" &
        cboCategoria.ItemData(cboCategoria.ListIndex) + IIf(optAna.Value, 8, 17) & ", "
        If Not frmTelnet.FmEjecutaComando(sComando) Then
            MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
            frmTelnet.txtPantalla.Find(sComando) = -1
                DoEvents
            End While
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
            frmTelnet.txtPantalla.Find(sComando) <> -1 Then
                MsgBox "La extensión ha sido activada" & Chr(13) & "Realizar el
                enlace físico", vbInformation, "Información"
                frmActualizaInf.cboExtension.AddItem
                Trim(lstExtensiones.List(lstExtensiones.ListIndex))
                frmActualizaInf.cboExtension.Text =
                Trim(lstExtensiones.List(lstExtensiones.ListIndex))
            End If
        End If
    End If
    MDIPrincipal.MousePointer = 8
    Unload Me
End Sub

Private Sub btnCancelar_Click()
    Unload Me
End Sub

Private Sub cboCategoria_Click()
    If lstExtensiones.ListIndex > -1 And (lstEQUD.ListIndex > -1 Or
    lstEQUD.ListIndex > -1) And (optAna.Value Or optDig.Value) And
    cboCategoria.ListIndex > -1 Then
        btnAceptar.Enabled = True
    Else
        btnAceptar.Enabled = False
    End If
End Sub

Private Sub frm_Load()
    Dim iCounter As Integer
    Dim iCounter2 As Integer
    Dim VntArreglo As Variant
    Dim sExtension As String

    MDIPrincipal.MousePointer = 11

    lstEQUD.Clear

    frmTelnet.LimpiaSeleccionesPantalla
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
    frmTelnet.txtPantalla.Find("PASSWORD") = -1 Then
        If Not frmTelnet.FmEjecutaComando("SUWIF:MUMTYP=EX:") Then
            MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
            frmTelnet.txtPantalla.Find("SUWIF:MUMTYP=EX:") = -1
                DoEvents
            End While
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
            frmTelnet.txtPantalla.Find("SUWIF:MUMTYP=EX:") <> -1 Then
                VntArreglo = Split(frmTelnet.txtPantalla.Text)
                For iCounter = 5 To UBound(VntArreglo) - 4
                    lstEQUD.AddItem Trim(Mid(VntArreglo(iCounter), 5, 12))
                Next
            End If
        End If
    End If

    lstExtensiones.Clear

    frmTelnet.LimpiaSeleccionesPantalla
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
    frmTelnet.txtPantalla.Find("PASSWORD") = -1 Then
        If Not frmTelnet.FmEjecutaComando("SUWIF:MUMTYP=EX:") Then
            MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
            Exit Sub
        Else
            DoEvents
            While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
            frmTelnet.txtPantalla.Find("SUWIF:MUMTYP=EX:") = -1
                DoEvents
            End While
            If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
            frmTelnet.txtPantalla.Find("SUWIF:MUMTYP=EX:") <> -1 Then
                VntArreglo = Split(frmTelnet.txtPantalla.Text)
                For iCounter = 5 To UBound(VntArreglo) - 3
                    sExtension = Trim(Mid(VntArreglo(iCounter), 36, 12))
                    If Mid(sExtension, 8, 1) = "-" Then
                        iCounter2 = Int(Mid(sExtension, 1, 4)) To
                        Int(Mid(sExtension, 8, 4))
                        lstExtensiones.AddItem Trim(Str(iCounter2))
                    End If
                Next
            End If
        End If
    End If

    MDIPrincipal.MousePointer = 8
End Sub

Private Sub lstEQ_Click()
    If lstExtensiones.ListIndex > -1 And (lstEQUD.ListIndex > -1 Or
    lstEQUD.ListIndex > -1) And (optAna.Value Or optDig.Value) And
    cboCategoria.ListIndex > -1 Then
        btnAceptar.Enabled = True
    Else
        btnAceptar.Enabled = False
    End If
End Sub

Private Sub lstEQUD_Click()
    If lstExtensiones.ListIndex > -1 And (lstEQUD.ListIndex > -1 Or
    lstEQUD.ListIndex > -1) And (optAna.Value Or optDig.Value) And
    cboCategoria.ListIndex > -1 Then
        btnAceptar.Enabled = True
    Else
        btnAceptar.Enabled = False
    End If
End Sub

Private Sub optAna_Click()
    optTelM.Enabled = False
    optTelF.Enabled = False

    lstEQUD.Enabled = True
    lstEQUD.ListIndex = -1

    If lstExtensiones.ListIndex > -1 And (lstEQUD.ListIndex > -1 Or
    lstEQUD.ListIndex > -1) And (optAna.Value Or optDig.Value) And
    cboCategoria.ListIndex > -1 Then
        btnAceptar.Enabled = True
    Else
        btnAceptar.Enabled = False
    End If
End Sub

Private Sub optDig_Click()
    optTelM.Enabled = True

```

```

optTelE.Enabled = True

lstEQUs.Enabled = False
lstEQUs.Enabled = True
lstEQUs.ListIndex = -1

End Sub

VERSION 5.00
Begin VB.Form frmActualizaInf
Caption = "Actualización de Información en Comutador"
ClientHeight = 4980
ClientLeft = 60
ClientTop = 345
ClientWidth = 6285
LinkTopic = "Focal"
MaxButton = 0 'False
MDIChild = -1 'True
MinButton = 0 'False
ScaleHeight = 4980
ScaleWidth = 6285
Begin VB.CommandButton btnCodigo
Caption = "Codigo"
Enabled = 0 'False
Height = 360
Left = 4920
Top = 32
TabIndex = 32
Width = 1230
End
Begin VB.CommandButton btnNombre
Caption = "Nombre"
Enabled = 0 'False
Height = 360
Left = 4920
Top = 2040
Width = 1230
End
Begin VB.CommandButton btnExtLibres
Caption = "Ext. Libres"
Height = 360
Left = 4920
Top = 30
TabIndex = 30
Width = 1230
End
Begin VB.CommandButton btnEQULibre
Caption = "EQU's Libres"
Height = 360
Left = 4920
Top = 120
TabIndex = 29
Width = 1230
End
Begin VB.CommandButton btnDesactivar
Caption = "Desactivar"
Enabled = 0 'False
Height = 360
Left = 4920
Top = 28
TabIndex = 28
Width = 1560
End
Begin VB.CommandButton btnActivar
Caption = "Activar"
Height = 360
Left = 4920
Top = 27
TabIndex = 27
Width = 1080
End
Begin VB.Frame Frame2
Height = 3495
Left = 120
Top = 4
TabIndex = 4
Width = 4695
Begin VB.TextBox txtIDW
Enabled = 0 'False
Height = 285
Left = 2760
Top = 22
TabIndex = 22
Width = 3120
End
Begin VB.TextBox txtGDM
Enabled = 0 'False
Height = 285
Left = 2760
Top = 21
TabIndex = 21
Width = 2760
End
Begin VB.TextBox txtGM
Enabled = 0 'False
Height = 285
Left = 2760
Top = 20
TabIndex = 20
Width = 2400
End
Begin VB.TextBox txtGCP
Enabled = 0 'False
Height = 285
Left = 2760
Top = 19
TabIndex = 19
Width = 2040
End
Begin VB.TextBox txtBell
Enabled = 0 'False
Height = 285
Left = 2760
Top = 18
TabIndex = 18
Width = 1680
End
Begin VB.TextBox txtCategoria
Enabled = 0 'False
Height = 285
Left = 1320
Top = 17
TabIndex = 17
Width = 1200
End
Begin VB.TextBox txtEQU
Enabled = 0 'False
Height = 285
Left = 1320
Top = 16
TabIndex = 16
Width = 870
End
Begin VB.TextBox txtNombre
Enabled = 0 'False
Height = 285
Left = 1320
Top = 15
TabIndex = 15
Width = 525
End
End
Begin VB.TextBox txtTipoExt
Enabled = 0 'False
Height = 285
Left = 1120
Top = 14
TabIndex = 14
Width = 195
End
Begin VB.Label lblIDW
AutoSize = -1 'True
Caption = "Número individual de diversión:"
Height = 195
Index = 4
Left = 600
Top = 13
TabIndex = 13
Width = 2285
End
Begin VB.Label lblGDM
AutoSize = -1 'True
Caption = "Grupo de no molestar:"
Height = 195
Index = 3
Left = 1125
Top = 12
TabIndex = 12
Width = 2805
End
Begin VB.Label lblGM
AutoSize = -1 'True
Caption = "Grupo de búsqueda:"
Height = 195
Index = 2
Left = 1230
Top = 11
TabIndex = 11
Width = 2445
End
Begin VB.Label lblGCP
AutoSize = -1 'True
Caption = "Grupo de contestación de llamadas:"
Height = 195
Index = 1
Left = 135
Top = 10
TabIndex = 10
Width = 2085
End
Begin VB.Label lblBell
AutoSize = -1 'True
Caption = "Timbre comda:"
Height = 195
Left = 1635
Top = 9
TabIndex = 9
Width = 1725
End
Begin VB.Label lblCategoria
AutoSize = -1 'True
Caption = "Categoría:"
Height = 195
Index = 0
Left = 465
Top = 8
TabIndex = 8
Width = 1245
End
Begin VB.Label lblTipoExt
AutoSize = -1 'True
Caption = "Tipo Extension:"
Height = 195
Left = 120
Top = 7
TabIndex = 7
Width = 1095
End
Begin VB.Label lblEQU
AutoSize = -1 'True
Caption = "EQU:"
Height = 195
Left = 825
Top = 6
TabIndex = 6
Width = 915
End
Begin VB.Label lblNombre
AutoSize = -1 'True
Caption = "Nombre:"
Height = 195
Left = 615
Top = 5
TabIndex = 5
Width = 570
End
Begin VB.CommandButton btnCerrar
Caption = "Cerrar"
Height = 160
Left = 4920
Top = 8
TabIndex = 8
Width = 1230
End
Begin VB.Frame Frame1
Height = 1455
Left = 120
Top = 1
TabIndex = 1
Width = 4695
Begin VB.TextBox txtDireccion
Enabled = 0 'False
Height = 285
Left = 1080
Top = 26
TabIndex = 26
Width = 600
End
Begin VB.TextBox txtComutador
Enabled = 0 'False
Height = 285
Left = 1080
Top = 25
TabIndex = 25
Width = 240
End
Begin VB.ComboBox cboExtension
Height = 315
Left = 1080
Sorted = -1 'True
Style = 2 'Dropdown List
TabIndex = 2
Top = 960
Width = 1215
End
Begin VB.Label lbl2
AutoSize = -1 'True
Caption = "Dirección:"
Height = 195
Left = 300

```



```

TabIndex = 24
Top = 600
Width = 720
End
Begin VB.Label lbl1
AutoSize = -1 'True
Caption = "Comutador:"
Height = 195
Left = 120
TabIndex = 23
Top = 240
Width = 900
End
Begin VB.Label lblExtension
AutoSize = -1 'True
Caption = "Extensión:"
Height = 195
Left = 120
TabIndex = 24
Top = 240
Width = 900
End
End
Attribute VB_Name = "frmActualizaInf"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Actualización de Información en el computador
'Modulo: frmActualizaInf.frm
'Version: 1.0
'Fecha: 22/04/2008
'Autor: Hector Manuel Garza Garza
Option Explicit
Public iComutador As Integer
Public sDireccion As String
Public sNombre1 As String
Public sNombre2 As String
Public bAceptar As Boolean
Public iCategoria As Integer
Public sCodigo As String
Public borrar As Boolean

Private Sub btnActivar_Click()
MDIPrincipal.MousePointer = 11
frmActivarExt.Show vbModal
MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnCerrar_Click()
Unload Me
End Sub

Private Sub btnCodigo_Click()
Dim sExtension As String
frmCodigoAut.Show vbModal
If Not bAceptar Then Exit Sub
MDIPrincipal.MousePointer = 11
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("PASSWORD") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
sExtension = cboExtension.List(cboExtension.ListIndex)
If Not frmTelnet.FnEjecutaComando("AUOC:DIR=" & sExtension & ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ";") = -1
DoEvents
If Not borrar And Right(frmTelnet.txtPantalla.Text, 2) = "<" &
Chr(28) And frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ";") <> -1 Then
If Not frmTelnet.FnEjecutaComando("AUOC:DIR=" & sExtension &
",CILCO=" & sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ",CILCO=" & sExtension &
",AUTH=" & sCodigo & ",CAT=" & iCategoria & ";") = -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ",CILCO=" & sExtension &
",AUTH=" & sCodigo & ",CAT=" & iCategoria & ";") <> -1 Then
cboExtension_Click
End If
Else
cboExtension_Click
End If
End If
End If
MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnDesactivar_Click()
Dim iRespuesta As VbMsgBoxResult
Dim sExtension As String
Dim sComando As String
iRespuesta = MsgBox("¿Desea desactivar la extensión?", vbExclamation +
vbYesNo, "Precaución")
If iRespuesta = vbNo Then Exit Sub
MDIPrincipal.MousePointer = 11
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("PASSWORD") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
sExtension = cboExtension.List(cboExtension.ListIndex)
If Not frmTelnet.FnEjecutaComando("AUOC:DIR=" & sExtension & ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ";") = -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("AUOC:DIR=" & sExtension & ";") <> -1 Then
sComando = If(txtTipoExt.Text = "ATS", "EXTX", "RXDE")
If Not frmTelnet.FnEjecutaComando(sComando & ":DIR=" & sExtension
& ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("YES") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" &
Chr(28) And frmTelnet.txtPantalla.Find("YES") <> -1 Then
MsgBox "Se eliminó la extensión " & sExtension,
vbInformation, "Información"
cboExtension.RemoveItem cboExtension.ListIndex
cboExtension_Click
End If
End If
End If
End If
MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnEQLibres_Click()
MDIPrincipal.MousePointer = 11
frmEQLibres.Show vbModal
MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnExtLibres_Click()
MDIPrincipal.MousePointer = 11
frmExtLibres.Show vbModal
MDIPrincipal.MousePointer = 0
End Sub

Private Sub btnMembre_Click()
Dim sRespuesta As String
Dim sExtension As String
Dim sComando As String
frmMembre.Show vbModal
If Not bAceptar Then Exit Sub
MDIPrincipal.MousePointer = 11
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("PASSWORD") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
sExtension = cboExtension.List(cboExtension.ListIndex)
sNombre1 = If(sNombre = "", "", sNombre1)
sNombre2 = If(sNombre2 = "", "", sNombre2)
If Not frmTelnet.FnEjecutaComando("MIMI:DIR=" & sExtension & ",NAME1=" &
Chr(34) & sNombre1 & Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34) &
",PRES=1:") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("MIMI:DIR=" & sExtension & ",NAME1=" & Chr(34) &
sNombre1 & Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34) & ",PRES=1:") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("MIMI:DIR=" & sExtension & ",NAME1=" & Chr(34) &
sNombre1 & Chr(34) & ",NAME2=" & Chr(34) & sNombre2 & Chr(34) & ",PRES=1:") <> -1
Then
cboExtension_Click
End If
End If
End If
MDIPrincipal.MousePointer = 0
End Sub

Private Sub cboExtension_Click()
Dim iCounter As Integer
Dim sExtension As String
txtTipoExt.Text = ""
txtBell.Text = ""
txtCategoría.Text = ""
txtEQU.Text = ""
txtGCP.Text = ""
txtGMD.Text = ""
txtIDM.Text = ""
txtMembre.Text = ""
btnDesactivar.Enabled = False
btnMembre.Enabled = False
btnCodigo.Enabled = False
If cboExtension.ListIndex = -1 Then Exit Sub
MDIPrincipal.MousePointer = 11
btnDesactivar.Enabled = True
btnMembre.Enabled = True
btnCodigo.Enabled = True
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("PASSWORD") <> -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
sExtension = cboExtension.List(cboExtension.ListIndex)
If Not frmTelnet.FnEjecutaComando("SUUI:DIR=" & sExtension & ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("SUUI:DIR=" & sExtension & ";") <> -1
DoEvents
End

```

```

If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("SUOIP:DIR=" & sExtension & "/") < -1 Then

    frmTelnet.LimpiarSeleccionPantalla
    iCounter = frmTelnet.txtPantalla.Find("TYPE OF DIRECTORY NUMBER")
    :)+ 28

    While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 27
        txtTipoExt.Text = txtTipoExt.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
        iCounter = iCounter + 1
    Wend

    frmTelnet.LimpiarSeleccionPantalla
    iCounter = frmTelnet.txtPantalla.Find("NAME")
    :)+ 28

    While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 27
        txtNombre.Text = txtNombre.Text &
Mid(frmTelnet.txtPantalla.Text, iCounter, 1)
        iCounter = iCounter + 1
    Wend

    frmTelnet.LimpiarSeleccionPantalla
    iCounter = frmTelnet.txtPantalla.Find("EQU")
    :)+ 28

    While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 22
        iCounter = iCounter + 1
        iCounter = iCounter + 2
        If iCounter > 24 Then txtEQU.Text =
Trim(Mid(frmTelnet.txtPantalla.Text, iCounter, 12))

        frmTelnet.LimpiarSeleccionPantalla
        iCounter = frmTelnet.txtPantalla.Find("CAT")
        :)+ 28

        While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 12
            txtbell.Text = txtbell.Text & Mid(frmTelnet.txtPantalla.Text,
iCounter, 1)
            iCounter = iCounter + 1
        Wend

        frmTelnet.LimpiarSeleccionPantalla
        iCounter = frmTelnet.txtPantalla.Find("GROUP CALL PICKUP")
        :)+ 19

        While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 18
            txtGCP.Text = txtGCP.Text & Mid(frmTelnet.txtPantalla.Text,
iCounter, 1)
            iCounter = iCounter + 1
        Wend

        frmTelnet.LimpiarSeleccionPantalla
        iCounter = frmTelnet.txtPantalla.Find("GROUP NOT DISTURB")
        :)+ 22

        While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 21
            txtGND.Text = txtGND.Text & Mid(frmTelnet.txtPantalla.Text,
iCounter, 1)
            iCounter = iCounter + 1
        Wend

        frmTelnet.LimpiarSeleccionPantalla
        iCounter = frmTelnet.txtPantalla.Find("INDIVIDUAL DIVERSION")
        :)+ 29

        While Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(13) And
Mid(frmTelnet.txtPantalla.Text, iCounter, 1) < Chr(10) And iCounter <
Len(frmTelnet.txtPantalla.Text) And iCounter > 28
            txtIDN.Text = txtIDN.Text & Mid(frmTelnet.txtPantalla.Text,
iCounter, 1)
            iCounter = iCounter + 1
        Wend

    End If
End If
MDIPrincipal.MousePointer = 8
End Sub

Private Sub Form_Load()
Dim VntArreglo As Variant
Dim iCounter As Integer
Dim rsConn As Recordset

frmActualizaInf.Height = 5385
frmActualizaInf.Width = 6405

If bConectado Then
MsgBox "Se esta utilizando la conexión al conmutador." & Chr(13) & "Cierre las
ventanas que la utilizan", vbCritical, "Error"
Exit Sub
End If

frmConmutador.Show vbModal

frmTelnet.Show

If iConmutador < 0 And sDireccion < "" Then
MDIPrincipal.MousePointer = 11

EjecutaQuery rsConn, "spSelConmutador" & iConmutador
If Not rsConn.EOF Then
txtConmutador.Text = iConmutador & " - " & rsConn.Fields("Descripcion"),
""
End If
txtDireccion.Text = sDireccion

If Not frmTelnet.FuConecta(sDireccion) Then
MsgBox "No es posible conectarse al conmutador", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) < "<" & Chr(28) And
frmTelnet.txtPantalla.Find("(C) COPYRIGHT ERICSSON") < -1
DoEvents
Wend

```

```

If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("(C) COPYRIGHT ERICSSON") < -1 Then
If Not frmTelnet.FuEjecutaComando("ESDOP:DIR=ALL:") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) < "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("ESDOP:DIR=ALL:") < -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmTelnet.txtPantalla.Find("ESDOP:DIR=ALL:") < -1 Then
VntArreglo = Split(frmTelnet.txtPantalla.Text)
For iCounter = 5 To UBound(VntArreglo) - 4
If Trim(Mid(VntArreglo(iCounter), 1, 5)) < "" Then
choExtension.AddItem Trim(Mid(VntArreglo(iCounter), 1,
5))
End If
If Not frmTelnet.FuEjecutaComando("ESDOP:DIR=ALL:") Then
MsgBox "No es posible ejecutar comandos", vbCritical,
"Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) < "<" &
Chr(28) Or frmTelnet.txtPantalla.Find("ESDOP:DIR=ALL:") < -1
DoEvents
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
And frmTelnet.txtPantalla.Find("ESDOP:DIR=ALL:") < -1 Then
VntArreglo = Split(frmTelnet.txtPantalla.Text)
For iCounter = 5 To UBound(VntArreglo) - 4
If Trim(Mid(VntArreglo(iCounter), 1, 5)) < ""
Then
choExtension.AddItem
Trim(Mid(VntArreglo(iCounter), 1, 5))
End If
End If
End If
End If
End If
MDIPrincipal.MousePointer = 8
Else
Exit Sub
End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
MDIPrincipal.MousePointer = 11

frmTelnet.FuLogoff
Unload frmTelnet

MDIPrincipal.MousePointer = 8
End Sub

VERSION 5.00
Object = ("C:\WINDOWS\WinSxS\x-wwww-39397d00-b060-408d-808c-1b01" & "MSBOLK32.CCX"
Begin VS.Form frmAssignExtension
Caption = "Asignación de Extensión"
ClientHeight = 2580
ClientLeft = 60
ClientTop = 345
ClientWidth = 6465
LinkTopic = "Form1"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 2580
ScaleWidth = 6465
StartupPosition = 2 'CenterScreen
Begin VS.ComboBox cboDepartamentoID
Height = 315
Left = 1920
Style = 2 'Dropdown List
TabIndex = 23
Top = 2160
Visible = 0 'False
Width = 975
End
Begin VS.ComboBox cboNegocioID
Height = 315
Left = 960
Style = 2 'Dropdown List
TabIndex = 23
Top = 2160
Visible = 0 'False
Width = 975
End
Begin VS.ComboBox cboSocioID
Height = 315
Left = 8
Style = 2 'Dropdown List
TabIndex = 21
Top = 2160
Visible = 0 'False
Width = 975
End
Begin VS.ComboBox cboTipoExtension
Height = 315
Left = 2640
Style = 2 'Dropdown List
TabIndex = 15
Top = 898
Visible = 0 'False
Width = 3735
End
Begin VS.TextBox txtExtension
Enabled = 0 'False
Height = 315
Left = 1560
TabIndex = 14
Top = 498
Width = 975
End
Begin VS.ComboBox cboAreaGenerica
Height = 315
Left = 2640
Style = 2 'Dropdown List
TabIndex = 13
Top = 1328
Visible = 0 'False
Width = 3735
End
Begin VS.ComboBox cboDepartamento
Height = 315
Left = 2640
Style = 2 'Dropdown List
TabIndex = 12
Top = 1688
Visible = 0 'False
Width = 3735
End
Begin VS.ComboBox cboNegocio
Height = 315
Left = 2640
Style = 2 'Dropdown List
TabIndex = 11

```

```

Top = 1320
Visible = 0 'False
Width = 3735
End
Begin VB.ComboBox cboSocio
Height = 315
Left = 2640
Style = 2 'Dropdown List
TabIndex = 10
Top = 1320
Visible = 0 'False
Width = 3735
End
Begin VB.TextBox txtComutador
Enabled = 0 'False
Height = 315
Left = 1560
TabIndex = 7
Top = 120
Width = 3300
End
Begin VB.CommandButton btnCancelar
Caption = "Cancelar"
Height = 360
Left = 5160
TabIndex = 6
Top = 2160
Width = 1230
End
Begin VB.CommandButton btnAceptar
Caption = "Aceptar"
Height = 360
Left = 3840
TabIndex = 5
Top = 2160
Width = 1230
End
Begin MS.MaskedTextBox txtTipoExtension
Height = 315
Left = 1560
TabIndex = 16
Top = 890
Width = 975
ExtentX = 1720
ExtentY = 556
Version = 327681
PromptInclude = 0 'False
MaxLength = 4
Format = "####"
Mask = "####"
PromptChar = "-"
End
Begin MS.MaskedTextBox txtSocio
Height = 315
Left = 1560
TabIndex = 17
Top = 1320
Visible = 0 'False
Width = 975
ExtentX = 1720
ExtentY = 556
Version = 327681
PromptInclude = 0 'False
MaxLength = 11
Format = ""
Mask = ""
PromptChar = ""
End
Begin MS.MaskedTextBox txtNegocio
Height = 315
Left = 1560
TabIndex = 18
Top = 1320
Visible = 0 'False
Width = 975
ExtentX = 1720
ExtentY = 556
Version = 327681
PromptInclude = 0 'False
MaxLength = 5
Format = ""
Mask = ""
PromptChar = ""
End
Begin MS.MaskedTextBox txtDepartamento
Height = 315
Left = 1560
TabIndex = 19
Top = 1680
Visible = 0 'False
Width = 975
ExtentX = 1720
ExtentY = 556
Version = 327681
PromptInclude = 0 'False
MaxLength = 10
Format = ""
Mask = ""
PromptChar = ""
End
Begin MS.MaskedTextBox txtAreaGenerica
Height = 315
Left = 1560
TabIndex = 20
Top = 1320
Visible = 0 'False
Width = 975
ExtentX = 1720
ExtentY = 556
Version = 327681
PromptInclude = 0 'False
MaxLength = 4
Format = "####"
Mask = "####"
PromptChar = ""
End
Begin VB.Label lblComutador
AutoSize = -1 'True
Caption = "Comutador:"
Height = 195
Left = 600
TabIndex = 9
Top = 165
Width = 900
End
Begin VB.Label lblExtension
AutoSize = -1 'True
Caption = "Extensión:"
Height = 255
Left = 765
TabIndex = 8
Top = 528
Width = 735
End
Begin VB.Label lblAreaGenerica
AutoSize = -1 'True
Caption = "Área Genérica:"
Height = 195
Left = 435
TabIndex = 4
Top = 1380
Visible = 0 'False
Width = 1065
End
Begin VB.Label lblDepartamento
AutoSize = -1 'True
Caption = "Departamento:"

```

```

Height = 195
Left = 450
TabIndex = 3
Top = 1740
Visible = 0 'False
Width = 1850
End
Begin VB.Label lblNegocio
AutoSize = -1 'True
Caption = "Unidad de Negocio:"
Height = 195
Left = 75
TabIndex = 2
Top = 1380
Visible = 0 'False
Width = 1425
End
Begin VB.Label lblSocio
AutoSize = -1 'True
Caption = "Socio:"
Height = 195
Left = 1850
TabIndex = 1
Top = 1380
Visible = 0 'False
Width = 450
End
Begin VB.Label lblTipoExtension
Caption = "Tipo de Extensión:"
Height = 255
Left = 165
TabIndex = 8
Top = 920
Width = 1335
End
End
Attribute VB_Name = "frmAsignarExtension"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Asignacion de extension
'Modulo: frmAsignarExtension.frm
'Version: 1.0
'Fecha: 22/04/2000
'Autor: Hector Manuel Garza Garza
Option Explicit

Private Sub btnAceptar_Click()
Dim rsExt As Recordset
Select Case cboTipoExtension.ItemData(cboTipoExtension.ListIndex)
Case 1
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmExtension.iComutadorID) & "," &
FnsqlNumero(frmExtension.iExtension) & "," &
FnsqlString(cboSocioID.List(cboSocio.ListIndex)) & "," & Chr(34) & " " & Chr(34)
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case 2
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmExtension.iComutadorID) & "," &
FnsqlNumero(frmExtension.iExtension) & "," &
FnsqlString(cboNegocioID.List(cboNegocio.ListIndex)) & "," &
FnsqlString(cboDepartamentoID.List(cboDepartamento.ListIndex))
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case 3
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmExtension.iComutadorID) & "," &
FnsqlNumero(frmExtension.iExtension) & "," &
FnsqlString(cboAreaGenerica.ItemData(cboAreaGenerica.ListIndex)) & "," & Chr(34) &
" " & Chr(34)
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case 4
EjecutaQuery rsExt, "spEliminaAsignacionExt " &
FnsqlNumero(frmExtension.iComutadorID) & "," &
FnsqlNumero(frmExtension.iExtension)
If rsExt("resultado") = 0 Then
MsgBox "No es posible desasignar la extension", vbCritical,
"Error"
End If
End Select
Unload Me
frmExtension.cboComutador_Click
End Sub

Private Sub btnCancelar_Click()
Unload Me
End Sub

Private Sub cboAreaGenerica_Click()
If cboAreaGenerica.ListIndex > -1 Then
txtAreaGenerica.Text = cboAreaGenerica.ItemData(cboAreaGenerica.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 3 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
Else
txtAreaGenerica.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboDepartamento_Click()
If cboDepartamento.ListIndex > -1 Then
txtDepartamento.Text = cboDepartamentoID.List(cboDepartamento.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 2 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
Else
txtDepartamento.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboSocio_Click()
If cboSocio.ListIndex > -1 Then
txtSocio.Text = cboSocioID.List(cboSocio.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 1 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
Else
txtSocio.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboTipoExtension_Click()
Dim rsExt As Recordset
Dim iCounter As Integer

```

```

MDPrincipal.MousePointer = 11
Me.MousePointer = 11
If cboTipoExtension.List(cboTipoExtension.ListIndex) = "" Borrer Asignacion
** Then
    btnAceptar.Enabled = True
Else
    btnAceptar.Enabled = False
End If

lblSocio.Visible = False
txtSocio.Visible = False
choSocio.Visible = False
lblUsegocio.Visible = False
txtUsegocio.Visible = False
choUsegocio.Visible = False
lblDepartamento.Visible = False
txtDepartamento.Visible = False
choDepartamento.Visible = False
lblAreaGenerica.Visible = False
txtAreaGenerica.Visible = False
choAreaGenerica.Visible = False
choDepartamento.Enabled = False
txtDepartamento.Enabled = False

If cboTipoExtension.ListIndex > -1 Then
    txtTipoExtension.Text =
cboTipoExtension.ItemData(cboTipoExtension.ListIndex)
Select Case cboTipoExtension.ItemData(cboTipoExtension.ListIndex)
Case 1
    choSocio.Clear
    txtSocio = ""
    EjecuteQuery2 rsLista, "spAllSocio"
    If Not rsLista.EOF Then
        For iCounter = 0 To FuCuentaRegistros(rsLista) - 1
            choSocio.AddItem rsLista("Apellido") & " " &
rsLista("Nombre"), iCounter
            choSocioID.AddItem rsLista("SocioID"), iCounter
            rsLista.MoveNext
        Next iCounter
    End If
    lblSocio.Visible = True
    txtSocio.Visible = True
    choSocio.Visible = True
Case 2
    choUsegocio.Clear
    txtUsegocio = ""
    choDepartamento.Clear
    txtDepartamento = ""
    EjecuteQuery2 rsLista, "spAllUsegocio"
    If Not rsLista.EOF Then
        For iCounter = 0 To FuCuentaRegistros(rsLista) - 1
            choUsegocio.AddItem rsLista("Descripcion"), iCounter
            choUsegocioID.AddItem rsLista("UsegocioID"), iCounter
            rsLista.MoveNext
        Next iCounter
    End If
    lblUsegocio.Visible = True
    txtUsegocio.Visible = True
    choUsegocio.Visible = True
    lblDepartamento.Visible = True
    txtDepartamento.Visible = True
    choDepartamento.Visible = True
Case 3
    choAreaGenerica.Clear
    txtAreaGenerica = ""
    LlenaCombo choAreaGenerica, "spAllAreaGenerica"
    lblAreaGenerica.Visible = True
    txtAreaGenerica.Visible = True
    choAreaGenerica.Visible = True
Case Else
    txtTipoExtension.Text = ""
End Select
Else
    txtTipoExtension.Text = ""
End If
MDPrincipal.MousePointer = 0
Me.MousePointer = 8

End Sub
Private Sub cboUsegocio_Click()
Dim rsLista As Recordset
Dim iCounter As Integer
If cboUsegocio.ListIndex > -1 Then
    txtUsegocio.Text = cboUsegocioID.List(cboUsegocio.ListIndex)
    choDepartamentoID.Clear
    choDepartamento.Enabled = True
    txtDepartamento.Enabled = True
    txtDepartamento.Text = ""
    EjecuteQuery2 rsLista, "spRepDepartamentoUsegocio" &
FuqString(txtUsegocio.Text)
    If Not rsLista.EOF Then
        For iCounter = 0 To FuCuentaRegistros(rsLista) - 1
            choDepartamento.AddItem rsLista("Descripcion"), iCounter
            choDepartamentoID.AddItem rsLista("DepartamentoID"), iCounter
            rsLista.MoveNext
        Next iCounter
    End If
    btnAceptar.Enabled = False
Else
    choDepartamentoID.Clear
    choDepartamentoID.Enabled = False
    choDepartamento.Enabled = False
    txtUsegocio.Text = ""
    btnAceptar.Enabled = False
End If
End Sub

Private Sub Form_Load()
Dim lista As Listitem
Dim indice As Long
Dim rsLista As Recordset
Dim iCounter As Integer

txtComandador.Text = frmExtension.txtComandador.Text & " -- " &
frmExtension.cboComandador.Text
indice = frmExtension.lvExtensiones.SelectedItem.Index
Set lista = frmExtension.lvExtensiones.ListItems(indice)
txtExtension.Text = frmExtension.lvExtensiones.ListItems(indice).Text
cboTipoExtension.Clear
cboTipoExtension.AddItem "" Borrar Asignacion ""
LlenaCombo cboTipoExtension, "spAllTipoExtension"

EjecuteQuery2 rsLista, "spRepPropietarioExtension" &
FuqNumero(frmExtension.iComandadorID) & ", " &
FuqNumero(frmExtension.iExtension)
If Not rsLista.EOF Then
    If Not IsNull(rsLista("TipoExtensionID")) Then
        Select Case rsLista("TipoExtensionID")
        Case 1
            txtTipoExtension.Text = "1"
            txtTipoExtension.LostFocus
            txtSocio.Text = rsLista("SocioID")
            txtSocio.Visible = True
            choSocio.Visible = True
            txtSocio.LostFocus
        Case 2

```

```

ItemData = "frmCodigoAut.frm":0000
Left = 840
List = "frmCodigoAut.frm":0010
Style = 2 'Dropdown List
TabIndex = 2
Top = 128
Width = 3015
End
Begin VB.CommandButton btnCancelar
Caption = "Cancelar"
Height = 360
Left = 1640
TabIndex = 1
Top = 960
Width = 1230
End
Begin VB.CommandButton btnAceptar
Caption = "Aceptar"
Enabled = 0 'False
Height = 360
Left = 1220
TabIndex = 0
Top = 960
Width = 1230
End
Begin VB.Label lblCategoria
AutoSize = -1 'True
Caption = "Categoría:"
Height = 195
Left = 0
TabIndex = 4
Top = 180
Width = 750
End
Begin VB.Label lblNombre
AutoSize = -1 'True
Caption = "Código:"
Height = 195
Left = 0
TabIndex = 3
Top = 540
Width = 540
End
Attribute VB_Name = "frmCodigoAut"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Título: Activar códigos de autorización
'Modulo: frmCodigoAut.frm
'Version: 1.0
'Fecha: 22/04/2000
'Autor: Hector Manuel Garza Garza
Option Explicit
Private Sub btnAceptar_Click()
If cboCategoria.ListIndex > -1 And txtCodAut.Text <> "" Or chkBorrar.Value = 1 Then
frmActualizaInf.bAceptar = True
frmActualizaInf.iCategoria = cboCategoria.ItemData(cboCategoria.ListIndex)
frmActualizaInf.sCodigo = txtCodAut.Text
frmActualizaInf.bBorrar = If(chkBorrar.Value = 1, True, False)
Unload Me
Else
MsgBox "Ingrese datos", vbCritical, "Error"
End If
End Sub
Private Sub btnCancelar_Click()
frmActualizaInf.bAceptar = False
frmActualizaInf.iCategoria = -1
frmActualizaInf.sCodigo = ""
frmActualizaInf.bBorrar = False
Unload Me
End Sub
Private Sub cboCategoria_Click()
If cboCategoria.ListIndex > -1 And Len(txtCodAut.Text) = 4 Or chkBorrar.Value = 1 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
End Sub
Private Sub Form_Load()
Dim sExtension As String
Dim iCounter As Integer
Dim j As Integer
MDIPrincipal.MousePointer = 11
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("PASSWORD") <> -1
DoEvents
Wend
If Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Then
sExtension =
frmActualizaInf.cboExtension.List(frmActualizaInf.cboExtension.ListIndex)
If Not frmTelnet.FEjecutaComando("AUCOF:DIR=" & sExtension & ";") Then
MsgBox "No es posible ejecutar comandos", vbCritical, "Error"
Exit Sub
Else
DoEvents
While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
frmTelnet.txtPantalla.Find("AUCOF:DIR=" & sExtension & ";") <> -1
DoEvents
Wend
frmTelnet.LimpiaSelecciónPantalla
If Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) And
frmTelnet.txtPantalla.Find("AUCOF:DIR=" & sExtension & ";") <> -1 Then
frmTelnet.LimpiaSelecciónPantalla
iCounter = frmTelnet.txtPantalla.Find("CUST")
txtCodAut.Text = Trim Mid(frmTelnet.txtPantalla.Text, iCounter
+ 21, 4))
iCounter = Int(Trim Mid(frmTelnet.txtPantalla.Text, iCounter +
43, 2)))
iCounter = iCounter + 19, iCounter - 17, iCounter)
For j = 0 To cboCategoria.ListCount - 1
If cboCategoria.ItemData(j) = iCounter Then
cboCategoria.ListIndex = j
End If
Next
Else
cboCategoria.ListIndex = -1
txtCodAut.Text = ""
End If
End If
End Sub
MDIPrincipal.MousePointer = 0
End Sub
Private Sub txtCodAut_Change()

```

```

If cboCategoria.ListIndex > -1 And Len(txtCodAut.Text) = 4 Or chkBorrar.Value = 1 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
End Sub
VERSION 5.00
Object = "(C:\WINDOWS\WinSxS\x-wwww-332-0000-0000-00000000\11.104\;www.332.ocx"
Begin VB.Form frmCommutador
Caption = "Conexión a Comutador"
ClientHeight = 1590
ClientLeft = 60
ClientTop = 345
ClientWidth = 3960
LinkTopic = "Form1"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 1590
ScaleWidth = 3960
StartupPosition = 2 'CenterScreen
Begin VB.CommandButton btnCancelar
Caption = "Cancelar"
Height = 360
Left = 2640
TabIndex = 6
Top = 1290
Width = 1230
End
Begin VB.CommandButton btnAceptar
Caption = "Aceptar"
Height = 360
Left = 1320
TabIndex = 5
Top = 1290
Width = 1230
End
Begin VB.Frame Frame1
Height = 1095
Left = 120
TabIndex = 0
Top = 0
Width = 3735
Begin MSMask.MaskEditBox txtDireccion
Height = 315
Left = 1080
TabIndex = 4
Top = 460
Width = 2535
ExtentX = 4471
ExtentY = 556
Version = 32768
PromptInclude = 0 'False
MaxLength = 15
PromptChar = ""
End
Begin VB.ComboBox cboCommutador
Height = 315
Left = 1080
Style = 2 'Dropdown List
TabIndex = 2
Top = 240
Width = 2535
End
Begin VB.Label Label1
AutoSize = -1 'True
Caption = "Comutador:"
Height = 195
Left = 120
TabIndex = 2
Top = 300
Width = 900
End
Begin VB.Label Label2
AutoSize = -1 'True
Caption = "Dirección:"
Height = 195
Left = 300
TabIndex = 1
Top = 660
Width = 720
End
Attribute VB_Name = "frmCommutador"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Título: Seleccionar el conmutador a conectarse
'Modulo: frmCommutador.frm
'Version: 1.0
'Fecha: 22/04/2000
'Autor: Hector Manuel Garza Garza
Option Explicit
Private Sub btnAceptar_Click()
If cboCommutador.ListIndex > -1 And txtDireccion.Text <> "" Then
frmActualizaInf.iCommutador =
cboCommutador.ItemData(cboCommutador.ListIndex)
frmActualizaInf.sDireccion = txtDireccion.Text
Unload Me
Else
MsgBox "Ingrese Datos", vbExclamation, "Error de Captura"
cboCommutador.SetFocus
End If
End Sub
Private Sub btnCancelar_Click()
frmActualizaInf.iCommutador = 0
frmActualizaInf.sDireccion = ""
Unload Me
End Sub
Private Sub cboCommutador_Click()
Dim rCom As Recordset
If cboCommutador.ListIndex > -1 Then
EjecutaQuery rCom, "spSelComutador" &
cboCommutador.ItemData(cboCommutador.ListIndex)
If Not rCom.EOF Then
txtDireccion.Text = rCom("DireccionIP", "")
Else
txtDireccion.Text = ""
End If
Else
txtDireccion.Text = ""
End If
End Sub
Private Sub Form_Load()
frmCommutador.Width = 4080
frmCommutador.Height = 1995
LlenaCombo cboCommutador, "spAllComutador"
End Sub
VERSION 5.00
Object = "(C:\WINDOWS\WinSxS\x-wwww-332-0000-0000-00000000\11.104\;www.332.ocx"
Begin VB.Form frmDetExtension
Caption = "Datos de Extensión"
ClientHeight = 4950

```

```

ClientLeft = 60
ClientTop = 345
ClientWidth = 4695
LinkTopic = "Total"
LockControls = -1 'True
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 4950
ScaleWidth = 4695
StartupPosition = 2 'CenterScreen
Begin VB.TextBox txtCometador
    Enabled = 0 'False
    Height = 285
    Left = 1320
    TabIndex = 8
    Top = 120
    Width = 3300
End
Begin VB.CommandButton btnCancelar
    Caption = "Cancelar"
    Height = 360
    Left = 3360
    TabIndex = 17
    Top = 4560
    Width = 1230
End
Begin VB.CommandButton btnAceptar
    Caption = "Aceptar"
    Height = 360
    Left = 2040
    TabIndex = 15
    Top = 4560
    Width = 1230
End
Begin MSMask.MaskedTextBox txtSignal
    Height = 285
    Left = 3720
    TabIndex = 3
    Top = 840
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 4
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtPuerto
    Height = 285
    Left = 1320
    TabIndex = 4
    Top = 1200
    Width = 1455
    _ExtentX = 2566
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 12
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtRack
    Height = 285
    Left = 1320
    TabIndex = 5
    Top = 1560
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 2
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtRegleta
    Height = 285
    Left = 3720
    TabIndex = 6
    Top = 1560
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 4
    Format = "####"
    Mask = "####"
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtFila
    Height = 285
    Left = 1320
    TabIndex = 7
    Top = 1920
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 4
    Format = "####"
    Mask = "####"
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtPar
    Height = 285
    Left = 3720
    TabIndex = 8
    Top = 1920
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 4
    Format = "####"
    Mask = "####"
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtRaiser
    Height = 285
    Left = 1320
    TabIndex = 9
    Top = 2280
    Width = 1455
    _ExtentX = 2566
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 8
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtMonDisp
    Height = 285
    Left = 1320
    TabIndex = 14
    Top = 4080
    Width = 3300
    _ExtentX = 5821
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 20

```

```

PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtTipoTelefono
    Height = 285
    Left = 1320
    TabIndex = 13
    Top = 3720
    Width = 1455
    _ExtentX = 2566
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 8
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtFisico
    Height = 285
    Left = 1320
    TabIndex = 12
    Top = 3360
    Width = 1455
    _ExtentX = 2566
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 7
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtRaiser
    Height = 285
    Left = 1320
    TabIndex = 11
    Top = 3000
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 2
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtRaiser
    Height = 285
    Left = 1320
    TabIndex = 10
    Top = 2640
    Width = 1455
    _ExtentX = 2566
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 8
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtExtension
    Height = 285
    Left = 1320
    TabIndex = 1
    Top = 465
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    BackColor = -2147483634
    PromptInclude = 0 'False
    MaxLength = 4
    Format = "####"
    Mask = "####"
    PromptChar = "-"
End
Begin MSMask.MaskedTextBox txtCTUA
    Height = 285
    Left = 1320
    TabIndex = 2
    Top = 840
    Width = 855
    _ExtentX = 1508
    _ExtentY = 503
    _Version = 327681
    PromptInclude = 0 'False
    MaxLength = 4
    Format = "####"
    Mask = "####"
    PromptChar = "-"
End
Begin VB.Label lblExtension
    AutoSize = -1 'True
    Caption = "Extensión:"
    Height = 255
    Left = 325
    TabIndex = 31
    Top = 480
    Width = 735
End
Begin VB.Label lblCTUA
    AutoSize = -1 'True
    Caption = "CTUA:"
    Height = 255
    Left = 765
    TabIndex = 30
    Top = 855
    Width = 495
End
Begin VB.Label lblCometador
    AutoSize = -1 'True
    Caption = "Cometador:"
    Height = 195
    Left = 360
    TabIndex = 29
    Top = 165
    Width = 900
End
Begin VB.Label Label6
    AutoSize = -1 'True
    Caption = "Nombre Display:"
    Height = 195
    Left = 185
    TabIndex = 28
    Top = 4125
    Width = 1155
End
Begin VB.Label Label5
    AutoSize = -1 'True
    Caption = "Tipo Telefono:"
    Height = 195
    Left = 225
    TabIndex = 27
    Top = 3765
    Width = 1835
End
Begin VB.Label Label4
    AutoSize = -1 'True
    Caption = "Lugar Fisico:"
    Height = 195
    Left = 138
    TabIndex = 26
    Top = 3405
    Width = 930
End
Begin VB.Label Label3
    AutoSize = -1 'True
    Caption = "Color Raiser:"

```

```

Height = 195
Left = 360
TabIndex = 25
Top = 1040
Width = 900
End
Begin VB.Label Label2
AutoSize = -1 'True
Caption = "Raisec:"
Height = 195
Left = 765
TabIndex = 24
Top = 1045
Width = 495
End
Begin VB.Label Label1
AutoSize = -1 'True
Caption = "Raisec Charto:"
Height = 195
Left = 765
TabIndex = 23
Top = 1045
Width = 1005
End
Begin VB.Label lblPar
AutoSize = -1 'True
Caption = "Par:"
Height = 195
Left = 3175
TabIndex = 22
Top = 1965
Width = 285
End
Begin VB.Label lblFila
AutoSize = -1 'True
Caption = "Fila:"
Height = 195
Left = 975
TabIndex = 21
Top = 1965
Width = 285
End
Begin VB.Label lblRegleta
AutoSize = -1 'True
Caption = "Regleta:"
Height = 195
Left = 3060
TabIndex = 20
Top = 1605
Width = 600
End
Begin VB.Label lblRack
AutoSize = -1 'True
Caption = "Rack:"
Height = 195
Left = 825
TabIndex = 19
Top = 1605
Width = 435
End
Begin VB.Label lblPuerto
AutoSize = -1 'True
Caption = "Puerto:"
Height = 195
Left = 750
TabIndex = 18
Top = 1245
Width = 510
End
Begin VB.Label lblSignal
AutoSize = -1 'True
Caption = "Señal:"
Height = 195
Left = 3165
TabIndex = 16
Top = 855
Width = 495
End
Attribute VB_Name = "frmExtension"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Consulta de informacion de extension
'Modulo: frmExtension.frm
'Version: 1.0
'Fecha: 22/04/2008
'Autor: Hector Manuel Garza Garza
Option Explicit

Private Sub btnAceptar_Click()
Dim rsExt As Recordset
Dim strSQLQuery As String
Dim lRespuesta As Integer

If frmExtension.lvsExtensiones.SelectedItem <> Nothing Then
strSQLQuery = "spUpdExtension " &
FnsqlNumero(frmExtension.txtComandador.Text) & ", " &
FnsqlNumero(frmExtension.txtCTUA.Text) & ", " &
FnsqlString(txtSignal.Text) & ", " &
FnsqlString(txtPuerto.Text) & ", " & FnsqlString(txtRack.Text) & ", " &
FnsqlNumero(txtRegleta.Text) & ", " & FnsqlNumero(txtFila.Text) & ", " &
FnsqlNumero(txtPar.Text) & ", " & FnsqlString(txtRaisec4.Text) & ", " &
FnsqlString(txtRaisec.Text) & ", " &
FnsqlString(txtFisico.Text) & ", " & FnsqlString(txtTipoTelefono.Text) & ", " &
FnsqlString(txtModDisp.Text)

EjecutaQuery rsExt, strSQLQuery
If rsExt("resultado") = 0 Then
MsgBox "El registro no pudo ser actualizado", vbCritical, "Error,"
Else
strSQLQuery = "spInsExtension " &
FnsqlNumero(frmExtension.txtComandador.Text) & ", " &
FnsqlNumero(frmExtension.txtCTUA.Text) & ", " &
FnsqlString(txtSignal.Text) & ", " &
FnsqlString(txtPuerto.Text) & ", " & FnsqlString(txtRack.Text) & ", " &
FnsqlNumero(txtRegleta.Text) & ", " & FnsqlNumero(txtFila.Text) & ", " &
FnsqlNumero(txtPar.Text) & ", " & FnsqlString(txtRaisec4.Text) & ", " &
FnsqlString(txtRaisec.Text) & ", " &
FnsqlString(txtFisico.Text) & ", " & FnsqlString(txtTipoTelefono.Text) & ", " &
FnsqlString(txtModDisp.Text)

EjecutaQuery rsExt, strSQLQuery
If rsExt("resultado") = 1 Then
lRespuesta = MsgBox("¿Desea asignar la extensión ?", vbYesno +
vbQuestion, "Frecuenta")
If lRespuesta = vbYes Then
frmExtension.lvsComandadorId = Int(frmExtension.txtComandador.Text)
frmExtension.lvsExtension = Int(txtExtension.Text)
frmAssignExtension.show vbModal
End If
Else
MsgBox "El registro no pudo ser actualizado", vbCritical, "Error,"
End If
End If
End Sub

Private Sub btnCancelar_Click()
Unload Me
End Sub

Private Sub Form_Load()
Dim lLista As Listitem
Dim indice As Long

txtComandador.Text = frmExtension.lstComandador.Text & " -- " &
frmExtension.cboComandador.Text

If frmExtension.lvsExtensiones.SelectedItem <> Nothing Then
indice = frmExtension.lvsExtensiones.SelectedItem.Index
Set lLista = frmExtension.lvsExtensiones.ListItems(indice)
txtExtension.Text = frmExtension.lvsExtensiones.ListItems(indice).Text
txtCTUA.Text = lLista.SubItems(3)
txtSignal.Text = lLista.SubItems(4)
txtPuerto.Text = lLista.SubItems(5)
txtRack.Text = lLista.SubItems(6)
txtRegleta.Text = lLista.SubItems(7)
txtFila.Text = lLista.SubItems(8)
txtPar.Text = lLista.SubItems(9)
txtRaisec4.Text = lLista.SubItems(10)
txtRaisec.Text = lLista.SubItems(11)
txtChaisec.Text = lLista.SubItems(12)
txtFisico.Text = lLista.SubItems(13)
txtTipoTelefono.Text = lLista.SubItems(14)
txtModDisp.Text = lLista.SubItems(15)
btnAceptar.Enabled = True
Else
txtExtension.Text = ""
txtExtension.Enabled = True
txtCTUA.Text = ""
txtSignal.Text = ""
txtPuerto.Text = ""
txtRack.Text = ""
txtRegleta.Text = ""
txtFila.Text = ""
txtPar.Text = ""
txtRaisec4.Text = ""
txtRaisec.Text = ""
txtChaisec.Text = ""
txtFisico.Text = ""
txtTipoTelefono.Text = ""
txtModDisp.Text = ""

btnAceptar.Enabled = False
End If
End Sub

Private Sub txtExtension_Change()
If txtExtension.Text = "" Then
btnAceptar.Enabled = False
Else
btnAceptar.Enabled = True
End If
End Sub

VERSION 5.00
Begin VB.Form frmEQULibre
Caption = "EQU's Libres"
ClientHeight = 3645
ClientLeft = 60
ClientTop = 345
ClientWidth = 3990
LinkTopic = "Form1"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 3645
ScaleWidth = 3990
StartupPosition = 2 'CenterScreen
Begin VB.Frame Frame1
Caption = "Digitales"
Height = 3135
Left = 2040
TabIndex = 3
Top = 0
Width = 1815
Begin VB.ListBox lstEQUD
Height = 2790
Left = 120
TabIndex = 4
Top = 240
Width = 1575
End
End
Begin VB.CommandButton btnCerrar
Caption = "Cerrar"
Height = 360
Left = 2640
TabIndex = 1
Top = 3240
Width = 1230
End
Begin VB.Frame frmEQu
Caption = "Analógicos"
Height = 3135
Left = 120
TabIndex = 2
Top = 0
Width = 1815
Begin VB.ListBox lstEQU
Height = 2790
Left = 120
TabIndex = 2
Top = 240
Width = 1575
End
End
Attribute VB_Name = "frmEQULibre"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Consulta de EQU's libres
'Modulo: frmEQULibre.frm
'Version: 1.0
'Fecha: 22/04/2008
'Autor: Hector Manuel Garza Garza
Option Explicit
Private Sub btnCerrar_Click()
Unload Me
End Sub

Private Sub Form_Load()
Dim lCounter As Integer
Dim VntArreglo As Variant

lstEQU.Clear

frmVntnet.LimpiaSeleccionPantalla
If Right(frmVntnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmVntnet.txtPantalla.Find("PASWORD") = -1 Then

```

```

Unload Me
frmExtension.cboComandador_Click
End Sub

Private Sub btnCancelar_Click()
Unload Me
End Sub

Private Sub Form_Load()
Dim lLista As Listitem
Dim indice As Long

txtComandador.Text = frmExtension.lstComandador.Text & " -- " &
frmExtension.cboComandador.Text

If frmExtension.lvsExtensiones.SelectedItem <> Nothing Then
indice = frmExtension.lvsExtensiones.SelectedItem.Index
Set lLista = frmExtension.lvsExtensiones.ListItems(indice)
txtExtension.Text = frmExtension.lvsExtensiones.ListItems(indice).Text
txtCTUA.Text = lLista.SubItems(3)
txtSignal.Text = lLista.SubItems(4)
txtPuerto.Text = lLista.SubItems(5)
txtRack.Text = lLista.SubItems(6)
txtRegleta.Text = lLista.SubItems(7)
txtFila.Text = lLista.SubItems(8)
txtPar.Text = lLista.SubItems(9)
txtRaisec4.Text = lLista.SubItems(10)
txtRaisec.Text = lLista.SubItems(11)
txtChaisec.Text = lLista.SubItems(12)
txtFisico.Text = lLista.SubItems(13)
txtTipoTelefono.Text = lLista.SubItems(14)
txtModDisp.Text = lLista.SubItems(15)
btnAceptar.Enabled = True
Else
txtExtension.Text = ""
txtExtension.Enabled = True
txtCTUA.Text = ""
txtSignal.Text = ""
txtPuerto.Text = ""
txtRack.Text = ""
txtRegleta.Text = ""
txtFila.Text = ""
txtPar.Text = ""
txtRaisec4.Text = ""
txtRaisec.Text = ""
txtChaisec.Text = ""
txtFisico.Text = ""
txtTipoTelefono.Text = ""
txtModDisp.Text = ""

btnAceptar.Enabled = False
End If
End Sub

Private Sub txtExtension_Change()
If txtExtension.Text = "" Then
btnAceptar.Enabled = False
Else
btnAceptar.Enabled = True
End If
End Sub

VERSION 5.00
Begin VB.Form frmEQULibre
Caption = "EQU's Libres"
ClientHeight = 3645
ClientLeft = 60
ClientTop = 345
ClientWidth = 3990
LinkTopic = "Form1"
MaxButton = 0 'False
MinButton = 0 'False
ScaleHeight = 3645
ScaleWidth = 3990
StartupPosition = 2 'CenterScreen
Begin VB.Frame Frame1
Caption = "Digitales"
Height = 3135
Left = 2040
TabIndex = 3
Top = 0
Width = 1815
Begin VB.ListBox lstEQUD
Height = 2790
Left = 120
TabIndex = 4
Top = 240
Width = 1575
End
End
Begin VB.CommandButton btnCerrar
Caption = "Cerrar"
Height = 360
Left = 2640
TabIndex = 1
Top = 3240
Width = 1230
End
Begin VB.Frame frmEQu
Caption = "Analógicos"
Height = 3135
Left = 120
TabIndex = 2
Top = 0
Width = 1815
Begin VB.ListBox lstEQU
Height = 2790
Left = 120
TabIndex = 2
Top = 240
Width = 1575
End
End
Attribute VB_Name = "frmEQULibre"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Consulta de EQU's libres
'Modulo: frmEQULibre.frm
'Version: 1.0
'Fecha: 22/04/2008
'Autor: Hector Manuel Garza Garza
Option Explicit
Private Sub btnCerrar_Click()
Unload Me
End Sub

Private Sub Form_Load()
Dim lCounter As Integer
Dim VntArreglo As Variant

lstEQU.Clear

frmVntnet.LimpiaSeleccionPantalla
If Right(frmVntnet.txtPantalla.Text, 2) = "<" & Chr(28) And
frmVntnet.txtPantalla.Find("PASWORD") = -1 Then

```

```

If Not frmTelnet.FmEjecutaComando("SYEVF:TYPE=EL6:") Then
    MsgBox "No es posible ejecutar comando", vbCritical, "Error"
    Exit Sub
Else
    DoEvents
    While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
    frmTelnet.txtPantalla.Find("SYEVF:TYPE=EL6:") = -1
        DoEvents
    Wend
    If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
    frmTelnet.txtPantalla.Find("SYEVF:TYPE=EL6:") <> -1 Then
        VntArreglo = Split(frmTelnet.txtPantalla.Text)
        For iCounter = 5 To Ubound(VntArreglo) - 4
            lstEqwD.AddItem Trim(Mid(VntArreglo(iCounter), 5, 12))
        Next
    End If
End If
End If
lstEqwD.Clear

frmTelnet.LimpiaSeleccionPantalla
If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
    frmTelnet.txtPantalla.Find("PASSWORD") = -1 Then
    If Not frmTelnet.FmEjecutaComando("SYEVF:TYPE=EL1:") Then
        MsgBox "No es posible ejecutar comando", vbCritical, "Error"
        Exit Sub
    Else
        DoEvents
        While Right(frmTelnet.txtPantalla.Text, 2) <> "<" & Chr(28) Or
        frmTelnet.txtPantalla.Find("SYEVF:TYPE=EL1:") = -1
            DoEvents
        Wend
        If Right(frmTelnet.txtPantalla.Text, 2) = "<" & Chr(28) And
        frmTelnet.txtPantalla.Find("SYEVF:TYPE=EL1:") <> -1 Then
            VntArreglo = Split(frmTelnet.txtPantalla.Text)
            For iCounter = 5 To Ubound(VntArreglo) - 4
                lstEqwD.AddItem Trim(Mid(VntArreglo(iCounter), 5, 12))
            Next
        End If
    End If
End If
End Sub

```

VERSION 5.00

Object = ("831FD016-0C5C-1102-A9FC-0000F8754DA182.060"; "MSCOMCTL.COCX"
Object = ("C932BA88-4374-101B-A5C6-00AA003668DC161.160"; "MSMAK32.COCX")

Begin VB.Frame frmExtensiones

Caption = "Extensiones"
ClientHeight = 6195
ClientLeft = 60
ClientTop = 345
ClientWidth = 11385
FillColor = &H00FFFFFF
LinkTopic = "Form1"
MaxButton = 0
MDIChild = -1
ScaleHeight = 6195
ScaleWidth = 11385
Begin VB.CommandButton btnPermisos

Caption = "Permisos"
Enabled = 0
Height = 360
Left = 9840
TabIndex = 9
Top = 2280
Width = 1230
End

Begin VB.Frame frmExtensiones

Caption = "Extensiones"
Height = 5655
Left = 120
TabIndex = 3
Top = 480
Width = 11175
Begin VB.CommandButton btnCorrer

Caption = "Correr"
Height = 360
Left = 9840
TabIndex = 8
Top = 2400
Width = 1230
End

Begin VB.CommandButton btnEliminar

Caption = "Eliminar"
Enabled = 0
Height = 360
Left = 9840
TabIndex = 6
Top = 840
Width = 1230
End

Begin VB.CommandButton btnAgregar

Caption = "Agregar"
Enabled = 0
Height = 360
Left = 9840
TabIndex = 5
Top = 360
Width = 1230
End

Begin MSComctlLib.ListView lstExtensiones

Height = 3295
Left = 120
TabIndex = 4
Top = 240
Width = 9615
ExtentX = 16960
ExtentY = 9340
View = 3
LabelEdit = 1
LabelWrap = -1
HideSelection = 0
FullRowSelect = -1
Version = 393217
ForeColor = -2147483640
BackColor = -2147483643
BorderStyle = 1
Appearance = 1
Items = 16
BeginProperty ColumnHeader(1) (8001F052-8508-11D1-B16A-00C0F0283628)

Text = "Extensiones"
Object.Width = 1764
EndProperty

BeginProperty ColumnHeader(2) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 1
Text = "T. Ext."
Object.Width = 1235
EndProperty

BeginProperty ColumnHeader(3) (8001F052-8508-11D1-B16A-00C0F0283628)

SubItemIndex = 2

Text = "Propietario"
Object.Width = 6174
EndProperty

BeginProperty ColumnHeader(4) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 3
Text = "CIVA"
Object.Width = 1235
EndProperty

BeginProperty ColumnHeader(5) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 4
Text = "Sedal"
Object.Width = 1764
EndProperty

BeginProperty ColumnHeader(6) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 5
Text = "Puerto"
Object.Width = 2540
EndProperty

BeginProperty ColumnHeader(7) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 6
Text = "Rack"
Object.Width = 1235
EndProperty

BeginProperty ColumnHeader(8) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 7
Text = "Regleta"
Object.Width = 1764
EndProperty

BeginProperty ColumnHeader(9) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 8
Text = "Fila"
Object.Width = 1235
EndProperty

BeginProperty ColumnHeader(10) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 9
Text = "Rc"
Object.Width = 1235
EndProperty

BeginProperty ColumnHeader(11) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 10
Text = "Raiser Cuarto"
Object.Width = 2540
EndProperty

BeginProperty ColumnHeader(12) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 11
Text = "Raiser"
Object.Width = 2646
EndProperty

BeginProperty ColumnHeader(13) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 12
Text = "Color Raiser"
Object.Width = 2540
EndProperty

BeginProperty ColumnHeader(14) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 13
Text = "Lugar Fisico"
Object.Width = 2540
EndProperty

BeginProperty ColumnHeader(15) (8001F052-8508-11D1-B16A-00C0F0283628)

Alignment = 2
SubItemIndex = 14
Text = "Tipo Tel4fono"
Object.Width = 2540
EndProperty

BeginProperty ColumnHeader(16) (8001F052-8508-11D1-B16A-00C0F0283628)

SubItemIndex = 15
Text = "Nombre Display"
Object.Width = 3528
EndProperty

End

Begin VB.ComboBox cboComunador

Height = 315
Left = 1920
Style = 2
TabIndex = 2
Top = 120
Width = 5895
End

Begin MSMask.MaskedTextBox txtComunador

Height = 315
Left = 1280
TabIndex = 1
Top = 120
Width = 615
ExtentX = 1885
ExtentY = 556
Version = 327681
PromptInclude = 8
AllowPrompt = -1
MaxLength = 4
Format = "9999"
Mask = "9999"
PromptChar = ""
End

Begin VB.Label lblComunador

AutoSize = -1
Caption = "Comunador:"
Height = 195
Left = 240
TabIndex = 8
Top = 180
Width = 900
End

End

Attribute VB Name = "frmExtension"

Attribute VB GlobalNameSpace = False

Attribute VB Creatable = False

Attribute VB PredeclaredId = True

Attribute VB Exposed = False

Titlebar: Administracion de Extensiones

Module: frmExtension.frm

Version: 1.0

Fecha: 22/04/2008

Autor: Hector Manuel Garcia Garcia

Option Explicit

Public iComunadorId As Integer

Public iExtension As Integer

Private Sub Limpiar(iComunadorId As Integer)

Dim rstExt As ADODB.Recordset

Dim lstExt As ListItems

lstExt.Items.Clear

Set lstExt = lstExt.Items.Add(, , gsNuevo)

EjecutaQuery rstExt = "spRepExtensionesComunador " & Str(iComunadorId)

While Not rstExt.EOF

Set lstExt = lstExt.Items.Add(, , rstExt("Extension"))

lstExt.SubItems(1) = rstExt("TipoExtensionID")

lstExt.SubItems(2) = rstExt("Propietario")

[illegible]

[illegible]

[illegible]

Begin VB.Picturebox Picture1

[illegible]

```

Width = 735
End
Begin VB.Label lblAreaGenerica
AutoSize = -1 'True
Caption = "Área Generica:"
Height = 195
Left = 435
TabIndex = 4
Top = 1380
Visible = 0 'False
Width = 1065
End
Begin VB.Label lblDepartamento
AutoSize = -1 'True
Caption = "Departamento:"
Height = 195
Left = 450
TabIndex = 3
Top = 1740
Visible = 0 'False
Width = 1050
End
Begin VB.Label lblUnegocio
AutoSize = -1 'True
Caption = "Unidad de Negocio:"
Height = 195
Left = 75
TabIndex = 2
Top = 1380
Visible = 0 'False
Width = 1425
End
Begin VB.Label lblSocio
AutoSize = -1 'True
Caption = "Socio:"
Height = 195
Left = 1050
TabIndex = 1
Top = 1380
Visible = 0 'False
Width = 450
End
Begin VB.Label lbl1
Caption = "Tipo de Extensión:"
Height = 255
Left = 165
TabIndex = 0
Top = 920
Width = 1335
End
Attribute VB_Name = "frmProcesarAsignacionExt"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'Titulo: Proceso de asignacion de extensión
'Modulo: frmProcesarExt.frm
'Version: 1.0
'Fecha: 22/04/2008
'Autor: Hector Manuel Garcia Garza
Option Explicit

Private Sub btnAceptar_Click()
Dim rsExt As Recordset
Select Case cboTipoExtension.ListIndex
Case 1
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmProceso.iComutador) & ", " & FnsqlNumero(frmProceso.iExtension) &
", 1," & FnsqlString(cboSocioID.List(cboSocio.ListIndex)) & ", " & Chr(34) & " " &
Chr(34)
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case 2
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmProceso.iComutador) & ", " & FnsqlNumero(frmProceso.iExtension) &
", 2," & FnsqlString(cboUnegocioID.List(cboUnegocio.ListIndex)) & ", " &
FnsqlString(cboDepartamentoID.List(cboDepartamento.ListIndex))
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case 3
EjecutaQuery rsExt, "spAsignaExtension " &
FnsqlNumero(frmProceso.iComutador) & ", " & FnsqlNumero(frmProceso.iExtension) &
", 3," & FnsqlNumero(cboAreaGenerica.IData(cboAreaGenerica.ListIndex)) & ", " &
Chr(34) & " " & Chr(34)
If rsExt("resultado") = 0 Then
MsgBox "No es posible asignar la extension", vbCritical, "Error"
End If
Case Else
EjecutaQuery rsExt, "spEliminaAsignacionExt " &
FnsqlNumero(frmProceso.iComutador) & ", " & FnsqlNumero(frmProceso.iExtension)
If rsExt("resultado") = 0 Then
MsgBox "No es posible desasignar la extensión", vbCritical,
"Error"
End If
End Select
Unload Me
'frmExtension.cboComutador_Click
End Sub

Private Sub btnCancelar_Click()
Unload Me
End Sub

Private Sub cboAreaGenerica_Click()
If cboAreaGenerica.ListIndex > -1 Then
txtAreaGenerica.Text = cboAreaGenerica.ItemData(cboAreaGenerica.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 3 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
Else
txtAreaGenerica.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboDepartamento_Click()
If cboDepartamento.ListIndex > -1 Then
txtDepartamento.Text = cboDepartamento.ID.List(cboDepartamento.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 2 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
Else
txtDepartamento.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboSocio_Click()
If cboSocio.ListIndex > -1 Then
txtSocio.Text = cboSocio.ID.List(cboSocio.ListIndex)
If cboTipoExtension.ItemData(cboTipoExtension.ListIndex) = 1 Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If
End Sub

Else
btnAceptar.Enabled = False
End If
Else
txtSocio.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub cboTipoExtension_Click()
Dim rsLista As Recordset
Dim iCounter As Integer
MDIPrincipal.MousePointer = 11
Me.MousePointer = 11
If cboTipoExtension.List(cboTipoExtension.ListIndex) = "" Borrer Asignacion
"" Then
btnAceptar.Enabled = True
Else
btnAceptar.Enabled = False
End If

lblSocio.Visible = False
txtSocio.Visible = False
cboSocio.Visible = False
lblUnegocio.Visible = False
txtUnegocio.Visible = False
cboUnegocio.Visible = False
lblDepartamento.Visible = False
txtDepartamento.Visible = False
cboDepartamento.Visible = False
lblAreaGenerica.Visible = False
txtAreaGenerica.Visible = False
cboAreaGenerica.Visible = False
cboDepartamento.Enabled = False
txtDepartamento.Enabled = False

If cboTipoExtension.ListIndex > -1 Then
txtTipoExtension.Text =
cboTipoExtension.ItemData(cboTipoExtension.ListIndex)
Select Case cboTipoExtension.ItemData(cboTipoExtension.ListIndex)
Case 1
cboSocio.Clear
txtSocio = ""
EjecutaQuery2 rsLista, "spAllSocio"
If Not rsLista.EOF Then
For iCounter = 0 To FncuentaRegistros(rsLista) - 1
cboSocio.AddItem rsLista("Apellido") & " " &
rsLista("Nombres") & " " & rsLista("Webster"), iCounter
cboSocioID.AddItem rsLista("SocioID"), iCounter
rsLista.MoveNext
Next iCounter
End If
lblSocio.Visible = True
txtUnegocio.Visible = True
cboUnegocio.Visible = True
lblDepartamento.Visible = True
txtDepartamento.Visible = True
cboDepartamento.Visible = True
Case 2
cboUnegocio.Clear
txtUnegocio = ""
cboDepartamento.Clear
txtDepartamento = ""
EjecutaQuery2 rsLista, "spAllUnegocio"
If Not rsLista.EOF Then
For iCounter = 0 To FncuentaRegistros(rsLista) - 1
cboUnegocio.AddItem rsLista("Descripcion"), iCounter
cboUnegocioID.AddItem rsLista("UnegocioID"), iCounter
rsLista.MoveNext
Next iCounter
End If
lblUnegocio.Visible = True
txtUnegocio.Visible = True
cboUnegocio.Visible = True
lblDepartamento.Visible = True
txtDepartamento.Visible = True
cboDepartamento.Visible = True
Case 3
cboAreaGenerica.Clear
txtAreaGenerica = ""
ListCombo cboAreaGenerica, "spAllAreaGenerica"
lblAreaGenerica.Visible = True
txtAreaGenerica.Visible = True
cboAreaGenerica.Visible = True
Case Else
txtTipoExtension.Text = ""
End Select
Else
txtTipoExtension.Text = ""
End If
MDIPrincipal.MousePointer = 0
Me.MousePointer = 0
End Sub

Private Sub cboUnegocio_Click()
Dim rsLista As Recordset
Dim iCounter As Integer
If cboUnegocio.ListIndex > -1 Then
txtUnegocio.Text = cboUnegocio.ID.List(cboUnegocio.ListIndex)
cboDepartamento.Clear
cboDepartamentoID.Clear
cboDepartamento.Enabled = True
txtDepartamento.Enabled = True
txtDepartamento.Text = ""
EjecutaQuery2 rsLista, "spDepDepartamentoUnegocio " &
FnsqlString(txtUnegocio.Text)
If Not rsLista.EOF Then
For iCounter = 0 To FncuentaRegistros(rsLista) - 1
cboDepartamento.AddItem rsLista("Descripcion"), iCounter
cboDepartamentoID.AddItem rsLista("DepartamentoID"), iCounter
rsLista.MoveNext
Next iCounter
End If
btnAceptar.Enabled = False
Else
cboDepartamento.Clear
cboDepartamentoID.Clear
cboDepartamento.Enabled = False
txtDepartamento.Enabled = False
txtUnegocio.Text = ""
btnAceptar.Enabled = False
End If
End Sub

Private Sub Form_Load()
Dim iLista As Integer
Dim iIndex As Long
Dim rsLista As Recordset
Dim iCounter As Integer

txtComutador.Text = frmProceso.txtComutador.Text
txtExtension.Text = frmProceso.txtExtension.Text
cboTipoExtension.Clear
cboTipoExtension.AddItem "" Borrer Asignacion ""
ListCombo cboTipoExtension, "spAllTipoExtension"

EjecutaQuery rsLista, "spDepPropietarioExtension " &
FnsqlNumero(frmProceso.iComutador) & ", " & FnsqlNumero(frmProceso.iExtension)
If Not rsLista.EOF Then
If Not IsNull(rsLista("TipoExtensionID")) Then
Select Case rsLista("TipoExtensionID")

```

```

End Sub
Private Sub btnVer_Click()
    Dim strSQL As String
    strSQL = "SELECT * FROM tblVer WHERE 1=1"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenSnapshot)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Ver"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Ver - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If
End Sub

Private Sub btnAtualizar_Click()
    Dim strSQL As String
    strSQL = "UPDATE tblVer SET Nome = '" & Text1.Text & "', Endereco = '" & Text2.Text & "', Telefone = '" & Text3.Text & "'"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenUpdate)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Atualizar"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Atualizar - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If
End Sub

Private Sub btnExcluir_Click()
    Dim strSQL As String
    strSQL = "DELETE FROM tblVer WHERE Nome = '" & Text1.Text & "'"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenDelete)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Excluir"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Excluir - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If
End Sub

Private Sub btnCancelar_Click()
    Dim strSQL As String
    strSQL = "SELECT * FROM tblVer WHERE 1=1"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenSnapshot)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Cancelar"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Cancelar - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If
End Sub

```

```

Case 1
    strSQL = "SELECT * FROM tblVer WHERE 1=1"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenSnapshot)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Ver"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Ver - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If

Case 2
    strSQL = "UPDATE tblVer SET Nome = '" & Text1.Text & "', Endereco = '" & Text2.Text & "', Telefone = '" & Text3.Text & "'"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenUpdate)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Atualizar"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Atualizar - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If

Case 3
    strSQL = "DELETE FROM tblVer WHERE Nome = '" & Text1.Text & "'"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenDelete)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Excluir"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Excluir - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If

Case 4
    strSQL = "SELECT * FROM tblVer WHERE 1=1"
    Dim rs As Recordset
    Set rs = DAO.OpenRecordset(strSQL, dbOpenSnapshot)
    If rs.EOF Then
        MsgBox "Nenhuma informação encontrada.", vbInformation, "Cancelar"
    Else
        Dim i As Integer
        i = 0
        Do While Not rs.EOF
            i = i + 1
            Dim frm As New Form
            frm.Caption = "Cancelar - " & rs.Fields("Nome").Value
            frm.Width = 300
            frm.Height = 100
            frm.Text1.Text = rs.Fields("Nome").Value
            frm.Text2.Text = rs.Fields("Endereco").Value
            frm.Text3.Text = rs.Fields("Telefone").Value
            frm.Show
            rs.MoveNext
        Loop
    End If
End Sub

```

Capiton	=	"Resposición de Aparato Telefonico"
Height	=	255

[illegible]

```

AgregaReporte "Reporte de proceso de Solicitud No. " &
fncMulo(rsol("Solicitud_ID"), 0)
AgregaReporte ""
If Not IsNull(rsol("Tipolinesa")) Then
    fncProcActivarExt.bolCorrecto = IIf(rsol("Tiponumero") = 1,
True, False)
    fncProcActivarExt.optIAsa.Value = IIf(rsol("Tipolinesa") = 1,
True, False)
    fncProcActivarExt.optIDig.Value = IIf(rsol("Tipolinesa") = 2,
True, False)
    fncProcActivarExt.Show vbModal
    fncProcMuestra.txtMuestra.Text = rsol("Display")
    fncProcMuestra.Show vbModal
    If Not bAceptar Then Exit Sub
    MDIPrincipal.MousePointer = 11
    DoEvents
    While Right(fncTelnet.txtPantalla.Text, 2) <> "<" & Chr(28)
And fncTelnet.txtPantalla.Find("PASSWORD") <> -1
        DoEvents
    End While
    If Right(fncTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
        sExtension = txtExtension.Text
        sMuestra = IIf(sMuestra = "", " ", sMuestra)
        sMuestra2 = IIf(sMuestra2 = "", " ", sMuestra2)
        If Not fncTelnet.FaEjecutaComando("MIINI:DIR=" &
sExtension & ",NAME1=" & Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) &
sMuestra2 & Chr(34) & ",PRES=1:") Then
            MsgBox "No es posible ejecutar comandos", vbCritical,
"Error"
        End Sub
    Else
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) Or fncTelnet.txtPantalla.Find("MIINI:DIR=" & sExtension & ",NAME1=" &
Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) & sMuestra2 & Chr(34) &
",PRES=1:") <> -1 Then
            AgregareReporte "Se asignó el nombre " & sMuestra &
" " & sMuestra2 & " a la extensión " & txtExtension.Text
            AgregareReporte ""
        End If
    End If
    MDIPrincipal.MousePointer = 0
End If
If Not IsNull(rsol("CodigoAut")) Then
    If txtExtension.Text <> "" Then
        Select Case rsol("CodigoAut")
            Case 3, 5, 6, 7
                fncProcCodigoAut.cboCategoria.ListIndex = 3
            Case 2, 4, 6, 7
                fncProcCodigoAut.cboCategoria.ListIndex = 2
            Case 1, 4, 5, 7
                fncProcCodigoAut.cboCategoria.ListIndex = 1
            Case Else
                fncProcCodigoAut.cboCategoria.ListIndex = 0
        End Select
        fncProcCodigoAut.Show vbModal
        If Not bAceptar Then Exit Sub
        MDIPrincipal.MousePointer = 11
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) And fncTelnet.txtPantalla.Find("PASSWORD") <> -1
            DoEvents
        End While
        If Right(fncTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
Then
            sExtension = txtExtension.Text
            If Not fncTelnet.FaEjecutaComando("AUCCO:DIR=" &
sExtension & ",") Then
                MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
            End Sub
        Else
            DoEvents
            While Right(fncTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",") <> -1
                DoEvents
            End While
            If Right(fncTelnet.txtPantalla.Text, 2) = "<"
& Chr(28) And fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",") <> -1
Then
                fncTelnet.LimpiaSelecciónPantalla
                iCounter =
fncTelnet.txtPantalla.Find("TYPE OF DIRECTORY NUMBER:") + 28
                sTipoExt = ""
                While Mid(fncTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And Mid(fncTelnet.txtPantalla.Text, iCounter, 1) <>
Chr(10) And iCounter < Len(fncTelnet.txtPantalla.Text) And iCounter > 27
                    sTipoExt = sTipoExt &
Mid(fncTelnet.txtPantalla.Text, iCounter, 1)
                    iCounter = iCounter + 1
                End While
                sTipoExt = IIf(Trim(sTipoExt) <> "",
Trim(sTipoExt), "ATN")
            End If
        End If
        If sTipoExt = "DTS" Then iCategoria = iCategoria +
17
        If Not fncTelnet.FaEjecutaComando("AUCCO:DIR=" &
sExtension & ",CILCOO=" & sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria &
",") Then
            MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
        End Sub
    Else
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
            DoEvents
        End While
        If Right(fncTelnet.txtPantalla.Text, 2) = "<"
& Chr(28) And fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
Then
            fncTelnet.LimpiaSelecciónPantalla
            iCounter =
fncTelnet.txtPantalla.Find("TYPE OF DIRECTORY NUMBER:") + 28
            sTipoExt = ""
            While Mid(fncTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And Mid(fncTelnet.txtPantalla.Text, iCounter, 1) <>
Chr(10) And iCounter < Len(fncTelnet.txtPantalla.Text) And iCounter > 27
                sTipoExt = sTipoExt &
Mid(fncTelnet.txtPantalla.Text, iCounter, 1)
                iCounter = iCounter + 1
            End While
            sTipoExt = IIf(Trim(sTipoExt) <> "",
Trim(sTipoExt), "ATN")
        End If
        If sTipoExt = "DTS" Then iCategoria = iCategoria +
17
        If Not fncTelnet.FaEjecutaComando("AUCCO:DIR=" &
sExtension & ",CILCOO=" & sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria &
",") Then
            MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
        End Sub
    Else
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
            DoEvents
        End While
        If Right(fncTelnet.txtPantalla.Text, 2) = "<"
& Chr(28) And fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
Then
            AgregareReporte "Se activó el código " &
sCodigo & " para categoría " & sCategoria
            AgregareReporte ""
        End If
    End If
    MDIPrincipal.MousePointer = 0
Else
    MsgBox "No fue capturada la extensión", vbCritical,
"Error"
End If
End If
If Not IsNull(rsol("DisplayNuevo")) Then
    fncProcMuestra.txtMuestra.Text = rsol("DisplayNuevo")
    fncProcMuestra.Show vbModal
    If Not bAceptar Then Exit Sub
    MDIPrincipal.MousePointer = 11
    DoEvents
    While Right(fncTelnet.txtPantalla.Text, 2) <> "<" & Chr(28)
And fncTelnet.txtPantalla.Find("PASSWORD") <> -1
        DoEvents
    End While
    If Right(fncTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
        sExtension = txtExtension.Text
        sMuestra = IIf(sMuestra = "", " ", sMuestra)
        sMuestra2 = IIf(sMuestra2 = "", " ", sMuestra2)
        If Not fncTelnet.FaEjecutaComando("MIINI:DIR=" &
sExtension & ",NAME1=" & Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) &
sMuestra2 & Chr(34) & ",PRES=1:") Then
            MsgBox "No es posible ejecutar comandos", vbCritical,
"Error"
        End Sub
    Else
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) Or fncTelnet.txtPantalla.Find("MIINI:DIR=" & sExtension & ",NAME1=" &
Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) & sMuestra2 & Chr(34) &
",PRES=1:") <> -1 Then
            AgregareReporte "Se asignó el nombre " & sMuestra &
" " & sMuestra2 & " a la extensión " & txtExtension.Text
            AgregareReporte ""
        End If
    End If
    MDIPrincipal.MousePointer = 0
End If

```

```

AgregaReporte ""
End If
End If
End If
MDIPrincipal.MousePointer = 0
Else
    MsgBox "No fue capturada la extensión", vbCritical,
"Error"
End If
End If
Else
    If Not IsNull(rsol("CodigoAut")) Then
        If txtExtension.Text <> "" Then
            Select Case rsol("CodigoAut")
                Case 3, 5, 6, 7
                    fncProcCodigoAut.cboCategoria.ListIndex = 3
                Case 2, 4, 6, 7
                    fncProcCodigoAut.cboCategoria.ListIndex = 2
                Case 1, 4, 5, 7
                    fncProcCodigoAut.cboCategoria.ListIndex = 1
                Case Else
                    fncProcCodigoAut.cboCategoria.ListIndex = 0
            End Select
            fncProcCodigoAut.Show vbModal
            If Not bAceptar Then Exit Sub
            MDIPrincipal.MousePointer = 11
            DoEvents
            While Right(fncTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) And fncTelnet.txtPantalla.Find("PASSWORD") <> -1
                DoEvents
            End While
            If Right(fncTelnet.txtPantalla.Text, 2) = "<" & Chr(28)
Then
                sExtension = txtExtension.Text
                If Not fncTelnet.FaEjecutaComando("AUCCO:DIR=" &
sExtension & ",") Then
                    MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
                End Sub
            Else
                DoEvents
                While Right(fncTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",") <> -1
                    DoEvents
                End While
                If Right(fncTelnet.txtPantalla.Text, 2) = "<"
& Chr(28) And fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",") <> -1
Then
                    fncTelnet.LimpiaSelecciónPantalla
                    iCounter =
fncTelnet.txtPantalla.Find("TYPE OF DIRECTORY NUMBER:") + 28
                    sTipoExt = ""
                    While Mid(fncTelnet.txtPantalla.Text,
iCounter, 1) <> Chr(13) And Mid(fncTelnet.txtPantalla.Text, iCounter, 1) <>
Chr(10) And iCounter < Len(fncTelnet.txtPantalla.Text) And iCounter > 27
                        sTipoExt = sTipoExt &
Mid(fncTelnet.txtPantalla.Text, iCounter, 1)
                        iCounter = iCounter + 1
                    End While
                    sTipoExt = IIf(Trim(sTipoExt) <> "",
Trim(sTipoExt), "ATN")
                End If
            End If
            If sTipoExt = "DTS" Then iCategoria = iCategoria +
17
            If Not fncTelnet.FaEjecutaComando("AUCCO:DIR=" &
sExtension & ",CILCOO=" & sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria &
",") Then
                MsgBox "No es posible ejecutar comandos",
vbCritical, "Error"
            End Sub
        Else
            DoEvents
            While Right(fncTelnet.txtPantalla.Text, 2) <> "<"
& Chr(28) Or fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
                DoEvents
            End While
            If Right(fncTelnet.txtPantalla.Text, 2) = "<"
& Chr(28) And fncTelnet.txtPantalla.Find("AUCCO:DIR=" & sExtension & ",CILCOO="
& sExtension & ",AUTH=" & sCodigo & ",CAT=" & iCategoria & ",") <> -1
Then
                AgregareReporte "Se activó el código " &
sCodigo & " para categoría " & sCategoria
                AgregareReporte ""
            End If
        End If
    End If
    MDIPrincipal.MousePointer = 0
Else
    MsgBox "No fue capturada la extensión", vbCritical,
"Error"
End If
End If
If Not IsNull(rsol("DisplayNuevo")) Then
    fncProcMuestra.txtMuestra.Text = rsol("DisplayNuevo")
    fncProcMuestra.Show vbModal
    If Not bAceptar Then Exit Sub
    MDIPrincipal.MousePointer = 11
    DoEvents
    While Right(fncTelnet.txtPantalla.Text, 2) <> "<" & Chr(28)
And fncTelnet.txtPantalla.Find("PASSWORD") <> -1
        DoEvents
    End While
    If Right(fncTelnet.txtPantalla.Text, 2) = "<" & Chr(28) Then
        sExtension = txtExtension.Text
        sMuestra = IIf(sMuestra = "", " ", sMuestra)
        sMuestra2 = IIf(sMuestra2 = "", " ", sMuestra2)
        If Not fncTelnet.FaEjecutaComando("MIINI:DIR=" &
sExtension & ",NAME1=" & Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) &
sMuestra2 & Chr(34) & ",PRES=1:") Then
            MsgBox "No es posible ejecutar comandos", vbCritical,
"Error"
        End Sub
    Else
        DoEvents
        While Right(fncTelnet.txtPantalla.Text, 2) <> "<" &
Chr(28) Or fncTelnet.txtPantalla.Find("MIINI:DIR=" & sExtension & ",NAME1=" &
Chr(34) & sMuestra & Chr(34) & ",NAME2=" & Chr(34) & sMuestra2 & Chr(34) &
",PRES=1:") <> -1 Then
            AgregareReporte "Se asignó el código " &
sCodigo & " para categoría " & sCategoria
            AgregareReporte ""
        End If
    End If
    MDIPrincipal.MousePointer = 0
End If

```

[illegible][illegible]


```

    CHAL.LATEL,VALUE = 8
    OPCHAL.DIG(8),VALUE = FALSE
    OPCHAL.DIG(7),VALUE = FALSE
    OPCHAL.DIG(6),VALUE = FALSE
    OPCHAL.DIG(5),VALUE = FALSE
    OPCHAL.DIG(4),VALUE = FALSE
    OPCHAL.DIG(3),VALUE = FALSE
    OPCHAL.DIG(2),VALUE = FALSE
    OPCHAL.DIG(1),VALUE = FALSE
    EXTERNALS.LAPLAY,TEXT = ""
END

```

```

Caption = "Cambio de Directo por Estensiynt"
Height = 255
Left = 120
TabIndex = 56
Top = 1335
Width = 2655
End
Begin VB.CheckBox chkMovExt
Caption = "Mover de Lugar una Estensiynt"
Height = 255
Left = 120
TabIndex = 55
Top = 960
Width = 2775
End
Begin VB.CheckBox chkRepTel
Caption = "Reposicion de Aparato Telefonico"
Height = 255
Left = 120
TabIndex = 54
Top = 800
Width = 3255
End
Begin VB.CheckBox chkChombChip
Caption = "Nombre en Display"
Height = 255
Left = 120
TabIndex = 53
Top = 240
Width = 1815
End
Begin VB.TextBox txtNumAct
Height = 285
Left = 4200
TabIndex = 51
Top = 1320
Width = 1575
End
Begin VB.TextBox txtChodoNuevo
Height = 285
Left = 4200
TabIndex = 49
Top = 945
Width = 1575
End
Begin VB.PictureBox Picture2
Appearance = 0 'Flat
BackColor = &H80000004
BorderStyle = 0 'None
ForeColor = &H80000008
Height = 195
Left = 3600
ScaleHeight = 195
ScaleWidth = 2175
TabIndex = 48
Top = 600
Width = 2175
Begin VB.OptionButton OptCanDig
Caption = "Analógico"
Height = 195
Index = 0
Left = 0
TabIndex = 80
Top = 0
Width = 1095
End
Begin VB.OptionButton OptCanDig
Caption = "Digital"
Height = 195
Index = 1
Left = 1080
TabIndex = 50
Top = 0
Width = 800
End
End
Begin VB.TextBox txtChombChip
Height = 285
Left = 3360
TabIndex = 41
Top = 240
Width = 2415
End
Begin VB.Label lblNumAct
AutoSize = -1 'True
Caption = "Numero Actual:"
Height = 195
Left = 3000
TabIndex = 52
Top = 1385
Width = 1095
End
Begin VB.Label lblNodoN
AutoSize = -1 'True
Caption = "Nodo Nuevo:"
Height = 195
Left = 3120
TabIndex = 50
Top = 960
Width = 990
End
End
Begin VB.Frame Frame4
Height = 1095
Left = 120
TabIndex = 39
Top = 240
Width = 5895
Begin VB.TextBox txtCodAut
Height = 285
Left = 4800
TabIndex = 47
Top = 240
Width = 855
End
Begin VB.CheckBox chkCodAut
Caption = "Codigo de Autorizaci3n"
Height = 375
Left = 120
TabIndex = 45
Top = 195
Width = 2955
End
Begin VB.CheckBox chkLocal
Caption = "Local"
Height = 255
Left = 2520
TabIndex = 44
Top = 860
Width = 735
End
End
Begin VB.CheckBox chkNacional
Caption = "Nacional"
Height = 255
Left = 3360
TabIndex = 43
Top = 660
Width = 975
End
Begin VB.CheckBox chkCInter

```

```

Caption = "Internacional"
Height = 255
Left = 4440
TabIndex = 42
Top = 660
Width = 1215
End
Begin VB.Label lblCodAut
AutoSize = -1 'True
Caption = "Codigo Autorizaci3n"
Height = 195
Left = 3800
TabIndex = 46
Top = 285
Width = 1125
End
End
Begin VB.CommandButton btnAceptar
Caption = "Aceptar"
Height = 380
Left = 5180
TabIndex = 32
Top = 6800
Width = 1230
End
Begin VB.Frame Frame3
Caption = "Activaci3n"
Enabled = 0 'False
Height = 3655
Left = 120
TabIndex = 20
Top = 3000
Width = 6135
Begin VB.TextBox txtActivaci3n
Height = 285
Left = 4440
TabIndex = 27
Top = 2025
Width = 1575
End
Begin VB.CheckBox chkAlter
Caption = "Interseccional"
Height = 255
Left = 4560
TabIndex = 26
Top = 1580
Width = 1215
End
Begin VB.CheckBox chkNacional
Caption = "Nacional"
Height = 255
Left = 3480
TabIndex = 25
Top = 1580
Width = 975
End
End
Begin VB.CheckBox chkLocal
Caption = "Local"
Height = 255
Left = 2040
TabIndex = 24
Top = 1580
Width = 735
End
End
Begin VB.TextBox txtNomDisplay
Height = 285
Left = 3480
TabIndex = 23
Top = 960
Width = 2415
End
Begin VB.CheckBox chkCodAut
Caption = "Codigo de Autorizaci3n"
Height = 375
Left = 240
TabIndex = 22
Top = 1580
Width = 2955
End
Begin VB.CheckBox chkAlinTel
Caption = "Linee Telefonica"
Height = 375
Left = 240
TabIndex = 21
Top = 600
Width = 1575
End
End
Begin VB.Frame Frame1
Height = 1095
Left = 120
TabIndex = 29
Top = 240
Width = 5895
Begin VB.PictureBox Picture1
Appearance = 0 'Flat
BackColor = &H80000004
BorderStyle = 0 'None
ForeColor = &H80000008
Height = 615
Left = 3360
ScaleHeight = 615
ScaleWidth = 1215
TabIndex = 35
Top = 120
Width = 1215
Begin VB.OptionButton OptAlinDig
Caption = "Digital"
Height = 195
Index = 1
Left = 0
TabIndex = 37
Top = 380
Width = 1095
End
Begin VB.OptionButton OptAlinDig
Caption = "Analógico"
Height = 195
Index = 0
Left = 0
TabIndex = 36
Top = 120
Width = 1095
End
End
Begin VB.OptionButton OptADirExt
Caption = "Extensiynt"
Height = 255
Index = 1
Left = 4680
TabIndex = 34
Top = 480
Width = 1095
End
Begin VB.OptionButton optADirExt
Caption = "Directo"
Height = 255
Index = 0
Left = 4680
TabIndex = 33

```


[illegible]

[illegible]

```
set espion = counterfactual.execute(sqlquery)
```



```

End If
If iCuenta >= 10000 Then
    FnConecta = False
End If
End With

End Function

Public Function FnLogOff() As Boolean
    Function: InLogOff
    Description: Desconecta al conmutador
    User: bVariable = InLogOff()
    Dim iCuenta As Long

    If WinsockClient.State <> sockClosed Then
        WinsockClient.SendData Chr(24)
        DoEvents
        iCuenta = 0
        While Right(bPantalla.Text, 2) <> "<" & Chr(26) And iCuenta < 10000
            iCuenta = iCuenta + 1
            DoEvents
        Wend
        If Right(bPantalla.Text, 2) = "<" & Chr(26) Then
            FnEjecutaComando "O"
            DoEvents
            iCuenta = 0
            While Right(bPantalla.Text, 11) <> "LOGGED OFF" & Chr(250) And iCuenta < 10000
                iCuenta = iCuenta + 1
                DoEvents
            Wend
            WinsockClient.Close
            DoEvents
            sRecibido = ""
            bPantalla.Text = ""
        End If
        If iCuenta >= 10000 Then
            FnLogOff = False
        Else
            FnLogOff = True
        End If
    End If
End Function

Public Function FnEjecutaComando(sComando As String) As Boolean
    Function: FnEjecutaComando
    Description: Ejecuta un comando en el conmutador
    User: bVariable = FnEjecutaComando(sComando)

    If WinsockClient.State <> sockClosed Then
        WinsockClient.SendData sComando & Chr(13) & Chr(10)
        sRecibido = ""
        FnEjecutaComando = True
    Else
        FnEjecutaComando = False
    End If
End Function

Public Sub EnviasS()
    Dim iCounter As Integer
    If WinsockClient.State <> sockClosed Then
        bSenEnvias = True
        iCounter = 0
        InvTimer.Interval = 5
    End If
End Sub

Private Sub btnConecta_Click()
    FnConecta InputBox("Direccion", "Direccion", "110.208.150.254")
End Sub

Private Sub btnEnviasS_Click()
    EnviasS
End Sub

Private Sub btnLogOff_Click()
    FnLogOff
End Sub

Private Sub InvTimer_Timer()
    If Right(bPantalla.Text, 3) <> "SSS" Then
        WinsockClient.SendData "S"
    Else
        InvTimer.Interval = 0
        WinsockClient.SendData "" & Chr(13) & Chr(10)
    End If
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    WinsockClient.Close
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Unload Me
End Sub

Private Sub Test2_KeyPress(KeyAscii As Integer)
    If WinsockClient.State <> sockClosed Then
        WinsockClient.SendData Chr(KeyAscii)
    End If
End Sub

Private Sub WinsockClient_Close()
    With WinsockClient
        .Close
        .RemotePort = 0
        .LocalPort = 0
    End With
    bConectado = False
End Sub

Private Sub WinsockClient_Connect()
    Dim sConnectString As String

    bSw_igoohead = True
    bSw_igoohead = False
    bSw_echo = True
    bSw_linemode = False
    bSw_terminent = False
    bSubstate = False

    sConnectString = Chr$(AC) & Chr$(DOTE) & Chr$(ECHO) & _
        Chr$(AC) & Chr$(DOTE) & Chr$(SGA) & _
        Chr$(AC) & Chr$(WILLTEL) & Chr$(NAWS) & _
        Chr$(AC) & Chr$(WILLTEL) & Chr$(TERMSPEED) & _
        Chr$(AC) & Chr$(WILLTEL) & Chr$(TERMSPEED)

    WinsockClient.SendData sConnectString
End Sub

Private Sub WinsockClient_DataArrival(ByVal bytesTotal As Long)
    Dim byCH() As Byte
    Dim iTest() As Integer
    Dim iCounter As Integer
    Static iCmd As Byte

    If Not bReceiving Then
        bReceiving = True

```

```

Else
    Exit Sub
End If

If (bytesTotal > 0) Then

    WinsockClient.GetData byCH, vbByte + vbArray, bytesTotal

    For iCounter = 0 To bytesTotal - 1
        Select Case iCmd
            Case GO_ICMD
                iCmd = FnProcess_Chcr(byCH(iCounter))
            Case GO_IAC1
                iCmd = iNAC1(byCH(iCounter))
            Case GO_IAC2
                iCmd = iNAC2(byCH(iCounter))
            Case GO_IAC3
                iCmd = iNAC3(byCH(iCounter))
            Case GO_IAC4
                iCmd = iNAC4(byCH(iCounter))
            Case GO_IAC5
                iCmd = iNAC5(byCH(iCounter))
            Case GO_IAC6
                iCmd = iNAC6(byCH(iCounter))
            Case GO_IAC7
                iCmd = iNAC7(byCH(iCounter))
        End Select
    Next iCounter
End If

bPantalla.Text = sRecibido
bReceiving = False
End Sub

Private Function iNAC1(CH As Byte) As Integer

    iNAC1 = GO_NORM

    Select Case CH
        Case DOTE
            iNAC1 = GO_IAC2
        Case DONTTEL
            iNAC1 = GO_IAC6
        Case WILLTEL
            iNAC1 = GO_IAC3
        Case WONTTEL
            iNAC1 = GO_IAC4
        Case SB
            iNAC1 = GO_IAC5
        Case SE
            iNAC1 = GO_IAC6
        Case TERMTYPE
            If iPantalla(1) = 1 Then
                WinsockClient.SendData Chr$(AC) & Chr$(SB) & Chr$(TERMTYPE) & "DEC-VT100" & Chr$(0) & _
                    Chr$(AC) & Chr$(SE)
            End If
        Case TERMSPEED
            If iPantalla(1) = 1 Then
                WinsockClient.SendData Chr$(AC) & Chr$(WILLTEL) & Chr$(CH) & _
                    WinsockClient.SendData Chr$(AC) & Chr$(SB) & _
                    Chr$(TERMSPEED) & Chr$(0) & _
                    Chr$(AC) & Chr$(SE)
            End If
        End Select
    End Function

Private Function iNAC2(CH As Byte) As Integer

    iNAC2 = GO_NORM

    Select Case CH
        Case BINARY
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(BINARY)
        Case ECHO
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(ECHO)
        Case NAWS
            WinsockClient.SendData Chr$(AC) & Chr$(SB) & Chr$(NAWS) & Chr$(0) & Chr$(0) & Chr$(0) & _
                Chr$(24) & Chr$(AC) & Chr$(SE)
        Case SGA
            If Not bSw_igoohead Then
                WinsockClient.SendData Chr$(AC) & Chr$(WILLTEL) & Chr$(SGA)
            End If
        Case TERMTYPE
            If Not bSw_terminent Then
                WinsockClient.SendData Chr$(AC) & Chr$(WILLTEL) & Chr$(TERMTYPE) & _
                    WinsockClient.SendData Chr$(AC) & Chr$(SB) & Chr$(TERMTYPE) & _
                    Chr$(0) & "VT100" & Chr$(AC) & Chr$(SE)
            End If
        Case TERMSPEED
            WinsockClient.SendData Chr$(AC) & Chr$(WILLTEL) & Chr$(TERMSPEED) & _
                WinsockClient.SendData Chr$(AC) & Chr$(SB) & Chr$(TERMSPEED) & Chr$(0) & _
                WinsockClient.SendData "57600,57600" & _
                WinsockClient.SendData Chr$(AC) & Chr$(SE)
        Case TFLOWCTRL
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case LINEMODE
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case STATUS
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case TIMING
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case DISLOC
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case ENVIRON
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case UNKNOWN39
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        Case AUTHENTICATION
            WinsockClient.SendData Chr$(AC) & Chr$(WILLTEL) & Chr$(CH)

            WinsockClient.SendData Chr$(AC) & _
                Chr$(SB) & _
                Chr$(AUTHENTICATION) & _
                Chr$(0) & Chr$(0) & Chr$(0) & Chr$(0) & _
                Chr$(AC) & _
                Chr$(SE)
        Case Else
            WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
        End Select
    End Function

Private Function iNAC3(CH As Byte) As Integer

```

```

INACS = GO_NORM
Select Case CH
Case ECHO
If Not bSw_echo Then
bSw_echo = True
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(ECHO)
End If
Case SGA
If Not bSw_ugoshead Then
bSw_ugoshead = True
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(SGA)
End If
Case TERMSPEED
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case TFLOWCNTRL
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case LINEMODE
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case STATUS
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case TIMING
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case DISPLC
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case ENVIRON
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case UNKNOWN9
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case Else
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
End Select
End Function

Private Function INACA(CH As Byte) As Integer
INACA = GO_NORM
Select Case CH
Case ECHO
If bSw_echo = True Then
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(ECHO)
bSw_echo = False
End If
Case SGA
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(SGA)
bSw_ugoshead = False
Case TERMSPEED
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case TFLOWCNTRL
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case LINEMODE
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case STATUS
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case TIMING
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case DISPLC
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case ENVIRON
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case UNKNOWN9
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
Case Else
WinsockClient.SendData Chr$(AC) & Chr$(DONTTEL) & Chr$(CH)
End Select
End Function

Private Function INACS(CH As Byte) As Integer
Dim ich As Integer
ich = CH
If ich = IAC Then
INACS = GO_IAC1
Exit Function
End If
iPpno = (Ppno) = ich
iPpno = iPpno + 1
INACS = GO_IACS
End Function

Private Function INACB(CH As Byte) As Integer
INACB = GO_NORM
Select Case CH
Case SE
Case ECHO
If Not bSw_echo Then
bSw_echo = True
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(ECHO)
End If
Case SGA
If Not bSw_ugoshead Then
bSw_ugoshead = True
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(SGA)
End If
Case TERMSPEED
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case TFLOWCNTRL
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case LINEMODE
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case STATUS
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case TIMING
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case DISPLC
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case ENVIRON
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case UNKNOWN9
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
Case Else
WinsockClient.SendData Chr$(AC) & Chr$(WONTTEL) & Chr$(CH)
End Select
End Function

Public Sub LimpiaSelecionParallela()
bParallelaSelStart = 0
bParallelaSelLength = 0
End Sub
Attribute VB_Name = "v100"
Title: v100
Module: v100.bas

```

Version: 1.0
Fecha: 23/04/2000
Autor: Hector Manuel Garza Garza

Option Explicit

Private Declare Function IsCharAlpha Lib "user32" Alias "IsCharAlpha" (ByVal cChar As Byte) As Long

Private Const GO_IAC1 = 6

Private bEscAsci As Boolean

Private sEscString As String

Public sRecibido As String

Public Function FrProcess_Char(CH As Byte)

If (bEscAsci) Then

Call EscapeProcess(CH)

Else

Select Case CH

Case 7

Beep

Case 8

sRecibido = Mid(sRecibido, 1, Len(sRecibido) - 1)

Case 9

sRecibido = sRecibido & " "

Case 10, 11, 12

sRecibido = sRecibido & Chr(10)

Case 13

Case 27

bEscAsci = True

sEscString = ""

Case 255

FrProcess_Char = GO_IAC1

Case Else

sRecibido = sRecibido & Chr(CH)

End Select

End If

End Function

Private Sub EscapeProcess(CH As Byte)

Dim c As String

Dim yOff As Integer

Dim xOff As Integer

c = Chr(CH)

If sEscString = "" Then

Select Case c

Case Chr(8)

sRecibido = Mid(sRecibido, 1, Len(sRecibido))

bEscAsci = False

Case 7

bEscAsci = False

Case 8

bEscAsci = False

Case 9

sRecibido = sRecibido & Chr(10)

bEscAsci = False

Case 10

sRecibido = sRecibido & Chr(10)

bEscAsci = False

Case 11

sRecibido = sRecibido & " "

bEscAsci = False

Case 12

bEscAsci = False

Case 13

bEscAsci = False

Case 27

bEscAsci = False

Case Else

bEscAsci = False

Exit Sub

End Select

End If

sEscString = sEscString & c

If IsCharAlpha(CH) = 0 Then

If Len(sEscString) > 15 Then

bEscAsci = False

End If

Exit Sub

End If

Select Case c

Case "A"

sEscString = Mid(sEscString, 2)

Case "B"

sEscString = Mid(sEscString, 2)

Case "C"

sEscString = Mid(sEscString, 2)

Case "D"

sEscString = Mid(sEscString, 2)

Case "E"

sEscString = Mid(sEscString, 2)

Case "F"

sEscString = Mid(sEscString, 2)

Case "G"

sEscString = Mid(sEscString, 2)

Case "H"

sEscString = Mid(sEscString, 2)

Case "I"

sEscString = Mid(sEscString, 2)

End Select

bEscAsci = False

sEscString = ""

End Sub

ANEXO C – REFERENCIAS DE INTERNET



June 1999 Issue, copyright 1999, Canada Computer Paper Inc.

Java and Visual Basic: programming's new breed

By Jacques Surveyer

Sure, C and, increasingly, C++ are the languages of choice for major independent software vendors (ISVs) like Corel, IBM, Microsoft, Oracle, SAS, and Symantec. They use these programming languages to build their bread-and-butter applications like CorelDraw, DB2 or Microsoft Office. But in the past six years two new programming languages--Sun Microsystems' Java and Microsoft's Visual Basic (VB)--have emerged and have quickly risen to dominant positions in software development. Even oldie-but-goodie languages like Ada, Cobol, Fortran, Pascal, and RPG are being refurbished with new graphical user interfaces (GUIs) and visual programming IDEs just like Java and Visual Basic.

So what makes these new languages so compelling? The answer for both is visual programming and its components. Visual Basic was one of the first languages, along with Paradox and Dbase, designed to take advantage of GUIs with visual programming in mind. Users build menus, forms and reports for a program using visual drag-and-drop operations. They then complete the coding by filling in the skeleton programs developed when creating the prototype of a program's interface.

Coding Visual Basic pioneered the use of standard third-party components such as VBX and now ActiveX. The developers of Java IDE (Interactive Development Environments) have carried this a step further with interaction wizards, which automatically generate large portions of the Java code required for a program. And Visual Basic, in the slightly simplified form of VBA (Visual Basic for Applications), has become the macro language for Windows applications. The result is two of the best rapid application development (RAD) programming languages available today.

But some major trends in program usage may render all this obsolete. One critical trend is the exploding use of the Internet, in which the Web browser becomes the interface of choice. A second trend is the continuing migration to packaged software starting with the office suites but extending into every corner of work. Examples are: document

management systems; Web, mail and comprehensive messaging systems like Lotus Domino/Notes, Microsoft Exchange and Novell's GroupWise; hundreds of financial and accounting packages; and, most tellingly, enterprise resource planning (ERP) suites that form the backbone for complete organizational systems.

The question is: so what if Java and VB are emerging as the top of the programming heap. Who needs to program when packaged programs deliver the goods with increasing ease of use and comprehensive coverage ? Is programming as we have known it for the past 30 to 40 years dead?

Yes and no. The crucial factor is that programming is changing dramatically. There are three major trends that are changing why and how programming is done.

Two primary trends in programming

The first trend is that the last remaining heights of programming complexity are being successfully scaled right now. Internet and Web development have meant that the toughest programming problem--distributed processing over wide area networks (WANs)--is finally falling out of the realm of high-risk, specialist-only development into a demanding but achievable task.

Microsoft's vice-president of development, David Vaskevitch, calls distributed development the last hurdle. Sure, challenges such as artificial intelligence (AI), multi-threaded parallel programming, and adaptive programming (automated code that detects and then fixes problems and bugs) will remain. But n-tier (multiple active servers and programs), heterogeneous (acting simultaneously on many different hardware, operating systems and database platforms), and distributed (acting in concert by communicating over one or more networks) programming are not just hard to describe but have also been--until open Internet-inspired programming tools and routines including Java became available--darn hard to do.

The second major trend is object-oriented (OO) languages, which do three things for programming in general.

First, they bring a new emphasis on reliable coding by deliberately hiding data and code except through controlled access points. They have built-in strong typing and reference checking. In effect, some of the flexibility to dynamically change code or data usage is sacrificed to make it work more reliably in OO languages like Java, C++, and Smalltalk.

Second, program development is speeded up and made more reliable by the inheritance of OOs. Unlike subroutine reuse, inheritance goes further by specifying exactly what data and routines will be changed or added to when borrowing and re-using classes (think of classes as code modules with data attached). Reuse of classes through inheritance is remarkably clever and has been used, in an ad-hoc fashion, for quite some

time by programmers. OOs just standardize and automate the process. The net result is that coding in Java is found to be 30 to 70 percent faster than in its close cousin, C.

The third innovation OO languages bring to programming is polymorphism and dynamic linking. This is the ability of an OO language to allow for the same method or subroutine to carry very different parameters with the runtime system invoking the right subroutine, including a possible error recovery routine. For example, "ScreenDraw (Circle, xx, yy)" and "ScreenDraw (Hexagon, xx, yy)" dynamically and automatically invoke two different ScreenDraw routines depending on the parameter passed. This may seem trivial, but a great programming house of cards may come crashing down trying to use ad-hoc methods to solve this problem.

It is important to note that Java and Visual Basic have taken two profoundly different approaches to OO programming. From the start, Java was designed as an OO language. You cannot write a Java program without using OO techniques.

Visual Basic has taken the slow-but-sure approach, so OO techniques and methods have been gradually incorporated into the language. Polymorphism and dynamic linking, followed by encapsulation and hiding, were added in Visual Basic 3.0 through 5.0. Inheritance may be added in version 7.0 or 8.0 (or may be a part of a new Microsoft language code-named Cool).

The bottom line is that Visual Basic is a hybrid OO language, for better or worse. Many timberlines worth of paper have been sacrificed in arguing whether a pure OO language like Java or Smalltalk is superior to a hybrid like C++ or Visual Basic. We invite you to decide for yourself in our upcoming tutorials on VB and Java.

The third trend: Visual programming and components

As noted previously, visual programming and the use of components is one of the pioneering strengths of Java and Visual Basic. OO methods make GUI and general program development using components even easier. But these techniques have been spawned by two very real needs. N-tier distributed processing has brought two real benefits: the elimination of both isolated islands of information and inefficient use of data and program resources.

With intranets and online processing, users can reach into every nook and cranny of an organization to get at the data and information they need to solve a problem. But the cost, as noted, is the much greater complexity of distributed programs--even ones that use Web browser front ends. Hence the move to outsourcing and/or packaged programs.

However, a second pressing need is to develop software much faster. For over 30 years two of the major complaints against IT departments have been that systems cannot be developed fast enough (hence, the two-to-three year backlog of IT projects) and when

they are delivered, changes and updates to programs cannot be done on a timely basis. But with the huge repository of computer systems built up over the last 30 years, and with the availability of complete backbone office systems from major ERP vendors, programming has been turned upside down.

Now IT staffs are being told "no" to re-inventing the IT wheel. It takes too long and is fraught with the high risk of complete failure (there is a 50 percent failure rate for large client/server projects according to the Standish Reports). Reuse and interfacing to new ERP backbones, as well as legacy systems, are the new marching orders for IT development teams at small as well as large businesses.

In effect, all programming has become maintenance programming. Very few programs are built from scratch. Programs have to interface with legacy databases or Web systems. Many have to link to or just become a customized user-called procedure of some ERP or other packaged program. What better way to create these component systems than--bingo!--visual programming and component development with Java and Visual Basic.

The essential advantages of Java and Visual Basic

Without a doubt, one of the essential advantages of both Java and Visual Basic is that programmers can develop a wide range of programs so much faster than with other languages. So-called 4GLs (fourth-generation languages) like Focus, PowerBuilder and Uniface once held an advantage in ease and speed of development (but at the cost of poor runtime performance).

But 3GLs (third-generation languages), especially Java and Visual Basic with their visual programming, components and use of clever wizards or third-party design tools, have matched if not surpassed 4GLs for speed of development and have superior runtime performance. Relative to other 3GLs, such as Ada, C/C++, Cobol, Fortran, and Pascal, Java and Visual Basic have three compelling advantages:

- * new, uniform and comprehensive APIs for every aspect of Web, client/server, and n-tier programming;
- * several modes of deployment including .EXEs, components, servlets and applets;
- * and superior visual programming development environs with many third-party suppliers of tools and components.

And the runtime performance of both Java and Visual Basic have steadily improved to the point that they are approaching within 10 to 30 percent of the fastest programs

developed in C/C++, Cobol, Fortran or Pascal. Finally, Java in particular but also Visual Basic can be used to deliver Web programs.

On the Web, Java and Visual Basic can be used as servlets. These are Web applications sitting on a server and sending down dynamically created HTML and scripts to your browser. Java, with its great cross-platform portability, can also be used to develop applets that run directly on any PC or client browser. With new ADSL and cable modems running 20 times faster than current 56Kbps modems, look for even more exciting (and secure) Java applets coming to your Web browser.

In sum, both Java and Visual Basic have come to the fore as programming languages because they were built to fit the mold for new system development, with rapid development through visual IDEs and comprehensive SDK/APIs to meet all the latest programming requirements, especially the Web.

Perhaps even more important are the OO methods and component-like code, which can stand alone or be linked to existing programs and data sources in a variety of ways, including servlet, downstream application, application server or Web applet. It is this versatility in deployment that makes both languages so attractive.

Over the next four or five years, the proliferation in programming languages should subside. This situation has persisted so long as client/server and n-tier processing needs were not being well served. But the current balance of comprehensive APIs, rapid development for a variety of deployment modes and competitive runtime performance make Java and Visual Basic compelling choices for many business applications.

So expect programming languages to gravitate towards the specialties they serve well: Prolog and some LISP variants for AI, Fortran for intensive engineering/numeric analysis systems; C/C++ for commercial software development; Cobol for transaction processing; and most notably Java and Visual Basic leading the way on the new Web and component-based systems that act as links and linchpins among back-office applications.

Java and Visual Basic comparison

On the surface, Java and Visual Basic share some common and compelling characteristics. Both are comparatively easy to learn and to begin developing useful, small programs with due to their approachability. In addition, both have very comprehensive APIs with the latest in Web and n-tier developments. Indeed if you can't find what you need directly in the language, both have a wealth of third-party tools, utilities and components available to complete their effectiveness.

Finally, both languages have the critical advantage of many modes of deployment. Java or Visual Basic code can stand alone, be bolted on as components of a larger system, or act as server-side linchpins between legacy systems, data sources, and PC and/or

Web browser clients. This is why Java and Visual Basic should emerge as the dominant programming languages over the next five to 10 years.

However, there are three major differences between the two languages. First, as noted previously, Java has been built from the ground up as an OO language, while Visual Basic is gradually acquiring complete OO capabilities. Second, Java has a number of security and reliability features (sandbox security model, strong typing, simplified memory model, integrity checking, etc.) built right into its core design.

But the third, and perhaps the critical difference, is that Java is cross-platform--running on just about any combination of hardware and operating system--while Visual Basic is master of Windows. VB runs on Windows 9x, NT (partly on Alpha versions), 2000 (the new name for NT 5.0), and CE (partly). However, Visual Basic no longer supports Windows 3.x, while Java does.

On the other hand, Visual Basic has become, through VBA, the scripting language for Windows applications. This is important because more systems will be built directly into key applications such as AutoCAD, Excel, Notes, Project, SAS, SAP, Word and hundreds of other major Windows programs.

If we compare Java and Visual Basic in detail, there are four major criteria we can use. The first is ease of development. It is here that Visual Basic would appear to have an advantage. After all it is basic. However, like Java, Visual Basic has grown and has huge APIs. More telling is that, in order to accommodate new OO and other functionality, Visual Basic has had to expand its basic syntax by dozens of commands. Even worse, the documentation for Visual Basic has deteriorated with recent releases. Explanations of commands are spread over three large manuals--printed versions of which cost more than \$100. Quick examples of how to use Visual Basic's numerous commands and functions are declining even in the electronic documentation.

Java is only slightly better. Fortunately, Java's huge API is organized into comprehensive references. But like Visual Basic, users have to look to third parties for documentation with quick examples of Java code. The fundamental problem is this: although developers of both languages have worked hard to make them easy to approach and learn, the sheer size and scope of their functionality make them difficult to thoroughly master.

Stability and reliability

Traditionally, programming languages have had very good records of stability and reliability for both development systems and the program code they produce. However, in the rush to get newest versions and features out the door, both Java and Visual Basic have let down developers.

Part of this may be inevitable. Rapid changes in Web and programming standards mean

some code and APIs become obsolete, as for example when many Abstract Windows Toolkit (AWT) classes were quickly superceded by the new event-listener classes and Swing GUI components and interfaces. In Visual Basic, a similar pattern has seen two or three data access methods proliferate into more than a dozen. Unfortunately, performance, reliability, and functionality trade-offs among the competing access methods make choosing an approach in Visual Basic a daunting task, even with the new OLE-DB standard. But perhaps the worst problem is the frequency of bugs in both languages.

To an extent bugs come with rapid change and huge APIs. But for the last three versions of Visual Basic, users have had to wait for the release of two or even three service packs to get reliable code for some new features either in the development system or in the language. (See the editorial in April 1999 *Visual Basic Programmer's Journal* "Five Things You Hate about VB6" for the latest episode on Visual Basic bugs.) Yes, most of the old code works. Yes, most of the bugs are clustered around new features. No, most users do not want to develop with what is in effect beta code and features. Want to kill a function or feature in a language? Make it buggy so developers have to spend extra hours and days proving that the bug is in the language not their code.

Java is somewhat better. But in the rush to get various releases of JDK 1.x out the door serious bugs crept into some of the new APIs. Also, compatibility of the GUI routines on various operating system platforms has lead to pungent humor--i.e., Java is the language you get to write once and debug everywhere.

Now it is important to point out that these are not catastrophic bugs by any means. Ninety five percent or more of the functionality of both languages is rock solid. However, when you are developing programs using any features in the bug-infested swamp that comprises the remaining five percent, you are not a happy camper.

In their defense, both Microsoft and Sun have dedicated themselves to producing more reliable code. Microsoft, for example, has made service packs easier to access and more quickly available. Sun has released later versions of JDK 1.x in smaller and more reliable chunks while delaying key features like some EJBs and Hotspot technology for many months.

But there is a lot at stake. Other languages like C/C++, ObjectPascal/Delphi, or the new versions of Cobol and PowerBuilder have better reliability records. In effect, by delivering buggy code and/or development systems, Java and Visual Basic may jeopardize their own critical advantage--speed of development.

Speed of development: The ultimate criteria?

Visual Basic is famous for making tough Windows programming very approachable. And the text editor and debugger in Visual Basic's IDE are still the best in the business

for client-side development. But Java is no slouch either. Java vendors have produced nearly comparable IDEs and have gone ahead of Visual Basic in the sophistication of their wizards and auto-generation capabilities. Borland JBuilder, IBM Visual Age, and Symantec Visual Cafe make the task of using the huge Java API more manageable with their interaction wizards.

Perhaps because there have been so many bugs and reliability problems, both Java and Visual Basic have very solid debuggers and third-party testing tools. Visual Basic is still the only major language that allows users to stop a program during debugging, change one or more lines of code and, in most cases, immediately resume running the program either from where it was halted or from a point determined by the user. Despite the fact that Java has sophisticated just-in-time (JIT) compilers and very fast interpreters, Java IDEs have only a pale approximation of this immediate restart debugging capability. However, on the profiling and testing side, both Java and Visual Basic have the support of some excellent testing tools. But perhaps the most compelling reason for speed of development is paradoxically the huge APIs.

Given short development time and complex coding requirements, reuse of tried and tested (hence the concern about bugs) Java or Visual Basic API code can and does cut out major chunks of development time. When time to market is the critical success factor for most businesses, speed and reliability of software developments that drive market-winning innovations also come to the fore.

The whole idea behind OO programming is structured inheritance and reuse of code. What Java and Visual Basic bring to the table is not only reuse but also the ability to deliver code in deployment packages (executables, components, servlets and applets) suited to a wide range of needs.

When to use

It is obvious from the chart that there are trade-offs between the two languages. In the coming months, *The Computer Paper* will feature tutorials on both Visual Basic and Java so you get to see some of the coding pros and cons in detail.

But right now you may need to decide which language to use and when. For applications on the server side--especially Web applications--Java has a distinct advantage simply because it runs on more enterprise-class operating systems. Also, major Web application server vendors such as BEA, Bluestone, IBM, Oracle, Sun and others have placed Java development ahead of even C/C++.

Visual Basic has a server niche in all-Windows shops, particularly when used in conjunction with Visual InterDev and/or FrontPage to develop ASP-based applications.

For PC-based applications, where Windows dominates the desktop, the situation is

reversed. Because Visual Basic is slowly but surely becoming the scripting language for Windows, it is easier to bolt Visual Basic through COM and other components onto existing commercial or homegrown Windows apps.

If, on the other hand, your delivery model uses the Net and a browser interface, you may want to give Java a careful look. The new Swing classes make Java GUI operations very fast (download a free copy of Together/J from <http://www.oi.com> to see exactly how fast pure Java can be on a PC).

In fact many analysts argue that with steady improvements, Java and Visual Basic will be matching C/C++ for speed of execution. Many Java applications do not take advantage of second-generation byte code optimizers, rarely use the new breed of Java compilers (the trade-off is less portable code vs. a 20 to 60 percent speed improvement), and are still waiting for the speed boost of Sun's dynamic Hotspot technology. Therefore, Java has plenty of performance-tuning capacity.

In contrast, Visual Basic 5.0 did show steady improvements to the point of reaching within 10 to 20 percent of equivalent C/C++ code. However, Visual Basic 6.0 saw the speed advantage slip by five to 15 percent in a number of categories--most notably in database access. But analysts argue that with steady compiler improvements still to come along with CPU performance continually improving also, both Java and Visual Basic will be taking on more enterprise programming tasks over the next five to 10 years.

Conclusion

If you have read all the way through to this point in the article, you undoubtedly picked up its tenor: great admiration for Java and Visual Basic mixed with ample frustration. Great programming languages like Java and Visual Basic should not be dogged by poor documentation or reliability and bug problems, especially when those small but significant percentages of bugs detract from the languages' greatest strength: speed of development.

Visual Basic, in particular, after being such a pioneer in rapid application development, is disappointing its constituency yet again with another version whose new features are buggy (see "Confessions of a VB Beta Tester," in the April 1999 issue of *Visual Basic Programmer's Journal*). However, the proof is in the programming.

Fortunately, both languages are readily available. Go to <http://www.sun.com/java> and download the latest JDK, which includes javac for compiling and java for running Java programs. Just add your favorite text editor (notepad.exe will do).

And Visual Basic is available in a learning edition for less than \$100. Or look for VBA in Office 97 and more than 80 other programs including AutoCAD, the latest version of CorelDraw, Peachtree Accounting, and System Architect. VBA uses most of the

Visual Basic interface but does lack some of the more advanced component and server side development features. Try them both.

Jacques Surveyer loves to chat about programming at <http://www.inforamp.net/~jbsurv>.

Visual programming resources

Beginning Visual Basic

Author: Peter Wright

Publisher: WROX

Price: \$56

Level: Beginner. It assumes the reader has no programming background.

Visual Basic 6 Unleashed

Author: Rob Thayer

Publisher: Sams

Price: \$58

Level: Intermediate. It has comprehensive coverage of features.

Visual Basic 6 Black Book

Author: Steven Holzner

Publisher: Coriolis

Price: \$70

Level: Intermediate-Advanced. Coverage gradually advances into components, Web, databases, etc.

Core Java 1.2

Authors: Cay S. Horstmann and Gary Cornell

Publisher: Prentice Hall

Price: \$57

Level: Intermediate. Includes great sample programs and tutorials.

The Complete Java Training Course

Authors: Harvey Dietel and Paul J. Dietel

Publisher: Prentice Hall

Price: \$150

Level: Beginner. With training CD, it is a solid intro to Java.

Algorithms in Java

Author: Robert Sedgewick

Publisher: Addison Wesley

Price: \$60

Level: NA. Has top-notch algorithms shown in Java code.

Java vs. Visual Basic		
Legend		
+ Good		
= Fair		
- Weak		
Characteristic	Java	Visual Basic
Ease of development	+ Very good visual-development environs	+ Very good
Number of syntax elements/routines	+ Lots of wizards and help	+ Lots of routines
Approachability	50 commands/thousands of routines	300-plus
	+ JVM/SDK free and easily available	+ Can do more
	+ Text editor and browser needed to run	+ VBA is easier to learn
	- OO design takes some extra effort	- Learning curve is steep
	- Even bigger API to master	- Huge number of features
Documentation	= Mixed quality of written material	- poorly documented
	+ Solid HTML reference documents	- electronic documents are hard to find
Availability of books, tutorials, courses etc.	= Lots of books and tutorials	= Lots of books and tutorials
	= Big in colleges and universities	+ Micro and macro
	+ Self-training easy	= mixed
Reliability and stability	- Constant and rapid change is drawback	- Constant change
Major weaknesses	- Numerous new classes/APIs	- Numerous new classes/APIs
	- Rapidly deprecating classes	- Just starting to get going
	- Huge API is bigger than Visual Basic's	- Always changing
Robustness	+ Cleaner try/exception handling & event listener models	+ Automatic memory management
	+ Runtime type, bounds checking	+ Direct access to hardware
	+ Restricted memory manipulation	- Open architecture
Security	++ Major advantage is sandbox model	- ActiveX security issues
Overall reliability	+ Reasonably reliable at start	+ Six years of experience
	- Bugs creep in early 1.1 SDK	- Still in development
	- GUI compatibility problems	- Hard to port to other platforms
	+ Major ISVs like Oracle, IBM, Sun, Sybase, etc. committed to it	+ The fact that it's Microsoft's product
Speed of development	+ Very good native and third-party tools	+ Very good
Editor	- Depends on IDE, some are weak	+ Constantly improving
Visual development	= IDEs are quite good	= IDEs are quite good
Debugging/Testing	= Client side debugging is good	++ Best in class
	= Good but mixed on server debugging	= Improving
	= Good profiling and test tools	= Some good
Project size	+ Medium scale with DBMS	+ Small scale
	+ Server or large-scale n-tier/Web	+ Medium scale
Other	+ Advantage is more secure and broad reaching components like JavaBeans, EJBs, and servlets	+ Big advantage is ease of use
	= Third-party tools, libraries, etc.	= Third-party tools, libraries, etc.
Runtime options	Gives away 10 to 30 percent speed deficit to C/C++	Gives away 10 to 30 percent speed deficit to C/C++

Cross platform	++ major advantage	- Could
Deployment options	= EXEs and byte code	= EXEs ;
	= Beans and EJB components	= Active
	= Web servlets	= Web s
Speed	+ Browser applets	+ Advan
	+ Advantage to Java on server	
	+ Byte code optimizers, trusted compiled apps just getting used	
Ace in the hole	JINI	VBA mac



[Back to Table of Contents...](#)



[Return to TCP Home Page](#)

TCP Online June 1999 Issue

TechWeb The IT Network

▶ Move

▶ Tech Contents

**Network
Computing**

THE TECHNOLOGY SOLUTION CENTER

Search

• NWC • Internet Si

Enter the APC Availability Contest.
Win a *Palm VII*.

[Getting Started](#) [Technology Guides](#) [Departments](#) [Research Center](#) [Sales](#)
Development

FEATURE

Contender or Champ? Judging Java

June 28, 1999

By Don MacVittie

Java vs. Visual Basic: Multiplatform Ease
Visual Basic, including VBScript (a specialized version of VB for Web use), competes with Java at two levels: as a Windows client and as a Windows server. If you use Microsoft Windows NT Servers as your midtier server or Microsoft Internet Information Server (IIS) as your Web server, VB remains the better option, but Java has the edge for development on other platforms.

[Print this Page](#)[E-mail this URL](#)

Microsoft has tightly bound VB to Windows, and VBScript to both IIS and Windows NT. This means VB can approximate the performance of lower-level GUI development tools such as Delphi, yet its interpreted nature makes it powerful and its gentle learning curve makes it popular. Using VB or VBScript, you can access Microsoft's COM+, DCOM (Distributed Component Object Model) and ActiveX controls directly. This offers a working set of components to compete with JavaBeans and Enterprise JavaBeans. While Enterprise JavaBeans is just now becoming reality, Microsoft's technologies are established, making the latter a safer bet for development today--at least for Microsoft-only systems.

SUMMARY
JAVA vs. VISUAL
BASIC

VB and VBScript's
strengths:

VB is still Basic, however, and Basic was never intended to be a production-development tool. For applications that will be written once and used without much maintenance, VB works very well. However, the cost of maintaining complex VB programs grows faster than the cost of maintaining similar programs

Advertisement



Interactive Buyer's Guide

ISP Services

[Article Guide](#)
[Full Chart](#)
[Customize Chart](#)
[Product Directory](#)
[Search](#)

Events

**June 4th 1:30PM
EST/10:30AM PS**

Save this date! Network Computing Publisher, Fri Nelson appearing on The Wave, airing on CNBC.

Network Design Manual

Building Voice over IP

To place an IP telephony you really need only a speaker microphone and the Internet right? Almost, the call must find its way through PBX local exchanges, gateways H.323 or SIP before anyone picks up your call.

Tools

▶ [Events](#)
▶ [IT Training](#)

- Web-server programming
- Distributed-systems programming
- Windows GUI programming

Consider Java to replace VB in:

- Web-server programming in environments that are not Microsoft-only
 - Windows GUI programming in environments that are not Microsoft-only
- Java is not prepared to compete in distributed systems today--particularly in those environments that use Microsoft exclusively. If you wish to move toward Java for distributed applications, look into Java RMI (Remote Method Invocation). When JINI reaches a level of acceptance comparable to JavaBeans' status today, it will be worthwhile to re-evaluate distributed systems.

in other languages, including Java.

If you ever have to weather the mandatory Microsoft upgrade program, you will understand why open systems are becoming so popular. If you rely on a single source for your development tool, the vendor controls your upgrade cycle: Microsoft stops supporting outdated versions, and, more important, finding compatible tools and components becomes more difficult as time passes. Microsoft is addressing the issue of letting multiple VB developers work on one project through the inclusion of source control and repository in its high-end products, but VB remains a difficult tool for collaborative development.

Many vendors produce Java development tools, so the language's upgrade path is wide open. The recent release of JDK 1.2 is a good example of how future releases might be handled. Of our informal survey of five vendors--Borland, IBM, Microsoft, Symantec and Sun--we didn't find any that planned to force users to upgrade or risk losing technical support (this has been an issue with VB in the past). One vendor, Microsoft, did not plan to implement JDK 1.2 in its product.

Of course, if you plan to use your code over the long term, you'll have to upgrade it eventually, and if you use classes specific to your IDE vendor, you

will have to follow the migration path. While the reality is that all these companies eventually will stop supporting the JDK 1.1 versions of their products, enough vendors exist to ensure an acceptable upgrade path, provided you do not use the classes specific to your vendor.

And unlike VB, Java is truly object-oriented. Ever since VB 4.0's release, Microsoft has been touting VB's object-oriented nature--but be skeptical. To be object-oriented, a language must pass four basic tests: inheritance, encapsulation, polymorphism and abstraction. Visual Basic fails two of these tests: Neither encapsulation nor polymorphism is part of VB.

Java, by comparison, is a pure object-oriented language, extended with the Internet in mind. You don't need a special version of it to write Web applications, and it's not tied to a particular hardware or software platform. Microsoft is taking three different approaches to make VB more useful for Web applications. First, it is encouraging partners to implement COM and DCOM under their Unix platforms,

- [Company Directory](#)
- [Reader Service](#)

Get NWC •

Subscribe to [Netw Computing](#) magaz and our Free wee E-Mail newsletter.

Spotlight •

Well-Connected Awards!

[Dial in to our site](#) to see t winning products posted l right from the stage tonig

7:30pm pacific.

offering access to Unix systems via VBScript; second, the company is implementing the same types of network-specific extensions added to Visual C++; and third, it's offering a slew of COM and DCOM objects to ease Web development in VBScript.

But those solutions haven't materialized. The extension of COM+ and DCOM will be challenged by JINI, Sun's Java-based approach to distributed computing, and Enterprise JavaBeans.

VB is used most frequently for client/server front ends or for standalone Windows development. These markets have attracted third-party VBXes (Visual Basic custom controls) and OCXes (OLE custom controls); VB also provides a relatively simple interface to Windows DLLs (Dynamic Link Libraries).

Because VBScript is closely bound to IIS and Windows NT, data stored in ODBC-supported databases is easy to access, and overall performance is enhanced when it is used with Microsoft database products.

But, for the same reason, you can't scale a VBScript application off a Windows PC if the load being sent through your server overburdens the PC architecture. For now, NT clustering is not a suitable replacement for a large Unix system in a high-volume situation. (See the InformationWeek article, ["A Delicate Balance,"](http://www.informationweek.com/718/18olbal.htm) at www.informationweek.com/718/18olbal.htm, for a thorough exploration of clustering technology.)

JDBC performance is based on the same key issue as ODBC--quality of vendor-written code. So with the possible exception of MS Access and MS SQL Server, database access depends on your vendor for both products. The only edge VB has over Java is in the performance of standalone Windows applications; Sun and its licensees are addressing this issue.

If your entire network consists of NT Servers and Windows clients, Java is probably a poor choice. VBScript can offer better performance and there's less to learn. But if you have or will have non-Microsoft resources that require access, Java is a better long-term bet for enterprise development.

Donald W. MacVittie is a lead technology architect for AAL, a nonprofit insurance/financial services organization. His written works include a book on object-oriented programming, and portions of ANSI/ISO standards for GIS data interchange. Send your comments on this article to him at don@nandgate.com.

[UnixWorld](#) | [Network Design Manual](#) | [Interactive Buyer's Guide](#) | [WANsites](#) | [Real-World Labs](#) | [Learn IT](#) | [Careers](#) | [Article Index](#) |

[Home](#) | [Technology Guides](#) | [Site Map](#) | [FAQ](#) | [Subscriptions](#) | [Contacts](#) | [Sales & Marketing](#) | [2000 Edit Calendar](#) |

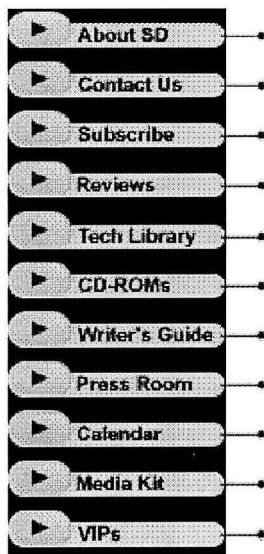
Network Computing

[Byte.com](#) | [CMPmetrics](#) | [eBusiness Expo](#) | [File Mine](#) | [InformationWeek](#) | [InternetWeek](#) | [PC
Expo](#) | [Planet IT](#) |

[TechCalendar](#) | [TechEncyclopedia](#) | [TechLearning](#) | [TechShopper](#) | [TechWeb News](#) | [Wall Street &
Technology](#) | [WebTools](#) | [Winmag.com](#) |

Try 3Com's® AirConnect™ Wireless LAN for 45 days

TechWeb is brought to you by [CMP Media, Inc.](#) Copyright © 2000 - [Privacy Statement](#)



Career & Training

DESIGNCENTER
UML Modeling
Component Central

RATE the articles

FOCUS ON
• E-DEV & SECURITY
• PROJECT MANAGEMENT

SOFTWARE
Development

Features

Inside SD Online:

[Search](#)

[Products](#)

[Features](#)

[Forum](#)

Putting Java Performance to the Test

Jan
Table

Along with Java's flexibility and power come some serious problems. You can avoid these and other performance problems through careful design and implementation of a Java application.

by *Steve Gutz*

Java has established itself as the development platform of the 1 beyond), thanks to an excellent marketing job by Sun Microsystems (and other significant industry players). Millions of developers in countries all over the world have flocked to Java as an alternative to C and C++ to gain an advantage over competitors. The reason for this might be obvious—Java helps write code better and faster. But along with flexibility and power come some potentially serious performance problems which can quickly squelch the enthusiasm that is common at the start of a new software project.

In this article, I will examine some of the common traps that Java developers fall into and for users of Java-based products. Some problems have no solution, but you can find acceptable workarounds by understanding them. Unwitting developers can actually design applications that introduce performance problems into an application—you can easily avoid these through careful design and implementation of the Java application. This article will only provide a high-altitude view of some actual Java code. I will also examine some specific performance tuning techniques that Java developers can apply to existing code. I will first look at some benchmarks for Java-based programs, which may produce some interesting results.

Hardware Considerations

To identify some of Java's performance problems, you need to look in an unexpected place—computer hardware. Java run-time performance is very sensitive to hardware, often producing poor results. By developing

simple benchmark applications and applets, which you can use the results of the target platforms, you can develop a baseline to minimum systems requirements for specific purposes.

The factors that define the minimum system requirements are compounded depending on whether the code will run as an application or an applet. Applications are straightforward since they typically do things outside the running code. Web browsers, however, add a new variable into the performance equation: the minimum requirements account for not only the applet, but also the browser. Further, the browser usually resides on a remote web server, so download time (and the overall network capabilities) is a significant performance consideration.

This minimum system requirement will not be the same for all applications. For example, an arcade-style game written in Java needs a relatively fast computer system due to the pseudo real-time speed that must occur. An application that manipulates an SQL database over a network will not need as much raw CPU horsepower because it likely spends most of its time waiting for the completion of queries generated by the application.

Having identified these two ends of the performance spectrum, a simple application, which is decidedly more "middle-of-the-road" application defined in Listing 1 is a simple "Hello World" style application. I will use to benchmark some common computer hardware. This implements a simple elapsed time component to time the creation and display of 200 Abstract Windowing Toolkit components. This program isn't a definitive measure of actual application performance, but it accurately demonstrates the time differences between various platform CPU speeds.

Listing 1. Benchmark program

```
import java.awt.*;

class Benchmark extends Frame
{
    private long    lStart = 0;
    private long    lLast = 0;
    private String  titleString;

    public Benchmark( String titleString )
    {
        this.titleString = titleString;
        logWrite( "Starting constructor" );

        // Configure the frame
        setSize( 270, 400 );
        setTitle( "Benchmark Test Program #1" );
        setLayout( new FlowLayout() );
    }
}
```



```

// Create some UI Components
for( int ctr = 0; ctr < 200; ctr++ )
{
    add( new Label( "Label " + ctr ) );
    add( new Checkbox( "Checkbox " + ctr ) );
    add( new Button( "Button " + ctr ) );
}
logWrite( "Components created" );

setVisible( true );
logWrite( "Done constructor" );
}

public void logWrite( String string )
{
    // Keep track of the current elapsed time and
    // time from start
    long lCurrent = System.currentTimeMillis();
    if( lStart == 0 )
    {
        lLast = lCurrent;
        lStart = lCurrent;
    }

    // Display the string along with a time stamp
    /*System.out.println( titleString + "\t" +
        (lCurrent-lStart)*1000
        + "\t" + (lCurrent-lLast) + "\t" +

    // Save the last time
    lLast = lCurrent;
}

public static void main( String args[] )
{
    Benchmark log = new Benchmark( "Benchmark1" );
}
}

```

Table 1. Benchmark Test Results

Platform	Benchmark Execution Time (mSec)
400MHz Pentium II, 256MB RAM, Windows NT 4.0	1,363
300MHz Pentium II, 128MB RAM, Windows NT 4.0	2,643
200MHz Pentium Pro, 64MB RAM, Windows 95	3,680

NEC 6230 Notebook, 14,140
 266MHz Pentium 64MB
 RAM,
 Windows NT 4.0

Table 2. Applet Benchmark Test Results

Platform	Benchmark Execution Time (mSec)
Microsoft Internet Explorer 4.0	1,363
Netscape Navigator 4.05	2,643
Sun Java plug-in with Netscape Navigator	1,152
Sun Java plug-in with Internet Explorer	1,143

Table 1 shows a list of tests performed using Listing 1's bench on several computers, noting the time required to create the A Windowing Toolkit components on various platforms. The times were produced by averaging ten executions of the program in using Sun's JVM 1.1.6 with just-in-time compilation enabled. loading test times may vary. You may easily notice the substantial differences in execution across systems of near-equal performance. The execution time for the 400MHz Pentium II is almost three times the 200MHz Pentium Pro, rather than what you would normally expect (approximately twice as fast). This suggests that the amount of memory and operating system can impact performance significantly. Since the benchmark program does not test Java thread execution, the Windows 95 threading model will be a factor in multi-threaded applications because the Windows 95 threading model is less efficient than Windows NT's. You should keep in mind if you are deploying complex Java applications on Windows

Notebook computers deserve some special consideration if you are targeting a notebook as a target platform. In Table 1, notice the high-end notebook computer executes the tests substantially slower than a comparable desktop and though only one notebook was tested, most suffer from the same performance problems. Though modern notebook PCs offer similar clocked CPUs as desktop systems, they have been optimized for portability rather than performance. To extend battery longevity, the internal notebook computers are usually clocked slower and can be much slower even though the CPU clock speed of a typical notebook might be stellar. Even though the CPU clock speed of a typical notebook might be stellar performance, it can spend much of its time waiting for the data and address busses. In light of the premium price paid for portability, this information is something notebook manufacturers don't want you to know; however, as Java developers, we need

that deployment on notebook PCs can be problematic from a perspective.

Applet Performance Applets pose special performance problems for developers. However, the Java code itself is not the issue, rather the browser platform on which the code is executed. On the PC platform there are currently three major browser-based Java virtual machines: Microsoft, Netscape, and Sun (with its Java plug-in), each having positive and negative characteristics. Table 2 shows the results of the previously described benchmark program as an applet on all machines. The times shown were produced by averaging ten executions of the test applet on a 300MHz Pentium II with 128MB of RAM running Windows NT 4.0. Due to CPU loading test times may vary.

There are a couple of notable performance indicators in Table 2. One obvious one of these is the sluggishness of Netscape's internal virtual machine which executes the benchmark test applet in approximately twice the time of its competitors. In the ongoing race to be the best, the Java plug-in proves that it is definitely faster than either the Netscape or Microsoft virtual machines, though only marginally so in the latter case.

Though this article is primarily focused on performance, I will not discuss the supposed ubiquity of the Java programming language, especially with respect to applets. Of the three virtual machines discussed here, only the Sun plug-in is absolutely pure. Java code executed in the native Microsoft or Netscape virtual machines may produce unexpected and possibly undesirable results, but 100% Pure Java code will always function with the Sun virtual machine. When you build products that reuse an applet, a complete investigation of the advantages and disadvantages of the Java plug-in is required. Aside from any political issues surrounding plug-ins, the Java plug-in simply works better.

Software Design Decisions What discussion of Java performance is complete without an examination of some common coding practices? The largest consideration for performance improvements should be the actual code you produce, since optimizing code will improve performance regardless of the platform or virtual machine you use.

The Golden Rule of Optimization In a typical program, 90% of the execution time will occur within 10% of the code (some people use a 10-to-20% ratio). Use a profiling tool to determine where the bulk of the execution time is occurring and concentrate on improving the performance there—ignore the 90% of the code where 10% of the time is spent. A little benefit is gained by improving the performance of code that is only used once or twice.

Since most applications involve some sort of looping condition

obvious first step to producing faster code. The following guid help you produce faster loops:

Avoid object creation within loops. Consider the following co which has a structure typical of loop code written by many dev

```
for( int iCtr = 0; iCtr < 20000; iCtr++ )
{
    String stringValue = new String( "Sample:"
    System.out.println( stringValue );
}
```

Notice that a new String object is created for each iteration of as a residual effect the object is destroyed at the end of each lo leaving fragmented memory for the Java garbage collector to c strings in Java are immutable, there is little you can do to impr implied "append" operation in this example, but you can elimin concrete object creation. The following improved code offers a performance improvement:

```
String stringValue = new String( "Sample:" );
for( int iCtr = 0; iCtr < 20000; iCtr++ )
{
    System.out.println( stringValue + iCtr );
}
```

Unroll loops for faster execution. Another way you can impro performance is by reducing the number of iterations that the lo must manage. The code to handle the loop control is definitely you can dramatically improve performance reducing the numbe iterations, especially for loops that execute a single line of code the following code:

```
for( int iCtr = 0; iCtr < 20000; iCtr++ )
{
    arrayA[iCtr] = iCtr * 12.5;
}
```

To reduce the number of iterations by a factor of two, and sign improve performance, you could write this code as:

```
for( int iCtr = 0; iCtr < 20000; iCtr += 2 )
{
    arrayA[iCtr] = iCtr * 12.5;
    arrayA[iCtr+1] = (iCtr+1) * 12.5;
}
```

Notice that in the improved code each iteration of the loop incr control by two rather than one, reducing the number of iteratio

Eliminate loops whenever possible. Another technique common optimizing compilers is known as loop elimination, which removes a loop completely if doing so will result in faster code. Assume you have the following code, which iterates a loop just three times:

```
for( int iCtr = 0; iCtr < 3; iCtr++ )
{
    array[iCtr] = iCtr;
}
```

A better way you can implement this is to unfold the loop into code.

```
array[0] = 0;
array[1] = 1;
array[2] = 2;
```

Though the code produced will be larger in some situations, it is faster because it eliminates the code required to manage the loop.

Manipulating loops is just one technique you can use to improve programs that must handle strings, which in Java are immutable, resulting in fragmented memory that becomes the garbage collector's responsibility. The garbage collector runs on a separate thread in the application to ensure that memory is released and returned to the heap. A simple way you can improve performance is by preventing the garbage collector from having any reason to start. You can do this by minimizing the use of Java strings, opting instead for `StringBuffer`. The following code is a factored-down version of the type of string manipulation that occurs in everyday Java code:

```
String string1 = new String( "" );
for( int iCtr = 0; iCtr < 20000; iCtr++ )
    string1 = string1 + "A";
```

On a given test system, this code required an average of 7.93 seconds to complete. The `StringBuffer` class, though often ignored by Java programmers, offers vastly better performance than a string. Rewriting this same code with `StringBuffer` yields:

```
StringBuffer string2 = new StringBuffer( "" );
for( int iCtr = 0; iCtr < 20000; iCtr++ )
    string2.append( "A" );
string1 = string2.toString();
```

The optimized example requires one more line of code, so it will have a slightly larger footprint; however, on the same test system, this requires just 0.15 seconds to execute— that's almost 50 times faster.

There are, of course, many more techniques that you can employ with Java code, most involving well-understood ideas that have been around for years. You can use any search engine to find several sources for information on the Internet—search for "Java optimization" and you will be rewarded with some excellent references.

Profiling Tools Some of the optimization concepts I described in the previous section assume that you can detect bottlenecks within your code. Fortunately, there are a number of excellent tools to help you do this. They not only point out places where CPU might be wasted, but also more obscure performance factors such as excessive object creation. There are many more available than I can cover here, so instead I will only discuss the most often used.

The easiest profiling tool to obtain is the one included with the Java Development Kit. However, this profiler offers only limited usefulness. It is clearly aimed at advanced developers. To use the built-in profiler, run the application as follows:

```
java -prof:out.txt Benchmark
```

Here, "out.txt" is the output file where profile data will be written. "Benchmark" is the name of the class that initializes and starts the program. A fragment of the profile results written to the output file is shown in Listing 2; and plainly, this format is not easy to read. Though it does contain a great deal of information you need to improve the performance of a program, it is likely you will spend a great deal of time sifting through the output.

```
count callee caller time
/*handles_used: 1070, handles_free: 26214, heap-used: 732272*/
sig count bytes indx
[C 149 34616 5
[B 7 27520 8
*** tab[963] p=1d48dc8 cb=f70140 cnt=646 ac=1 al=1
    Ljava/lang/String; 646 7752
    [Ljava/lang/String; 1 4
*** tab[925] p=1d48b68 cb=f700a8 cnt=2 ac=1 al=4
    Ljava/lang/ThreadGroup; 2 80
    [Ljava/lang/ThreadGroup; 1 16
*** tab[883] p=1d488c8 cb=f70000 cnt=3 ac=2 al=8
    Ljava/lang/Thread; 3 132
    [Ljava/lang/Thread; 2 32
.
.
.
```

Listing 2. Profile out Sun Microsystems also provides a profiler called JProfiler. It is both fast and easy to use. This profiler lets you see the execution time of each method at a millisecond resolution, and this value

or exclude the time used by other code that the method may call during its lifetime.

Figure 1 shows the profiler output produced by a simple test program that includes two methods. Notice that in the top window, the main method is highlighted. The middle window shows the times used by any main() calls. The lower window would show the times for calls if there were any.

JProbe from KL Group is the final profiling tool I will examine. This product, though a bit sluggish, can tell you everything you ever want to know about a program's performance. Unlike the Java Workshop, which is limited to displaying CPU time usage and frequency, JProbe shows execution time, execution frequency, and object creation—which lets you generate a general graphical profile of the entire program down to a single method if desired.

Figure 2 shows the JProbe method detail screen for a method named copySlow(). The screen is divided into three parts: parent method information, method information, and child information. The parent section lists methods that call copySlow(), while the section describing child methods lists data about methods that copySlow() calls. In each case, the CPU time and calling frequency are listed. Within each of the three sections is a diagram graphically illustrating the meaning of the values in the associated data tables.

Unlike applications written in C or C++, most Java code is interpreted. Performance that C++ programmers take for granted must be carefully designed into Java code. This is the only way to ensure the results users expect. All Java programmers need to arm themselves with a notch profiling tool. I have discussed three excellent commercial tools here and also ways to profile code without incurring any costs. The Internet also offers several capable tools for optimizing code inexpensively. Go to any search engine on the World Wide Web and type "Java profiler" and you should find some fine tools to make your

Handling Performance Issues Within the confines of a single article, it is impossible to present an ultimate solution to all the performance problems associated with deploying Java applications. Instead, I have focused on a few key areas where a little awareness can solve many problems that customers ever see with your products. Some problems, such as the mismatch between the target computer hardware and the actual hardware, are often overlooked and unavoidable. Even on notebook computers simply run slower. But if you enter into the market prepared to handle performance issues, you won't be surprised by unexpected schedule overruns while your team tries in vain to solve the problem.

Regardless of the target platform, you should optimize all Java have examined several of the common coding techniques that employ to make code execute faster. Additionally, I have discussed the tools that you can use to identify where bottlenecks occur in Every Java developer needs at least one of these tools at his or for daily use.

The Java language and virtual machine are still infants, and though growing up quickly, they still pose some potentially disastrous headaches for all of us. In the future, JavaSoft promises a significant virtual machine containing a new technology called HotSpot, but we all have to live within the limits of what we currently know

Further Reading:

- Steven Gutz, *Up to Speed with Swing*, Manning Publications, 1998 ISBN1-884777-64-3
- David Flanagan, *Java in a Nutshell 2nd Edition*, 1997, ISBN 1-56592-262-x
- Douglas Bell, *Make Java Fast: Optimize!*, JavaWorld IDG Publications, 1997

Sun Microsystems, not oblivious to the performance concerns by most Java developers, is currently working on a new technology called HotSpot, which promises to deliver near C++ performance Java programs. Scheduled to ship soon after Java Development the HotSpot virtual machine promises to include some impressive performance improvements.

To the applause of all Java developers, the current garbage collector will be rewritten to include a much smarter collector mechanism. Though most Java developers believe that memory leaks do not in Java, the garbage collector in the current release can sometimes "fool" into thinking that data is still in use, causing fragmented memory and ever-increasing memory requirements for a running application. With HotSpot, garbage collector problems will be distant memory.

Another key feature of HotSpot is a smart optimizer mechanism performs "adaptive inlining" of hot spots or performance-critical. This lets the virtual machine examine running code dynamically inline slow code to make it faster. Other HotSpot features that to boost performance dramatically include an ultra-fast synchronization mechanism and direct object pointers—the latter eliminates the superfluous memory references to access objects.

HotSpot certainly will offer a performance boost over the current virtual machine, but will it be as dramatic as Sun is suggesting time will tell. Assuming that the Sun engineers can meet their Java will take on a completely new role in the industry and will be impossible for development houses to ignore.



*Copyright © 2000 Software Development magazine. All rights reserved.
If you have any comments or questions about this site, please direct them to the
[Webmaster](#). View our [contact page](#) for advertising or other business-related information.
Please view our [sponsors page](#). Read our [Privacy Policy](#).*

Timing Trials, or, the Trials of Timing: Experiments with Scripting and User- Interface Languages

Brian W. Kernighan
Bell Laboratories
Murray Hill, NJ 07974
bwk@bell-labs.com

Christopher J. Van Wyk
Department of Mathematics and Computer Science
Drew University
Madison, NJ 07940
cvanwyk@drew.edu

ABSTRACT

This paper describes some basic experiments to see how fast various popular scripting and user-interface languages run on a spectrum of representative tasks. We found enormous variation in performance, depending on many factors, some uncontrollable and even unknowable. There seems to be little hope of predicting performance in other than a most general way; if there is a single clear conclusion, it is that no benchmark result should ever be taken at face value.

A few general principles hold:

- * Compiled code usually runs faster than interpreted code: the more a program has been ``compiled" before it is executed, the faster it will run.
- * Memory-related issues and the effects of memory hierarchies are pervasive: how memory is managed, from hardware caches to garbage collection, can change runtimes dramatically. Yet users have no direct control over most aspects of memory management.
- * The timing services provided by programs and operating systems are woefully inadequate. It is difficult to measure runtimes reliably and repeatably even for small, purely computational kernels, and it becomes significantly harder when a program does much I/O or graphics.

Although each language shines in some situations, there are visible and sometimes surprising deficiencies even in what should be mainstream

applications. We encountered bugs, size limitations, maladroitness features, and total mysteries for every language.

BenchmarkingPerformance evaluationScripting languages

INTRODUCTION

This paper describes experiments to compare the performance of scripting languages (like Awk, Perl, and Tcl) and interface-building languages (like Tck/Tk, Java, and Visual Basic) on a set of representative computational tasks. We found this challenging, with more difficulties and fewer clear-cut results than we had expected.

Since scripting and interface-building languages are usually interpreted, using them sacrifices some speed in return for programming convenience. This "wasted" CPU time doesn't matter for short, single-use programs, or for programs that run in only a few seconds, nor does it matter if the scripting language is mainly used as "glue" to control efficient large-scale operations. But small programs for small data sets can evolve into big programs that run on big data sets; for instance, Awk and Perl have been used to write programs of thousands of lines that process megabyte data sets. And sometimes the glue itself evolves into a major component of an application. By quantifying how fast different languages run, performance comparisons could suggest which languages would be appropriate for different parts of an application, and what the penalty might be for choosing convenience over efficiency.

At first blush, comparing runtimes seems straightforward. Choose a comprehensive set of representative tasks, write equivalent programs in the different languages, measure the runtimes on several systems, then present them all in a table. This approach is an obvious generalization of test programs that create a cost model for primitive operations in a single language like C [Bentley 91] and benchmarks that compare implementations of a single language like Java [Hardwick 97, Caffeine 97].

The seemingly easy job of comparison, however, has proved to be much more than a routine exercise. We encountered problems, puzzles, and surprises at every step. We hope that this journal of our expedition will encourage readers to view published comparisons with caution, and to design and perform their own experiments carefully.

When reporting results of our experiments, we often say that a program in one language ran some amount faster or slower than an equivalent program in another language. All such results represent a snapshot, in the summer of 1997, of the performance of language processors that were readily available to us. They provide no warrant for drawing conclusions about the relative performance of languages on other machines or using other versions of software.

Even though we cannot state many firm conclusions, there are some themes that recur often in the experiments described below.

* Compilation wins over interpretation, and interpretation from an intermediate

representation wins over repeated interpretation of the original program text. The exact speedup is unpredictable, however, and compilation may even slow the program down instead. For the most part, we have tested only interpreters, since they are always available, while compilers are often experimental, unsupported, inaccessible, or an extra-cost option. The limited results presented in Section 7 indicate that compilation is generally, but not always, of modest benefit.

* Memory matters. Of course the absolute amount available on a machine is important, especially once paging (or thrashing) begins. But buffers and caches (at several different levels of hardware and software) and the implementation of memory allocation and garbage collection can also have dramatic effects on performance. These factors often lie beyond user control.

* Accurate timing is very hard. Even simple timing loops can exhibit erratic runtimes for no obvious reason. Once a program interacts with its environment by doing significant I/O or graphics, the variation in runtime measurements becomes even larger. These well known problems are exacerbated by systems with unreliable internal "timing" routines. We also have not found a satisfactory way to charge startup costs to all systems, so head-to-head comparisons between languages are sometimes difficult.

1 Methodology

We started to construct a table with three dimensions: task, programming language, and machine. Eventually we added the size of the problem solved by the program as a fourth dimension, and we changed the presentation from tables to graphs. Varying the problem size helped us to detect unusual runtime effects, while a graphical presentation highlights patterns and trends in runtime instead of individual performance scores.

We present measurements for each task separately. Unlike some benchmarking studies, we do not combine the results of several different experiments into a single number. One problem with this common practice is that one can choose different weighting factors to create almost any desired outcome [Becker 87].

Choice of Tasks

It is hard to devise small yet representative tasks that do not give too big an edge to one language or unduly penalize another. Ideally, each task will test only one or two language features, and will require short, simple, programs that are easy and natural to express in any language. For the comparison to be sensible, the programs also must be expressed in a "colloquial" fashion in each language. While we do not claim that any of the tasks we present is definitive, we believe that each is reasonably fair.

We have organized the presentation by type of task: basic language features (Section 3); arrays and strings (Section 4); input/output (Section 5); basic GUI operations (Section 6). Section 7 describes experiments with compilation. Each section includes further explanations of the tasks. The programs and test data are available at www.cs.bell-labs.com/~bwk.

Choice of Languages

We report timings for the eight languages summarized in Table 1.

...

	C	Awk		Perl	Tcl	Java
arrays	indexed	assoc		indexed and assoc	assoc	indexed and assoc
graphics	libraries	none		Tk embed- ding(1)	yes (Tk)	standard library
linux version	native compiler	Bell 1996	Labs	Perl 5.003 Perl 5.004	Tcl 7.6 Tk 4.2	
Solaris version	native compiler	Bell 1996	Labs	Perl 5.003 Perl 5.004	Tcl 7.6 Tk 4.2	Sun JDK 1.1
Windows 95 version	Microsoft Visual C++ 4.1	Bell 1996	Labs	Perl 5.003	Tcl 7.6 Tk 4.2	Sun JDK 1.1

(1) Embeddings of Tk for Perl and Scheme appear to be experimental and unsupported; we did not try them.

Table 1: Language Characteristics

C is included as a baseline: its performance suggests the size of the runtime penalty for using an interpreted language. C is the only one of the eight languages that does not have built-in string operations and automatic garbage collection.

Awk and **Perl** are similar in style, although Perl has many more features. Strings and numbers are the only data types; associative arrays are the main data structure (the only one in Awk); both are normally interpreted. One of the authors admits to a paternal interest in Awk.

Tcl is an interpreter reminiscent of the Unix C shell; besides strings, it provides associative arrays. **Tk** is a user interface toolkit normally packaged to be run from Tcl scripts through a "windowing shell" called Wish. Tcl 7.6 is a pure interpreter; the more recent Tcl 8.0 includes an on-the-fly bytecode compiler.

Java is derived from C and C++. Java comes with a standard user interface library, the Abstract Window Toolkit or AWT. Java programs can be run standalone or as "applets" invoked from within a Web browser; in either case, the intermediate representation may be interpreted, or it may be compiled just-in-time for the machine

on which the program is being run.

Visual Basic is an interpreter loosely in the same class as Java. It provides an especially easy environment for creating graphical interfaces. Visual Basic is available only for Windows 95 and NT.

Limbo is a language in the same general class as Java and attacks much the same problem domain with somewhat the same facilities; it too is normally interpreted. Limbo is part of Inferno, a virtual operating system that runs on Windows 95 and NT and many flavors of Unix. Limbo has its own *ab initio* implementation of Tk for writing graphical interfaces. The authors admit to personal and corporate interests in Limbo.

Scheme is a functional language extended by assignment and I/O. We tested the MIT Scheme versions shown in Table 1, both of which are pure interpreters that date from 1993. We tried several Scheme compilers that are accessible on the Web, including Scheme48, Stalin, VSCM, and Edscheme, but each failed to run one of our test programs or was not readily available for all of our computing platforms. (See [Clinger 97] for more Scheme comparisons.)

Given more time and energy, one might test other scripting languages like Unix shells, Python and REXX, functional languages like SML, and interface-building languages like Delphi.

Choice of Machines

We ran our tests on three machines:

- * 100 MHz MIPS R4000 SGI Indy, Irix 5.3
- * 50 MHz Sun Sparcstation-10, Solaris 2.5 (SunOS 5.5)
- * 100 MHz Pentium (32Mb Micron Millennia, Diamond Stealth64 2001 video), Windows 95. This PC appears to be identical to the one used for tests in [Booth 97].

Although elderly, these machines are very similar in speed. For us, they offered the further advantage that we could ensure that no one else was using them during our tests, which removed one source of variability. We could run every test on each system, except for Visual Basic, which we ran only on the PC, and Java, for which we did not have an implementation on Irix. Except where noted, we used the same implementations across systems.

Gathering Timing Statistics

The difficulty of measuring program runtimes reliably and reproducibly is well known. We encountered the following obstacles, among others:

- Runtime measurements become more erratic as a time-shared machine becomes more heavily used. Even on single-user machines, system processes can spring into activity in the middle of a test, with unpredictable effects on runtimes. Process timings on PC's are notoriously unreliable, especially for short runs, but they are no better on Unix systems. Wall clock times can be much greater than the sum of system and user times, even on apparently quiescent systems.
- I/O times can be badly misleading when the computer is far from the file, as with networked file systems. Times for graphics operations can be badly misleading when the display is remote from the computer, as with X terminals.
- Some programs run faster the more often they are run, as if they get better with practice. Sometimes this is due to caching; for example, a program that reads a big file may run faster on the second and subsequent runs because the program and the file are already in memory.
- A few programs take longer to run the more often they are run. We suspect that the increasing runtime is due to memory management problems.

Times for C, Awk, Perl, and Tcl are the sum of ``user time" and ``system time" reported by the `time` command on Unix, or the total time reported by the MKS Toolkit `time` command on Windows 95. Thus, these include the time to invoke the language processor (for Awk, Perl, and Tcl) or to load and start the program (for C) as well as to read, process and execute the programs.

Times for Java, Visual Basic, and Limbo are computed by the internal program timer, such as `Date` in Java and `Timer` in VB. These exclude all startup times, which can amount to several seconds. For example, the Java interpreter exhibits a noticeable startup delay, presumably from unpacking the standard class files.

Timing Scheme programs posed a challenge. The internal timing function in MIT Scheme, `runtime`, reports values that are only loosely correlated with wall-clock time. (The `runtime` function appears in the standard Scheme text [Abelson 96], although it is not mentioned in the Scheme reference [Scheme 91].) On Irix and Windows, `runtime` reports values that are too low by 10 to 25 percent. On Solaris, `runtime` reports times that are too high by 30 to 65 percent. Consequently, we used a stopwatch to record Scheme times. The Scheme times also exclude the interpreter startup time.

To gather the timings, we ran each test several times, usually with a loop like this:

```
for i in 1 2 3 4 5 6 7 8 9 10
do time commandline
done
```

We discarded the first time to compensate (partially) for caching. If the runtime was more than a minute or so, we normally ran the test only 2 or 3 times: when one version of a program runs in a few seconds, but another takes minutes, there is no reason to fix the latter time with pinpoint accuracy. For times measured in seconds, we normally ran between 5 and 10 trials. Notes accompanying the graphs explain unusual situations,

such as large variability in the runtimes.

2 Tests of Basic Features

The first tests exercise basic language features like arithmetic, loops and function calls.

Loops and arithmetic

First we test the implementation overhead of loop mechanisms by counting to n . Using large values of n helps to reduce noise in the timing measurements. Verifying that runtime is linear in the value of n helps to detect ambitious optimizers (see below).

Programs in most of the languages look similar. For example, here is the C version:

```
int i, n = atoi(argv[1]), sum = 0;
for (i = 0; i < n; i++)
    sum++;
```

and the Tcl version:

```
set sum 0
set n [lindex $argv 0]
for {set i 0} {$i < $n} {incr i} {
    incr sum
}
```

We timed the following Scheme version of the program, which is conventional in its use of tail-recursion to implement the loop, but unusual in using `set!` to modify a global variable repeatedly:

```
(define sum 0)
(define (tail-rec-aux i n)
  (if (< i n)
      (begin (set! sum (+ sum 1)) (tail-rec-aux (+ i 1)
          sum))
      sum))
(define (tail-rec-loop n)
  (tail-rec-aux 0 n))
```

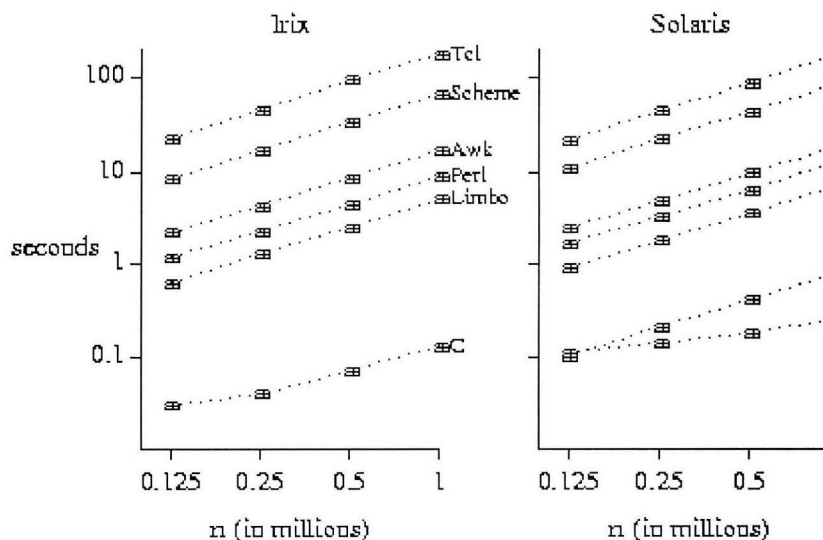
An alternative version that passes three arguments to `tail-rec-aux` and only uses `set!` at the end of the recursion runs about five percent faster.

For unreconstructed imperative-style programmers, Scheme defines an iteration construct (`do`), with which the loop above could be written as follows:

```
(define sum 0)
(define (do-loop n)
  (do ((i 0 (+ i 1)))
      ((>= i n) sum)
      (set! sum (+ sum 1)))))
```

With MIT Scheme, the runtime for this version is the same as for the tail-recursive

version. With Edscheme, a commercial PC implementation, the `do` version ran twice as fast as the tail-recursive form. We could not do more experiments with Edscheme since our evaluation copy expired before we finished our tests.



Basic loop test

This graph exhibits several features that are common to all of the graphs in this paper. The amount of computational work appears on the horizontal axis and runtime appears on the vertical axis. Both axes are plotted on a logarithmic scale, which allows us to display a wide range of data values. It would be convenient to choose input sizes that give reasonable runtimes, ideally around 10 seconds, but this ideal cannot be achieved when runtimes for different languages are four orders of magnitude apart.

We designed tests whose runtime should grow linearly with the size of the problem: **runtime** = **m**×**size** + **b**. Thus, if we choose **size** to be large enough to justify ignoring the fixed overhead (**b**), the log-log plot should show a straight line of unit slope. Exceptions indicate anomalous behavior that deserves further attention.

For example, on Windows 95 the line connecting C runtimes appears absolutely horizontal. In fact, runtime continues to be constant when $n = 10^7$ and $n = 10^8$. This happens because the optimizer eliminates the entire loop, replacing it by `sum = n`. When optimization is disabled, times on Windows 95 grow **very** slowly with n , from 0.25 seconds at $n = 125,000$ to 0.33 seconds at $n = 10^6$.

The graph for Windows 95 also shows clearly the advantage that Java and Visual Basic enjoy because our timings do not charge them for startup.

The pattern in these graphs reappears in many test results. Compiled native code (C) runs fastest; next fastest are interpreted byte codes (Java, Limbo, Visual Basic); next come interpreters that construct and execute an internal representation like an abstract syntax tree (Awk, Perl); slowest of all are interpreters that repeatedly scan the original source (Scheme, Tcl). Between each consecutive pair of stages there is a factor of 5-10 difference in runtime, with a total range that exceeds three orders of magnitude.

Function Calls

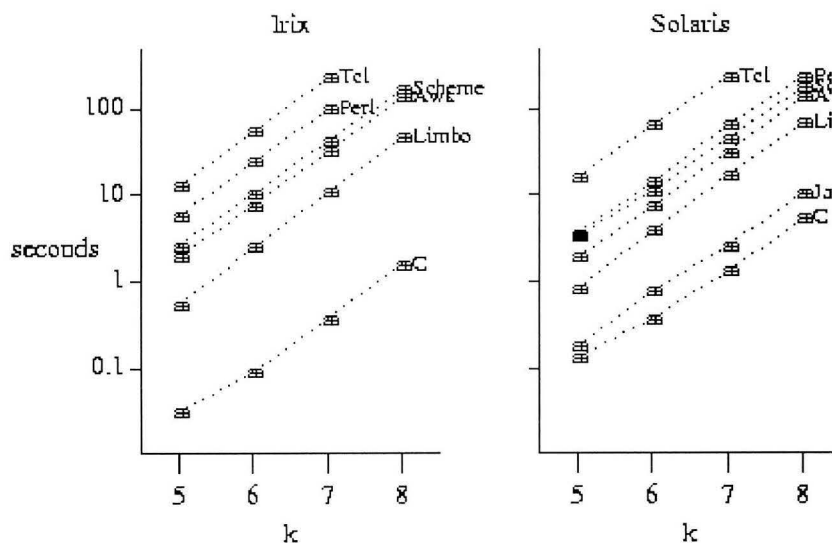
The next program evaluates Ackermann's function. Computing $\text{ack}(3,k) = 2^{(k+3)} - 3$ requires at least $4^{(k+1)}$ function calls, and reaches a recursive depth of $2^{(k+3)} - 1$, so this test gives the function call mechanism a thorough workout. The code looks similar in most languages; here it is in C and Java:

```
int ack(int m, int n) {
    if (m == 0)
        return n+1;
    else if (n == 0)
        return ack(m-1, 1);
    else
        return ack(m-1, ack(m, n-1));
}
```

and in Scheme:

```
(define (ack m n)
  (cond ((= m 0) (+ n 1))
        ((= n 0) (ack (- m 1) 1))
        (else (ack (- m 1) (ack m (- n 1))))))
```

...



Ackermann's function test: `ack(3,k)`

On Windows 95, we killed the Perl program computing `ack(3,8)` after an hour of no apparent progress. On Irix, Perl 5.004 aborted with no message on `ack(3,8)`; Perl 5.001 took 350 seconds to compute the correct answer.

By default, Tcl sets the `maxNestingDepth` of the runtime stack to 1000. For our tests, we increased this limit to 5000 and recompiled Tcl on Irix and Solaris (but not on Windows 95). Even with this upper bound, however, Tcl overflowed when computing `ack(3,8)`, because control structures and expression evaluations, as well as procedure calls, require a recursive invocation of the Tcl interpreter.

For the largest problem size ($k=8$), the order of languages by increasing runtime remains the same across all three systems. (The basic loop test results also exhibited cross-system consistency, but with the languages in a different order.) Here, Scheme and Awk ran faster than Perl, and Limbo ran faster than Visual Basic. Such total consistency across systems is rare.

3 Arrays and Strings

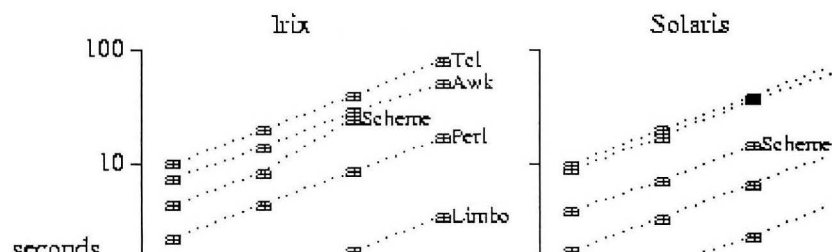
All of these languages provide richer arrays and strings than those in C, including perhaps associative arrays, dynamic arrays, storage management for strings, and garbage collection. This section investigates some of their properties.

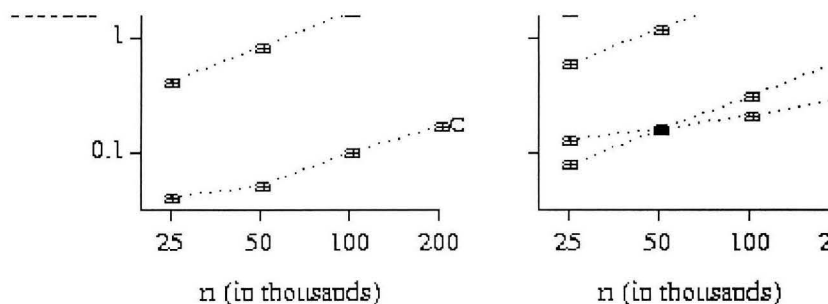
Indexed Arrays

The first program uses arrays as if they were indexed. It sets n elements of an array to integer values, then copies the array to another array beginning at index $n-1$. Here is the code in C, Awk, Java, and Limbo, for example:

```
for (i = 0; i < n; i++)
    x[i] = i;
for (j = n-1; j >= 0; j--)
    y[j] = x[j];
```

...





Indexed array test

The Scheme test failed to terminate on all three systems when the array was too long.

This test turned out to be tougher than we expected. Languages that offer only associative arrays (like Awk and Tcl) typically use hash tables to implement the simple ``indexing'' needed for this example. This accounts for their large runtimes. Indeed, rewriting the Perl program to use associative arrays instead of indexed arrays triples its runtime on Unix, and quintuples it on Windows 95.

On the PC (with 32Mb of memory) Awk and Tcl thrash badly when $n = 200,000$, with runtimes about of about 450s (Awk) and 170s (Tcl).

String manipulation

The next test exercises several string operations: computing string lengths, concatenating strings, and extracting substrings. The program constructs the strings to avoid incurring I/O time by reading them from a file. Here is the Awk version:

```
for (j = 0; j < 10; j++) {
    s = "abcdef"
    while (length(s) <= n) {
        s = "123" s "456" s "789"
        s = substr(s, length(s)/2) substr(s, 1, length(s))
    }
}
```

This program computes the length of s twice in the second assignment in the while-loop. A version that stores the length in a temporary variable only runs about five percent faster, so we tested the straightforward version shown above.

The C program, on the other hand, keeps track of the length of the growing string:

```
int j, len, n = atoi(argv[1]);
char *s = NULL, *p;

for (j = 0; j < 10; j++) {
```

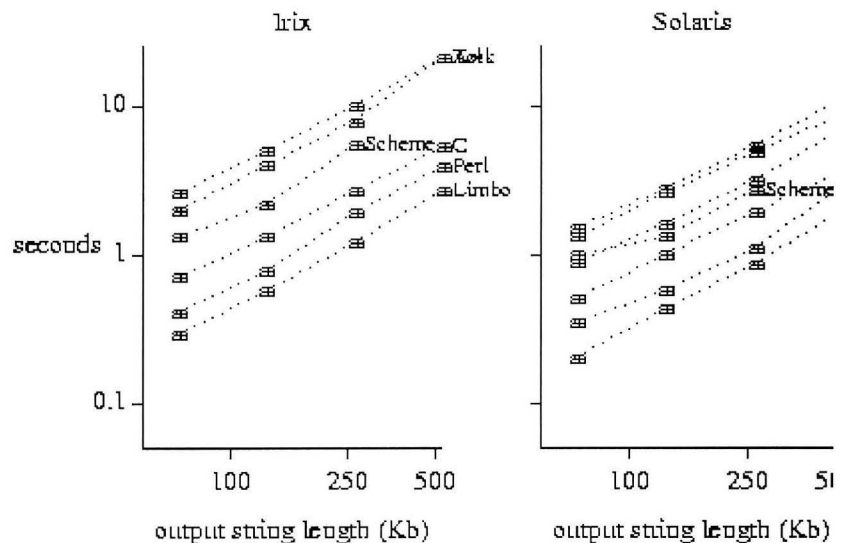
```

free(s);
s = strdup("abcdef");
while ((len = strlen(s)) <= n) {
    p = (char *) malloc(2 * len + 10);
    sprintf(p, "123%s456%s789", s, s);
    free(s);
    s = p;
    len = strlen(s);
    p = (char *) malloc(len + 2);
    strcpy(p, s + len/2);
    strncat(p + len/2, s, len/2+1);
    free(s);
    s = p;
}
}

```

On the Unix machines, a version that does not store `len` runs about 40% slower than the program shown above. But on Windows 95, the version that does not use `len` is only about five percent slower.

...



String test

Scheme runtimes for this test are unusually fast, but Scheme failed to terminate when string `s` became too long.

The C runtimes are very sensitive to the details of the program: they get faster the fewer times the string is traversed from beginning to end (as in `strlen`, `strcpy`, and `strcat`). The graphs suggest that Awk, Tcl, and Java represent strings as null-terminated byte sequences, but Perl and Limbo include length information in their string data structures.

This test also exercises the storage allocator. Evidently the C library version, on which both Awk and Tcl rely, is slow. Perl may benefit from carrying its own version of `malloc`.

Associative Data Structures

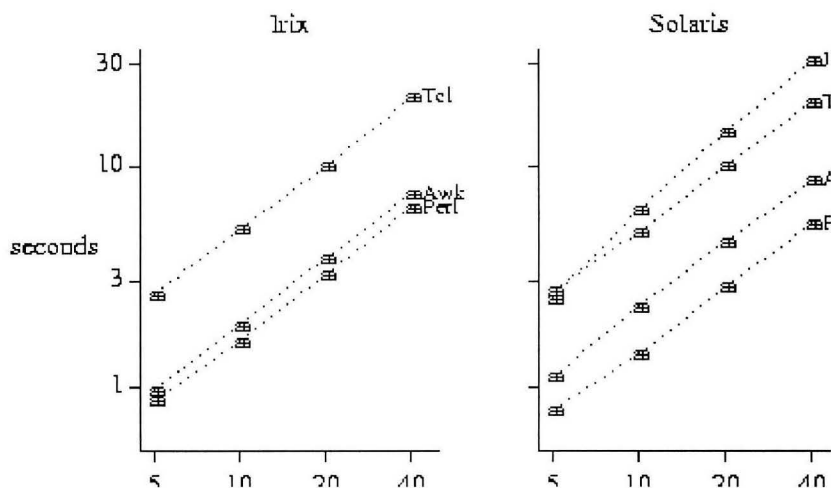
Several of these languages offer a built-in facility (variously called associative arrays, hash tables or hashes) that can be used to associate values with keys so that lookups can be performed in constant time. Our test program synthesizes key values from numbers, to avoid performing any I/O. In order to exercise both lookups that succeed and lookups that fail, the program stores keys created from the numbers 1 to n expressed as hexadecimal strings, but attempts to retrieve keys using the numbers 1 to n expressed as decimal strings. The program in Perl is

```
for ($i = 1; $i <= $n; $i++) {
    $X{sprintf('%x', $i)} = $i;
}
for ($i = $n; $i > 0; $i--) {
    if (defined $X{$i}) {
        $c++;
    }
}
```

and in Tcl

```
for {set i 1} {$i <= $n} {incr i} {
    set x([format "%x" $i]) $i
}
set c 0
for {set i $n} {$i > 0} {incr i -1} {
    if {[info exists x($i)]} {
        incr c
    }
}
```

...

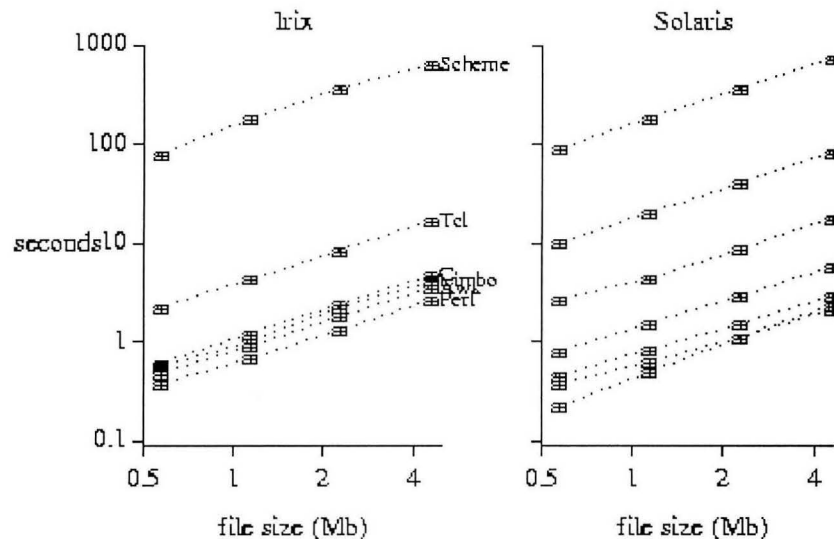


character at a time. Here is the C version, which uses `stdio`:

```
main(int argc, char *argv[]) {
    int c;
    FILE *fp = stdin;

    if (argc > 1)
        fp = fopen(argv[1], "r");
    while ((c = getc(fp)) != EOF)
        putc(c, stdout);
}

...
```



File copy test

The graphs show that programs that work at the level of lines usually run faster than programs that work at the level of characters. Of course, the line-at-a-time programs might be imposing restrictions on line length that are not revealed by using the King James Bible as input. (The longest verse is Esther 8:9, with 92 words and 529 characters.)

Thus, the graphs confirm the conventional wisdom that the choice of "chunk" size on which a program operates is important to its performance. Indeed, we can double the speed of the Perl program on Irix by changing it to read its input in one chunk. On the other hand, changing the Tcl program on Irix to read its input all at once makes it run about six times slower.

Our tests also show the importance of I/O buffering.

Unix systems implicitly buffer I/O that has been redirected from or to files, but this has

not been carried over to Windows 95. This meant that our original Tcl program buffered its output one line at a time, and ran nearly 20 times slower than the version reported above. We modified it by explicitly requesting `full` buffering on `stdout`.

The Java times are for a program that uses the `BufferedInputStream` class. Using unbuffered classes for I/O increases the runtime by at least 50%. Even with buffering, Java programs incur large runtime because the I/O methods are interpreted, not compiled. The JDK1.1 version of Java also imposes some overhead associated with synchronizing multiple threads [Gosling 97].

The enormous Scheme runtimes appear to be caused by the lack of facilities for buffered input and output. We wrote a C program that reads and writes one byte at a time, and it took even longer to run than the Scheme program, so the Scheme implementation may be buffering deep inside.

Counting words

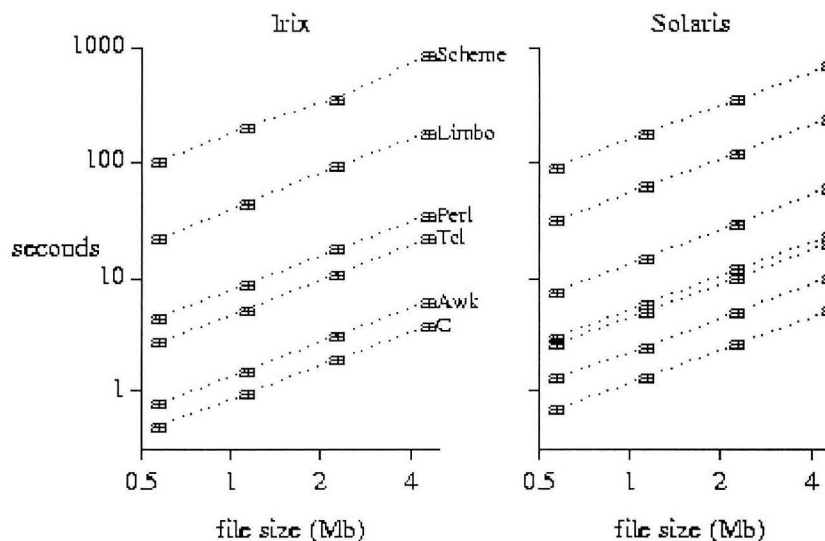
The second test is to count lines, words, and characters in the input, as in the Unix `wc` program. This tests input speed and the ability to parse lines into smaller units. There are at least two basic approaches. In Awk, Perl, and Tcl, the natural program reads each line and splits it into "fields," each of which corresponds to a word. Here is `wc` in Awk:

```
    { nl++; nw += NF; nc += length($0) + 1 }
END { print nl, nw, nc }
```

In C, Java, Visual Basic, Limbo, and Scheme, the natural program reads the input one character at a time and uses a state machine to count transitions into and out of words. Here is the Java version:

```
int nl = 0, nw = 0, nc = 0;
int b;
boolean inword = false;
while ((b = System.in.read()) > -1) {
    ++nc;
    if (b == '\n')
        ++nl;
    if (Character.isSpace((char)b))
        inword = false;
    else if (inword == false) {
        ++nw;
        inword = true;
    }
}
```

We timed a Scheme version of this program that `set!`s three global variables to do this task. A tail-recursive version that passes these variables as parameters, perhaps more typical Scheme style, ran about five percent slower.



Word count test

The graphs confirm the importance of "chunk size" mentioned above for *cat*. Notice how much worse Visual Basic and Limbo fare when they operate on single characters instead of lines.

Since *wc* does more computation on the input than does *cat*, we expected it to have longer runtimes. About half the time, it does. But there are also places where *wc* is faster than *cat*, including Tcl on Windows 95, and Java on both Solaris and Windows 95; perhaps this is because the output of *wc* is only one line, much smaller than the output from *cat*.

Reversing a file

The next test is to read the entire document into an array, then print out the lines in reverse order. This test checks the ability to build and access a very large array. The Awk version is

```
{ x[NR] = $0 }
END { for (i = NR; i >= 1; i--)
      print x[i]
}
```

The Perl version reads the entire file into an array with a single statement:

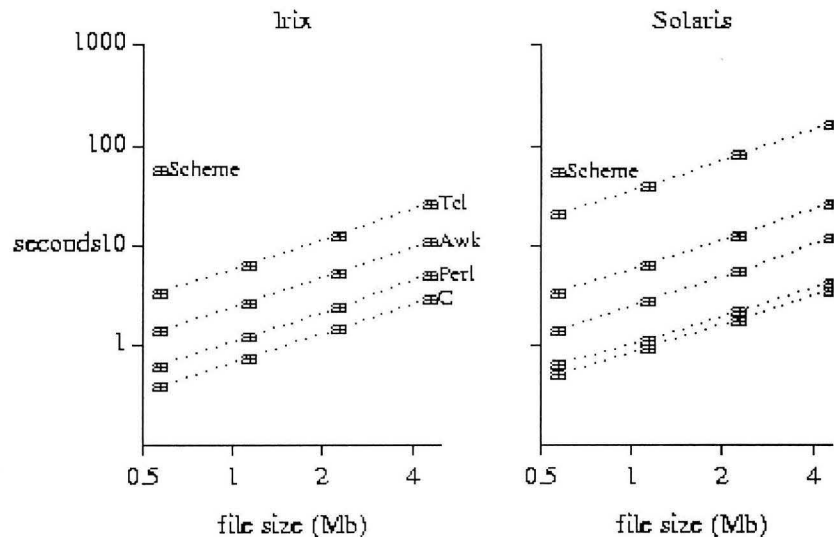
```
@a = <>;
for ($i = $#a; $i >= 0; $i--) {
    print $a[$i];
}
```

```
}

```

Unlike the Unix command *tail -r*, our C implementation does not seek from the end of the file, but reads the input into an array.

```
...
```



File reversal test

The Scheme version terminated with an "out of memory" message on each system while trying to reverse the first quarter of the bible.

The order of languages by runtime is the same across all three systems. The word-counting task also exhibited cross-system consistency, but the languages appeared in a different order.

sum

The next test exercises input, numeric conversion, and arithmetic. The programs read 100,000 floating point numbers like this

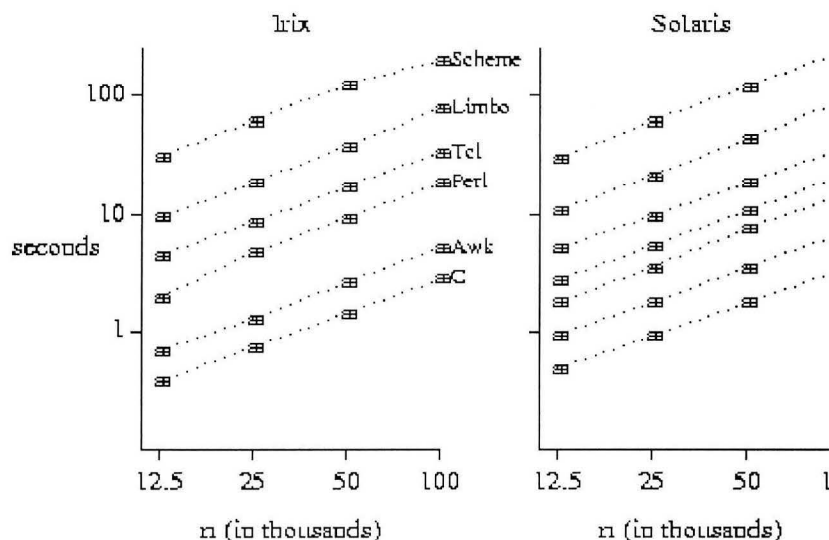
```
513.871
-175.726
308.634
...
```

from a file and compute their sum. This program is easy in Awk:

```
{ s += $1 }
END { print s }
```

and not much harder in other languages.

...



Sum test

The order of languages by runtime is comparable to the order for word counting, except for Tcl (which got slower) and Visual Basic (which got faster). In fact, our original Tcl program ran faster, but computed a different sum than any of the other programs; the timings shown are for a Tcl program that explicitly sets `tcl_precision` to 17.

5 Graphical User Interface Tests

There are two major components of user interfaces where speed matters most: manipulating text in a "text widget," and drawing graphical objects like lines, rectangles and circles. Of course, operations like raising dialog boxes or pulling down menu items should happen quickly, but their runtime is of little significance to overall performance.

Text Tests

The text test appends n short lines to a text box, one at a time, and causes the last line to be displayed each time. The Tcl loop looks like this:

```
for {set i 1} {$i <= $n} {incr i} {
    .t insert end "$i\n"
    .t see end
    update
}
```

```
}
```

The Limbo version is a direct transliteration:

```
for (i := 1; i <= n; i++) {
    tk->cmd(t, ".t insert end " + string i + "\n");
    tk->cmd(t, ".t see end");
    tk->cmd(t, "update");
}
```

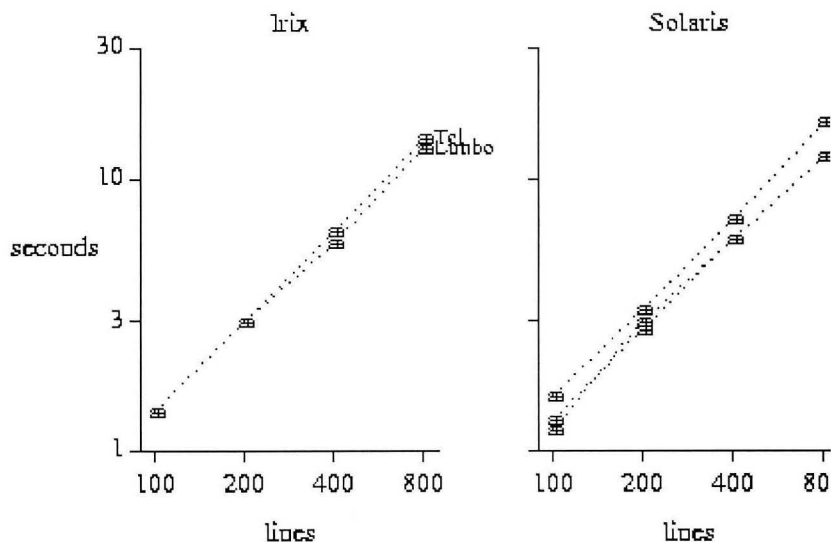
In Visual Basic, it is

```
For i = 1 To n
    bigtext.Text = bigtext.Text & CStr(i) & crlf
    bigtext.SelStart = Len(bigtext.Text)
    bigtext.SelLength = 0
    DoEvents
Next i
```

and in Java

```
for (int i = 1; i <= n; i++) {
    bigtext.appendText(i + "\n");
    Toolkit.getDefaultToolkit().sync();
}
```

...



Text insertion test

We explicitly requested a screen update after each insertion, except in Visual Basic. Omitting these mandatory updates reduces runtimes significantly, because changes to the display are buffered so that only the final result is visible. Runtimes for this test are

even less trustworthy than in earlier tests, since the display on Unix systems is managed by an X server that adds its own times into the mix. We used a stopwatch for Limbo on Unix, where reported times were unrelated to observed times.

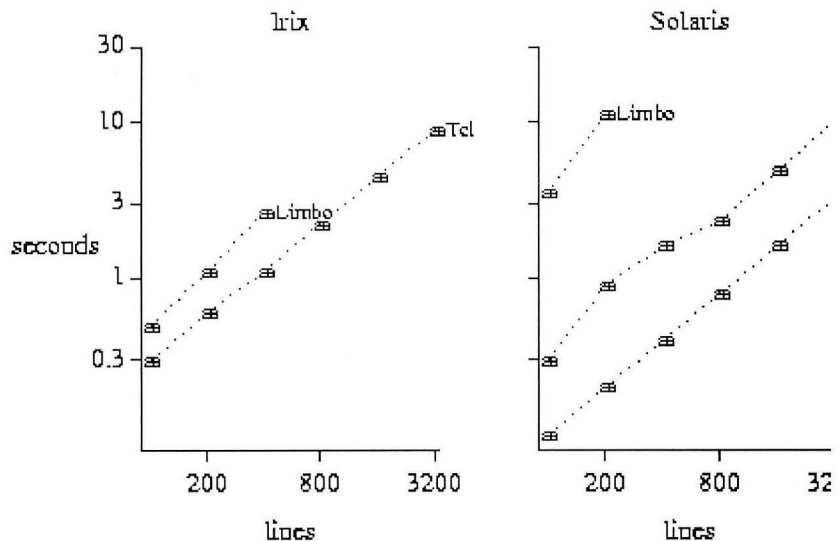
Visual Basic has no method for appending text to a `TextBox`, so creating each new display would appear to be a quadratic process. In spite of this, Visual Basic runtimes are quite reasonable. The screen flashes and slows the display, apparently because Visual Basic redisplay the first line after each insertion. Visual Basic offers the alternative `RichTextBox`, which has more features (size, font, color), and also overcomes the 32,000 character limit of the regular `TextBox`. The same test takes about 30 percent longer with a `RichTextBox`.

Drawing Tests

The other major time-consuming activity for user interfaces is drawing graphical objects. By measuring the time it takes to draw n lines in a fan, each with a small colored circle at the end, we hope to capture only the most basic behavior. We have not attempted to animate a scene, move any objects, or interact with them after they have been drawn.

The Java version is representative:

```
for (int i = 1; i <= lim; i++) {
    g.setColor(Color.blue);
    g.drawLine(MINX, MINY, MAXX, MINY + (int) (i * dy));
    g.setColor(Color.black);
    g.fillOval(MAXX, MINY + (int) (i * dy), d, d);
    g.setColor(Color.red);
    g.drawOval(MAXX+2, MINY + (int) (i * dy)+2, d-4, d-4);
}
...
```



Line/circle drawing test

Memory limits prevented testing Limbo for larger input sizes on Solaris and Windows, where an earlier version of the system is being used. In some cases, Limbo times appear to increase the more often the test is run.

These times are the least reliable, since each reflects design decisions in the implementation of the corresponding system. The times reported by internal timers in the programs are a lower bound on what a user sees; we have not included any additional delay that occurs before the screen is updated.

Runtimes increase significantly if explicit updates are added after each drawing operation, as they might be in some kinds of animations. Java and Visual Basic appear to update the screen continuously, even though this behavior was not requested. Tk and Limbo updates are turned off and either happen all at once (Tk) or sporadically (Limbo).

In general, graphics facilities differ significantly; our tests focus on elementary tasks. Tk's canvas widget (through Tcl or Limbo) is far richer and more flexible than what is available in Java or Visual Basic, but this test does not exercise those capabilities at all.

6 Compilation versus Interpretation

In simpler times, ``compilation" meant translation from source code via assembler into native instructions. A program was compiled once, perhaps with a bit of optimization, and the result was an executable binary image that never changed. Today, ``compilation" is more complicated.

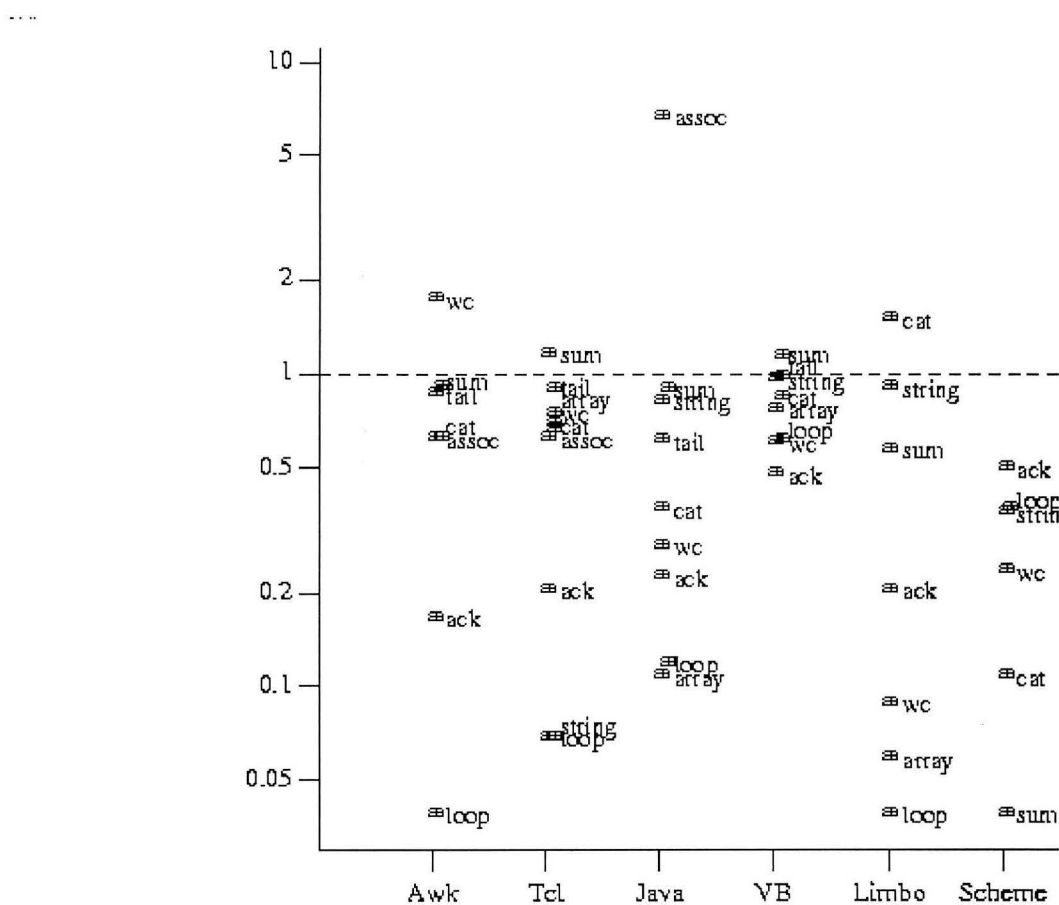
Scripting languages offer several intermediate positions, with multiple meanings even for ``interpretation." Tcl 7.6 and Scheme 7.3 are pure interpreters that repeatedly parse the source code as they run; this is extremely flexible, but slowest of all, as the graphs above show clearly. Awk translates its program into a tree that is walked during execution, Visual Basic creates an internal p-code representation, and Java and Limbo translate into byte codes for virtual machines (another form of p-code). Such preprocessing affords the speedups depicted in the graphs below.

In each case, it is possible to take the process further, either by translating the original program into something faster like C or C++, or by compiling the byte codes into machine instructions, either before execution or on demand during execution (``just in time compilation"). The graphs below show the results of some experiments with the ``compilation" facilities provided by several of these languages:

- The Awk to C++ translator described in [Kernighan 91] produces C++ that we compiled with Visual C++.
- Tcl 8.0b2 compiles into an internal byte code.

- Class files produced by Visual J++ are run with Microsoft's `jview`, which we believe to be a just-in-time compiler. The comparison is against Sun's JDK1.1 interpreter.
- Visual Basic 5.0 includes a native-mode compiler. The comparison is against Visual Basic 4.0.
- Limbo has a just-in-time compiler analogous to those for Java.
- VSCM [Blume 94] is a compact, portable implementation using a virtual machine written in C and a bytecode compiler written in Scheme itself.

The graphs show the ratio of compiled runtime to interpreted runtime. For this experiment, we ran the tests only on the PC, where we had the most languages available. We only tried the largest of the inputs used earlier, to reduce the contribution of startup to the runtime measurements.



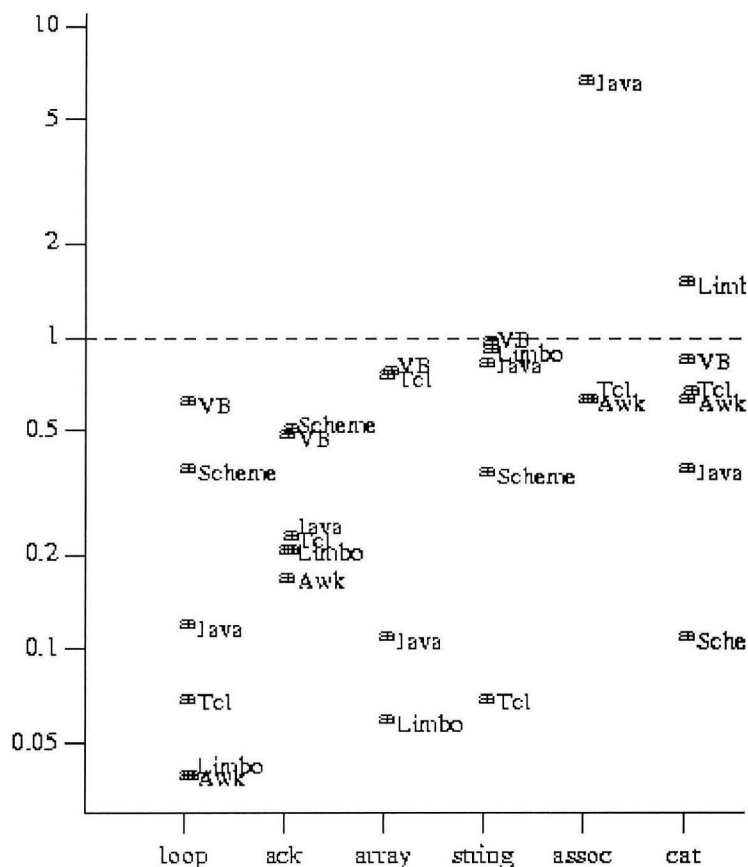
Compiled time / interpreted time, by language

The outlier here is the Java time for the associative array test. We believe the

problem is with storage management.

Some Scheme tests are omitted because they ran into trouble: failure to parse the input or an infinite loop. On the bright side, times returned by VSCM are very accurate.

...



Compiled time / interpreted time, by task

These graphs reveal a number of interesting facts. Compilation usually, but not always, reduces runtime; in a few cases it made performance much worse. Most of the improvements are modest; some languages (like Limbo) exhibit a large range of improvement, while others (like Visual Basic) exhibit only a narrow range. Not surprisingly, the largest improvements are for the simplest language features—those most like regular programming languages; thus, the basic loop test and Ackermann's function improve most, while I/O tests improve least (unless the I/O system is also coded in the language in question, as it is with Limbo and seems to be with Java).

7 Discussion and Conclusions

Clearly one might perform many other experiments, for instance to fill in more language features more systematically, to investigate regular expressions (in Awk, Perl and Tcl) or string searches (in Java and Visual Basic), or to study more user-interface components.

Another possibility would be to compare implementations of the same tools, since several of these languages are available in different implementations. For example, one might measure Awk, Gawk, Mawk and (on Windows 95) MKS Awk. There are many Java implementations, mainly focused on Windows. Unfortunately, these usually run only from within browsers, where security restrictions make it impossible to do the I/O tests.

Some of the I/O runtimes are remarkably long. Published benchmarks rarely include I/O tests. Perhaps if they did, language implementations would perform better than they do.

The amount of real memory makes a large difference in PC applications; once a program starts paging, runtimes become long and erratic. This happened with (at least) Awk, Perl, and Tcl. Upgrading our PC from 16Mb to 32Mb sped up several tests by a factor of two or more. (All results here are for 32Mb.)

Memory management also matters. It seems clear that some of our tests encountered problems with the implementation of garbage collection. The usual symptom is runtimes that increase during a series of tests. Other factors are simply beyond control, and perhaps even beyond knowing. Modern machines use several levels of cache to make slow main memory appear to operate nearly at the speed of cache memory, but different processors (all with the same nominal speed) have different amounts and kinds of cache.

The limitations of these results are important and bear repetition. These comparisons apply to specific language processors running on specific machines, and cannot be used to draw conclusions about overall performance differences of various languages in general. As we have worked on these experiments, we have been struck repeatedly by how often intuition is wrong. Some apparently small changes lead to unexpectedly large performance differences. For example,

- keeping track of the string length matters for the C string-construction test on Unix systems, but not on PC's;
- using Perl associative arrays where indexed arrays would serve can increase runtimes dramatically;
- in Tcl8.0, ``Code inside a procedure body is substantially faster than code outside any procedure." [Ousterhout, private communication.]

Even the intuitions of native speakers can be wrong. In response to comments, we

modified some examples to be more colloquial, and found that these ``better" versions sometimes did not run much faster, and might even run slower. For example,

- we wrote two Scheme versions of the timing loop, string construction, and word count, one using the tail-recursion favored by Scheme aficionados, the other using the barbarian import `set!`; the difference in performance between the two versions never exceeded ten percent, and did not consistently favor one version over the other.
- in earlier versions of this manuscript, Perl experts noticed leftovers introduced by the Awk-to-Perl translator *a2p*, and offered the reasonable criticism that it was unfair to compare the performance of Perl code that was mechanically generated with hand-coded programs in other languages. Removing spurious `chop` statements from the programs did improve their runtimes, but never by more than ten percent. On the other hand, it is about ten percent faster to create the array of lines in *tail* explicitly, rather than reading the input file all at once.

Needless to say, we advise all who want to know which version of a program will run faster to construct test programs and find out the truth for their language processor and machine.

Despite the preceding disclaimer, we essay the following summary of our observations:

Awk and Perl are similar in performance. Awk ran faster in about half the tests, contrary to the conventional wisdom that Perl is faster than Awk. Tcl is significantly slower than Awk and Perl, typically by a factor of five or ten, but the latest bytecode version narrows that gap significantly. Scheme appears to be very roughly comparable with Tcl (aside from volume I/O, where interpreted Scheme's performance is hopeless); again, various compilation techniques speed improve this behavior. Our experiments do not support the folklore that Java interpreters are 10 times slower than C, except when the excruciatingly slow I/O libraries are involved; otherwise the ratio is much smaller. Using buffered I/O functions improves runtimes, but by no more than a factor of two. Just-in-time compilation can have a significant effect, usually beneficial.

Runtime is only one measure of a programming language; another is suitability for task. Scripting languages are meant to be easy to program in; one sacrifices runtime and control for ease of expression. We did not try to measure expressiveness, but program size offers some clue. The total number of lines of code for the non-graphics tests ranges from 66 for Awk, 96 of Perl, 105 of Tcl, 170 of C and Scheme, 200 of Visual Basic, to around 350 for Java and Limbo.

Comparisons are a mainstay of computer literature. Conferences, journals, and magazines are full of tables and colorful charts that compare execution time or memory usage of one or more programs on different machines or implementations. In light of

the variability in results that we saw, however, we wonder whether similar variation lurks behind published benchmark studies as well. It does seem wise to take all such experiments including these with a large grain of salt.

Acknowledgements

Thanks to Jon Bentley, Mark-Jason Dominus, Lorenz Huelbergen, Brian Lewis, Doug McIlroy, John Ousterhout, Rob Pike, Arnold Robbins, Wolfram Schneider, Howard Trickey, and Phil Wadler, for helpful suggestions and comments on the manuscript. We are also grateful to Phil Wadler for performing some experiments with ML, and to Will Clinger for improving our Scheme programs and for his experiments comparing a number of Scheme implementations on our test programs [Clinger 97].

References

[Abelson 96] *Structure and Interpretation of Computer Programs* (second edition), H. Abelson, G. J. Sussman, J. Sussman, McGraw-Hill, 1996.

[Becker 87] R. A. Becker, L. Denby, R. McGill, A. R. Wilks, "Analysis of Data From the **Places Rated Almanac**," *The American Statistician*, August, 1987. By adjusting the weights with which individual components are combined into a final score, one can make almost any place look good or bad.

[Bentley 91] *An Elementary C Cost Model*, by Bentley, Kernighan and Van Wyk (Unix Review, February, 1991), describes one attempt to do systematic measurements of the cost of basic operations in a single language, C.

[Blume 94] *VSCM A Portable Scheme Implementation*, Matthias Blume.
www.cs.Princeton.edu/~blume/vscm.

[Booth 97] *Inner Loops*, by Rick Booth (Addison-Wesley, 1997), is an excellent reference on tuning PC programs.

[Caffeine 97] <http://www.webfayre.com/cm.html>. "CaffeineMark Java Benchmark," Pendragon Software.

[Clinger 97] William C. Clinger,
<http://www.ccs.neu.edu/home/will/TwoBit/kvwbenchmarks.html>. Compares the tests from this paper on a variety of Scheme implementations.

[Gosling 97] Private communication.

[Hardwick 97] <http://www.cs.cmu.edu/~jch/java/optimization.html>. Jonathan Hardwick's "Java Optimization" Web page contains a variety of benchmarks for Java programs and links to related material.

[Kernighan 91] B. W. Kernighan, "An AWK to C++ Translator," USENIX C++ Conference, Washington, DC, April, 1991.

[Scheme 91] The **Revised**⁴ *Report on the Algorithmic Language Scheme* (Nov 1991), edited by Klinger and Rees, is the official definition.

The following lists standard references for the various languages and some related material on performance evaluation. These are not cited in the text.

The AWK Programming Language, by Aho, Kernighan and Weinberger (Addison-Wesley, 1988), is the standard reference; Arnold Robbins' *Effective AWK Programming* (SSC, 1996) is a newer reference.

Perl 5 is best described in *Programming Perl, Second Edition* by Wall, Christiansen and Schwartz (O'Reilly, 1996).

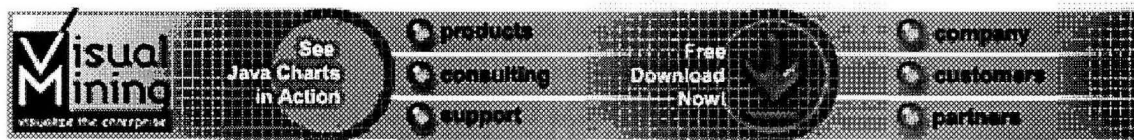
Ousterhout's *Tcl and the Tk Toolkit* (Addison-Wesley, 1994) is a definitive description of version 7.4/3.6.

Flanagan's *Java in a Nutshell* (O'Reilly, 1996 [first edition]) is an excellent introduction; Arnold and Gosling's *The Java Programming Language* (Addison-Wesley, 1996) is a good alternative.

Limbo and Inferno are best described on the Inferno web page,
<http://inferno.bell-labs.com>.

There are myriad books on Visual Basic; Microsoft's reference and user guides are definitive.

Copyright © 1998 Lucent Technologies Inc. All rights reserved.



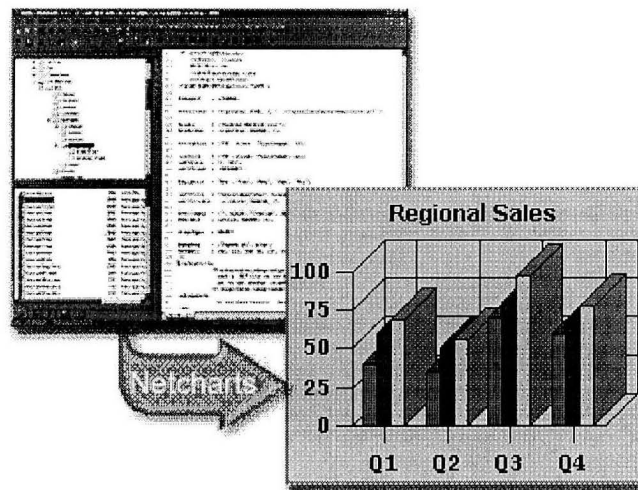
Products

Products

- Overview
- NetCharts
- NetCharts Beans
- ChartWorks
- More Info

NetCharts™

NetCharts is a suite of Java applets that allows HTML authors, web developers, and Java programmers to quickly and easily create on-line charts from virtually any source, with minimal coding.



NetCharts applet in HTML code yields an instant Bar

NetCharts is the industry-leading Java charting components and have been incorporated into products from Oracle, Sybase, Cisco, Hewlett-Packard, Net Networks, and others.

- [About NetCharts](#)
- [Key Benefits](#)
- [Programmatically Drive the Look and Feel of Charts](#)
- [Integration](#)
- [Product Specifications](#)

About NetCharts

NetCharts is a suite of Java applets that allows HTML authors, web developers, and Java programmers to quickly and easily create on-line charts from virtually any source, with minimal coding.

➤ [See the Chart Examples](#) to discover the rendering power of NetCharts.

Key Benefits

- Provides a fast and easy system for building sophisticated on-line charts

- Reduces the time and costs in building on-line charts
- No need to start from scratch, use one of our 50 ready-to-use examples
- Charts can be dynamically updated in real time from virtually any database, your charts are never out of date
- Allows you to exploit the power of the Internet for disseminating power to help manage your business processes and promote your products
- Runs on any Java supported platform including UNIX, Windows NT, Macintosh, OS/2, Network PCs and JavaStations.

Programmatically Drive the Look and Feel of Charts

NetCharts gives developers the ability to create and manipulate on-demand charts through a rich set of attributes or commands called the Chart Definition Language. Charts described in CDL are rendered on the fly in much the same way that Postscript documents. This unique approach gives developers the ability to programmatically "drive" a chart's look and feel.

Integration

- NetCharts can be incorporated into HTML, DHTML, popular application tools and web development tools, including:
 - Microsoft Visual Basic
 - Microsoft Visual InterDev
 - Cold Fusion
 - Silver Stream
 - NetDynamics
 - Inprise JBuilder
 - IBM Visual Age
 - Sun Java Workshop
 - Symantec Visual Café
 - Independently configurable axes
- Date/time information plotting for real-time data feeds
- NetCharts allows use of over 100 colors and fonts

Product Specifications

NetCharts supports any Java 1.0.x or 1.1.x enabled platform, including P network computers. All applets can be viewed through any Java compatible browser and support the use of the JDBC interface to retrieve parameter definitions from one or more database servers or data sources, including Oracle, Sybase DB2, Microsoft SQL Server, dBase, Access and Excel.

All company names and product names referenced are trademarks or registered trademarks of their respective companies.



[products](#) | [consulting](#) | [support](#) | [examples](#) | [company](#) | [customers](#) | [partners](#) | [free download](#) | [contact us](#) | [site map](#)

©2000 Visual Mining, Inc. • 15825 Shady Grove Road • Suite 20 • Rockville, MD 4008

voice: 1-800-308-0731, 301-795-2200 • fax: 301-947-8293 • info@visualmini.com



Product Overview

The NetCharts suite of Java¹ applets has been developed specifically to provide both Web authors and Java developers with an easy-to-use method for including powerful graphics in their applications, *without the need for programming*. NetCharts provides the most commonly used business and scientific charts, including:

- Piechart
- Barchart
- Line Chart
- XY Chart
- Box Chart
- Diagram
- Combo Chart
- Stock Chart
- Strip Chart
- Time Chart

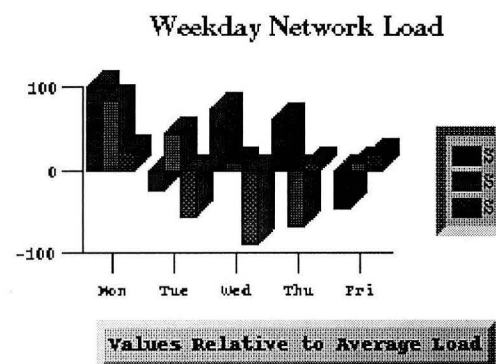
As well as the following useful applets and utilities:

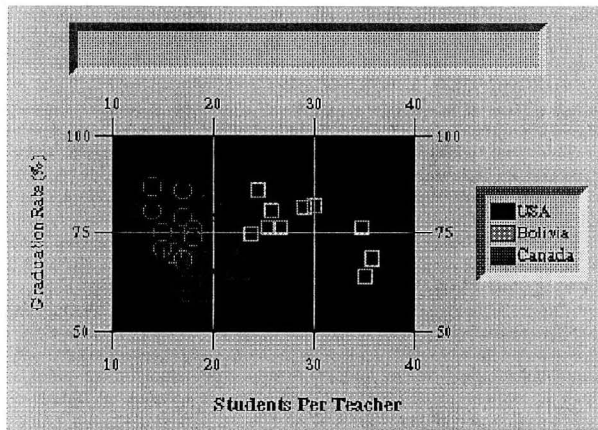
- MultiChart - for displaying multiple, resizable charts in a single applet
- Snapshot Applet/Server - for printing HTML documents containing charts
- Snapshot Utility - for generating snapshots of HTML documents containing charts without using a browser

All of the NetCharts applets use the same easy-to-learn and easy-to-use syntax to define various parameters associated with each applet. (See [Parameter Definitions](#) for details concerning parameter definitions.)

HTML Authoring

HTML authors can quickly include graphics in their HTML presentations simply by adding an applet tag, with a few parameter definitions, to their documents. With over 50 example files provided, Web authors can see the range of possible displays and can easily modify them for their specific needs. NetCharts can be installed and operational on your Web site in less than 30 minutes.



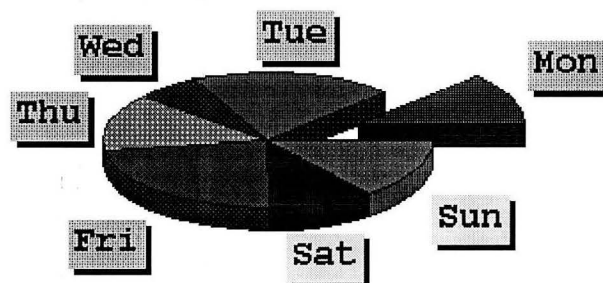


User Interaction

All NetCharts charts allow the HTML or Java developer to specify one or more *Active Labels* that are automatically displayed when the mouse cursor *dwells* (or hovers) over a given data region. Such labels can be used to present additional information to the user, such as the exact data value, or to explain something about that particular data region. The developer has complete control over the look-and-feel of the Active Labels through parameter definitions.

In addition to displaying a label, the developer can also specify an optional URL associated with each label. If the user clicks the mouse on the Active Label, then NetCharts will automatically retrieve the given URL via its associated Web Server and display the resulting document in a new window specified by the developer. For example, users may click on a given bar to display tabular data associated with it, or they may click on a piechart slice in order to see the breakdown for that slice, displayed in another bar chart or piechart. By chaining one chart with another, the developer can create a Web of charts that allow the user to freely browse a data depository.

Since all NetCharts charts allow the use of GIF files as a background image, it is very easy to create an Image Map using an XY chart to display data points on the map. As the user points at a point, an associated Active Label could automatically display information about that location. GIF files representing geographic locations, building layouts, or network topologies can be processed using the same techniques. (A specialized NetCharts applet is currently being developed to make this process easier and even more flexible.)

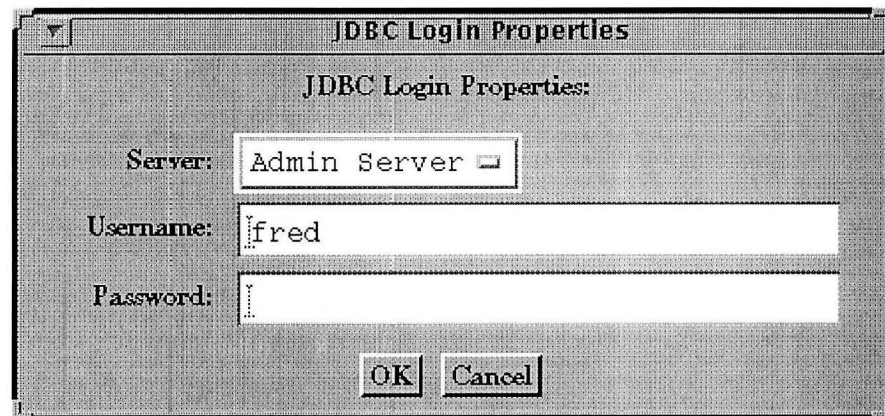


Average Daily Network Utilization

Integrated Database Support

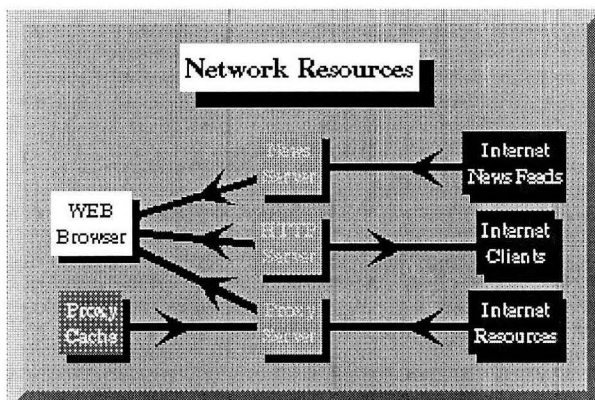
All NetCharts applets support the use of the JDBC interface to retrieve parameter definition data values from one or more database servers or data sources, including Oracle, Sybas Informix, Microsoft SQL Server, dBase, Access, Excel, just to name a few.

The JDBC interface is completely platform and server independent, providing a wide range of implementation alternatives. The JDBC requires the use of one or more drivers in order to connect to the database server. Such drivers are sold by a wide range of vendors, including major database vendors and quite a few 3rd party vendors. NetCharts applets can be used with any JDBC-compliant software, allowing the widest range of database applications.



CGI Script Generation

When static documents don't meet end-user needs, it is common to generate documents dynamically using a range of CGI programming techniques on the Web Server. NetCharts applets can easily fit into such an environment, because **all** data chart parameters, including values, can be defined in an HTML document or in a text file accessible via the Web server. NetChart applet definition can easily be generated by any CGI script, including those that access a database. For example, a PERL script may be used to access the database and to generate an HTML document containing a Piechart. Only a handful of HTML lines are needed to specify the data for the Piechart, the NetCharts Java classes take care of the rest. See [Parameter Definitions](#) for more details.



Dynamic Updates

Increasingly, HTML documents are being used to provide a user-interface for controlling a system or network. For example, Web applications have been developed to manage the performance load of multiple servers on a network, displaying that information through a Web browser. In such situations, all NetCharts applets can be configured to automatically connect to a *Parameter Server* on start.

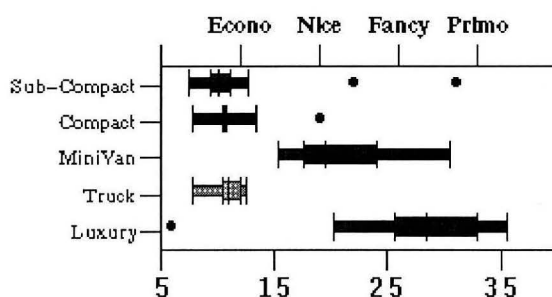
The Parameter Server can be built using any programming language or script that supports TCP/IP communication. Once connected, the Parameter Server can update any or all of applet parameters, thereby causing it to update its display at any time. In this way, continuously updated charts can be displayed using a Web browser. The parameter definitions passed between the Parameter Server and the NetCharts applet are in the exact same format as used in the HTML file, thereby decreasing integration problems. See [Parameter Definitions](#) for more details.



JavaScript Control

Netscape Navigator ² developers who use JavaScript commands and functions to control Web applications can easily interact with any chart embedded in the HTML document through the **loadParams()** method that is available for all NetCharts applets. The **loadParams()** accepts a text string that consists of one or more parameter definitions. Any number of **loadParams()** calls can be made and any number of parameters can be defined with each. Once all parameters have been defined or updated, the "Update" command is sent (via the **loadParams()** method) to cause the NetCharts applet to redraw the chart. Here again, any NetCharts parameter can be defined or dynamically updated from JavaScript using the same commands used in HTML documents, CGI scripts, or parameter servers. See [Parameter Definitions](#) for more details.

Applications and Applets



For those projects that require either standalone Java applications or custom applets, the NetCharts applets can be either embedded without modification into a custom application or applet, or can be *extended* via the standard object mechanism. In such cases, the Java programmer can choose between the convenience of the

Car Prices (\$K)

loadParams() interface or the control a by accessing the underlying NetCharts directly. See Embedded Applets for m details.

Programmer documentation is provided (with the Developer License) for all NetCharts and underlying classes, allowing the Java programmer to use the objects in any way con with Java programming practices. A number of examples are also provided to assist nov programmers with embedding NetCharts applets into their applications or applets. A fe of code is all that is needed in most situations.

¹Java is a registered trademark of Sun Microsystems.

²Netscape Navigator is a registered trademark of Netscape.

Copyright 1996-98, NetFactory, Inc.

403590

BIBLIOTECA
UNIVERSIDAD DE MONTERREY