

D
SDI
\$ 2500.-

UNIVERSIDAD DE MONTERREY

DIVISION DE ARQUITECTURA, DISEÑO E INGENIERIA



DESARROLLO DE UN SISTEMA DE TARJETAS DE
PREPAGO INTERNET CON HERRAMIENTAS NET, SQL
SERVER, EQUIPOS MAXTNT Y CISCO ACS RADIUS,
UTILIZANDO LA METODOLOGIA MS SOLUTION
FRAMEWORK

PROGRAMA DE EVALUACION FINAL
QUE PRESENTAN:

FABIAN GUADALUPE ORTA GARZA
GERARDO MANUEL CANEPA MARTINEZ

EN OPCION AL TITULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES

SAN PEDRO GARZA GARCIA, N. L.

DICIEMBRE, 2003

UNIVERSIDAD DE MONTERREY

DIVISIÓN DE ARQUITECTURA, DISEÑO E INGENIERÍA



DESARROLLO DE UN SISTEMA DE TARJETAS DE
PREPAGO INTERNET CON HERRAMIENTAS .NET, SQL
SERVER, EQUIPOS MAXTNT Y CISCO ACS RADIUS,
UTILIZANDO LA METODOLOGÍA MS SOLUTION
FRAMEWORK

PROGRAMA DE EVALUACIÓN FINAL QUE PRESENTAN:

FABIAN GUADALUPE ORTA GARZA
GERARDO MANUEL CANEPA MARTINEZ

040.0016
077d
2003

EN OPCIÓN AL TÍTULO DE:

INGENIERO EN SISTEMAS COMPUTACIONALES

SAN PEDRO GARZA GARCÍA, N. L.

DICIEMBRE, 2003.



Universidad de Monterrey

Vicerrectoría de Educación Superior

pef
PROGRAMA DE
EVALUACION
FINAL

Vicerrectoría de Educación Superior

FormPEF#8

Carta para aprobación del documento académico final

San Pedro Garza García, 26 de Noviembre 2003

Ing. Elizabeth Gutiérrez de la Garza

Director del Programa de Ingeniero en Sistemas Computacionales
División de Arquitectura, Diseño y Ingeniería
Universidad de Monterrey.

Por medio de la presente le comunico a usted que he revisado el documento académico final cuyo título es Desarrollo de un Sistema de Tarjetas de Prepago Internet con Herramientas .NET, SQL Server, Equipos MaxTNT y Cisco ACS Radius Utilizando la Metodología MS Solution Framework respecto a su forma y contenido, el cual es presentado por los alumnos del Programa de Evaluación Final **Gerardo Manuel Canepa Martínez y Fabián Guadalupe Orta Garza** de la Universidad de Monterrey. Considero que el documento académico final tiene las características de ejecución y calidad que la Universidad de Monterrey demanda, además cumple con los objetivos y alcance planteado en el anteproyecto autorizado por el Comité de PEF al inicio del semestre.

Por mi parte, estoy a su disposición para cualquier aclaración que usted desee hacer.

Cordialmente,

Ing. Marta Sylvia del Río Guerra
Departamento de Ciencias Computacionales

c.c.p. Ing. Alejandro Pozas
c.c.p. Gerardo Manuel Canepa Martínez
c.c.p. Fabián Guadalupe Orta Garza

DEDICATORIAS

A **mis padres**, mi **hermano** y **mis 2 hermanas** que siempre me apoyaron desde iniciar mi carrera, fracasar y recuperarme del mismo para llegar al final de mi carrera.

A mis **amigos** y **amigas**, su entendimiento y paciencia valió la pena.

A mi tía **María de Jesús Martínez González**, que está peleando contra una enfermedad muy difícil, por todos los días que no te fui a ver.

"I never had all the answers, in this life I have failed as much as I've succeeded, but I love my wife, I love my life, and I wish you MY KIND OF SUCCES." Jerry Mcguire

Gerardo Manuel Canepa Martínez

A **mi esposa**, **mis padres**, **mis 2 hermanos** y **mi hermana**, que me apoyaron durante el transcurso de toda la carrera.

A todos mis **amigos**, **compañeros** y **maestros**, por su ayuda y paciencia.

"La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica." Aristóteles

Fabián Guadalupe. Orta Garza

AGRADECIMIENTOS

De Gerardo:

A **Juan Carlos Balboa** por darnos la oportunidad de desarrollar este producto de suma importancia para la compañía.

A **Erich V. Hauske** y **Jorge López** por su valioso apoyo, aportación y asesoría durante el desarrollo del producto.

A **Ruth Alemán** y **Rene Pereyra** por sus aportaciones y asesoría necesaria para cumplir los objetivos necesarios.

A **Marta Sylvia del Río**, **Consuelo Jiménez**, **Martín Jiménez** y **Elizabeth Gutiérrez** por su tiempo, dedicación, asesoramiento y paciencia durante el desarrollo del sistema y la carrera.

Al **Lic. Mario Ortiz Mendoza**, aunque ya no es parte de la UDEM y lleve solo 2 meses de clase con él, fue pieza fundamental para entender como tratar con los clientes durante el desarrollo de un proyecto, conceptos base que aprendí para el desarrollo de todos mis proyectos.

A **Gustavo García**, **Arturo Caballero**, **Miguel Tadeo**, **Jorge Aldrette** y **Francisco Aguilar**, **Jorge Llerenas**, **Enrique Rocha**, **Hugo Lara**, **Edgar Guzmán** y **Edgar Jiménez**. Los compañeros de equipo y buenos amigos que tuve en diferentes proyectos a lo largo de la carrera... ¡GRACIAS!

A **Axtel**, aunque estuve solo 6 meses, la experiencia laboral que obtuve fue muy valiosa para el éxito del proyecto.

De Fabián:

A **Juan Carlos Balboa**, **Erich Von Hauske** y **Jorge López** por brindarnos el tiempo y la oportunidad para el desarrollo del proyecto. Además por su gran apoyo, asesoramiento y paciencia durante el desarrollo del proyecto.

A **Ruth Alemán** y **René Pereyra** por sus valiosísimas aportaciones en el proyecto.

A **Marta Sylvia del Río**, **Consuelo Jiménez**, **Martín Jiménez** y **Elizabeth Gutiérrez** por su tiempo, dedicación, asesoramiento y paciencia durante el desarrollo del sistema y la carrera.

A **Isadora Rocha**, **Iris Leal**, **Marcela Bovio** y **Olivier Sens**, compañeros y amigos de los cuales aprendí muchísimo durante el transcurso de la carrera.

Prologo

"Lo más humano de nosotros es la tecnología,
que vive como extensión de sus inventores."
Marshall McLuhan

En la llamada "era de la información" el estar constantemente comunicado se ha convertido en una necesidad para las personas. Por ello, las empresas de telefonía, que surgieron en 1878 en México, han prosperado en el país. La International Telecommunications Union (ITU), basándose en datos de la SCT, estima que para el año 2009 habremos alcanzado niveles de telefonía comparables a los de los países más desarrollados.

En la obra de mayor repercusión sobre las telecomunicaciones modernas, "A Mathematical Theory of Communication", C. E. Shannon y W. Weaver definen el concepto de comunicación de una manera muy sencilla: "comunicación son todos aquellos procedimientos por medio de los cuales una mente afecta a otra". Esto incluye voz, texto impreso o escrito, música, artes, teatro y danza. Incluso, se amplía la idea anterior para incluir la posibilidad de comunicación entre máquinas: "comunicación son todos aquellos procedimientos por medio de los cuales un mecanismo afecta la operación de otro", y se menciona explícitamente, como ejemplo, el control de aviones.

Al observar este entorno y el auge que las telecomunicaciones han adquirido, resulta claro que día a día se desarrollan nuevos productos para que el consumidor explote al máximo los beneficios de estar comunicado. Productos para voz, video, Internet o localización satelital son ofertados cada día por las diferentes compañías telefónicas. Aun más, la interconexión de tecnologías ha permitido que hoy en día pueda accederse Internet por un celular, por ejemplo. Todo esta enfocado en la satisfacción del cliente.

El siguiente proyecto es un esfuerzo de Marcatel por continuar siendo una empresa competitiva y de vanguardia. Desea ofrecer a sus clientes las tarjetas prepagadas para voz e Internet. Este Proyecto de Evaluación Final aporta una nueva forma de acceso a los clientes, a la vez que integra el servicio de voz que ya ofrecía.

A su vez, el proyecto permitió estrechar aun más los vínculos de Marcatel con la academia, en este caso, el Programa de Ingeniero en Sistemas Computacionales de la UDEM. Tanto Fabián como Gerardo, alumnos de esta carrera, tuvieron oportunidad de aprender muchísimo durante todo el desarrollo del PEF. Considero que han logrado resultados excelentes, y que este sistema es una prueba de la calidad y entusiasmo que aplican a todos sus proyectos. En especial, felicito a Fabián Orta por su dedicación a la programación y a Gerardo Canepa por el esfuerzo invertido en el documento. No me resta más que felicitarlos ampliamente y exhortarlos a que continúen poniendo en alto el nombre de la Universidad de Monterrey.

Ing. Marta Sylvia del Río, M.C.

San Pedro, Garza García, a 28 de noviembre del 2003

ÍNDICE

INTRODUCCIÓN	1
1. EMPRESA	3
1.1 Datos Generales de la Empresa	3
1.2 Misión de la Empresa.....	4
1.3 Productos y Servicios.....	4
1.4 Infraestructura Computacional de la Empresa	7
1.5 Organigrama de la Empresa.....	8
1.6 Resumen y Conclusión del Capítulo	10
2. JUSTIFICACION DEL PROYECTO	11
2.1 Situación Actual y descripción del Problema.....	11
2.2 Solución Propuesta.....	12
2.3 Alcances del Proyecto	15
2.4 Objetivo General.....	16
2.5 Objetivos Particulares.....	16
2.6 Lo que han hecho otras organizaciones	17
2.6.1 Todito.com – Todito Card	17
2.6.2 Terra Prepago	18
2.6.3 Comparación de productos existentes con el propuesto.....	18
2.7 Importancia del Proyecto.....	19
2.8 Resumen y Conclusión del Capítulo	20
3. MARCO TEÓRICO	21
3.1 Plataforma Microsoft .NET.....	21
3.1.1 Herramientas Microsoft Visual Studio .NET y Framework.....	22
3.1.1.1 .NET Framework	22
3.1.1.2 Visual Basic .NET	24
3.1.1.3 ASP .NET	28
3.1.2 Microsoft SQL Server 2000	31

3.2	Equipo Lucent y Cisco	34
3.2.1	Cisco ACS Raduis	34
3.2.2	MAX TNT Universal Gateway.....	36
3.4	Metodología Microsoft Solution Framework aplicada según las necesidades de MarcaTel	37
3.4.1	Introducción.....	38
3.4.1.1	El Modelo en Espiral y Cascada.....	38
3.4.1.2	El ciclo de vida de <i>MSF</i>	39
3.4.1.3	Los principios que utiliza <i>MSF</i>	40
3.4.1.4	Conceptos a Conocer de <i>MSF</i>	40
3.4.1.4.1	Clientes o Consumidores.....	40
3.4.1.4.2	Inversionistas.....	41
3.4.1.4.3	¿Qué es una solución?.....	41
3.4.1.4.4	Lineamientos.....	42
3.4.1.4.5	Alcance.....	42
3.4.2	Fases del Modelo <i>MSF</i> y Metas.....	42
3.4.3	Fase de Previsión - Estrategia y Alcance.....	43
3.4.4	Fase de Planeación - Planificación y Pruebas de Concepto.....	47
3.4.5	Fase de Desarrollo.....	50
3.4.6	Fase de Estabilización.....	51
3.4.7	Fase de Liberación - Despliegue.....	53
3.5	Resumen y Conclusión de Capitulo.....	55
4.	ANÁLISIS	56
4.1	Alcance y Estrategia	56
4.1.1	Oportunidad de Negocio.....	56
4.1.2	Visión.....	57
4.1.3	Solución.....	57
4.1.4	Beneficios.....	58
4.1.5	Análisis de Usuarios.....	67
4.1.6	Requerimientos.....	68
4.1.7	Equipo de Trabajo, distribución de competencias y responsabilidades.....	71
4.1.8	Plan de trabajo.....	74
4.1.9	Matriz de Riesgos y Planes de Contingencia.....	75
4.2	Planeación	79
4.2.1	Arquitectura de Sistema.....	79
4.2.2	Plan de Laboratorio – Escenarios.....	81
4.3	Oportunidad de Negocio, Visión y Solución para 2do y 3er Ciclo de MSF 89	
4.3.1	Segundo Ciclo MSF, Sistema administrativo de tarjetas.....	91
4.3.1.1	Beneficios.....	91
4.3.1.2	Análisis de Usuarios.....	95

4.3.1.3	Requerimientos	95
4.3.1.4	Equipo de trabajo, distribución de competencias y responsabilidades.....	95
4.3.1.5	Arquitectura de Sistema	97
4.3.1.6	Plan de Laboratorio	100
4.3.2	Tercer Ciclo MSF, Sistema de perfiles	104
4.3.2.1	Beneficios.....	104
4.3.2.2	Análisis de Usuarios	107
4.3.2.3	Requerimientos	107
4.3.2.4	Equipo de trabajo, distribución de competencias y responsabilidades.....	107
4.3.2.5	Arquitectura de Sistema	108
4.3.2.6	Plan de Laboratorio	110
4.4	Conclusiones del Capítulo	114
5.	DISEÑO	115
5.1	Bases de Datos.....	115
5.1.1	Modelo Entidad Relación.....	116
5.1.2	Construcción de Tablas de Tarjetas	122
	Tabla Serie	122
	Tabla Tarjetas.....	122
5.1.3	Construcción de Tablas para Log de Transacciones.....	123
	Actividad TC	124
	LogActividades	124
5.1.4	Integración de Tablas de Tarifas	124
	AreaGeográfica	124
	DefHorario	125
	DominioAreaGeografica	125
	DominioTipoDia	125
	Horario.....	125
	Tarifa	125
	TipoDialD.....	125
5.1.4	Integración de Tablas Radius y Conexión de Usuarios.....	126
	RadiusAccounting.....	126
	UsersConnected.....	126
5.2	Diseño de Pantallas para Servicio a Cliente	126
5.3	Diseño de Pantalla para Monitoreo de Saldos.....	128
5.4	Diseño de pantalla de reportes	130
5.5	Conclusiones del capítulo	132

6. CONSTRUCCIÓN	133
6.1 Estándares de Programación	134
6.2 Construcción del Sistema	135
6.2.1 Módulos del Cliente	135
6.2.1.1 Objeto Tarjeta.....	136
6.2.2 Módulos Sistema y Cliente	138
6.2.1.2 Servicio para manejo de conexiones y tarifas	138
6.2.1.3 Componente de Monitoreo de Saldos	142
6.2.1.4 Plataforma SNMP, Telnet y MaxTNT	145
6.2.1.5 Módulo Bancario.....	147
6.2.3 Módulos Administrativos.....	151
6.2.3.1 Tarifas para <i>Dial-up</i>	151
6.2.3.2 Conciliaciones Bancarias	152
6.2.3.3 Generación de Archivo de Pagos	152
6.2.3.4 Adaptación a bases de datos de tarjetas voz	152
6.2.3.5 Reportes establecidos	153
6.3 Gestión de Incidencias	155
6.3.4 Módulo de Consulta de Saldo	156
6.3.5 Módulo de Traspaso de Saldo de Tarjeta a Tarjeta.....	159
6.3.6 Módulo de Recarga de Tarjeta con Tarjeta de Crédito.....	163
6.3.7 Módulo de Compra de Tarjeta con Tarjeta de Crédito.....	173
6.3.8 Módulo de Monitoreo de Saldos	176
6.3.9 Monitoreo de Conexiones y Saldos	178
6.3.10 Generación de Cuentas Válidas para Tarjetas Prepago	179
6.3.11 Pruebas sobre ambiente Cisco ACS Radius y Max TNT Server ..	180
6.3.12 Generación de archivo de pagos para Facturación	182
6.4 Pruebas piloto (maqueta)	183
6.5 Plan de Despliegue (Liberación)	185
6.6 Captura de 2da Incidencia (Versiones Futuras)	185
6.7 Resumen de herramientas y su aplicación al proyecto.	186
6.8 Conclusiones del capítulo	186
CONCLUSIONES	187
APÉNDICE REFERENCIAS BIBLIOGRÁFICAS DE INTERNET	191
ANEXO A INTERFACES	226
ANEXO B CÓDIGO FUENTE DEL SISTEMA	232

ÍNDICE DE TABLAS

TABLA 2.1 TARIFAS DE TERRA PREPOAGO	18
TABLA 2.2 COMPARACIÓN DE PRODUCTOS.....	19
TABLA 3.1 VALOR Y DESCRIPCIÓN DE CÓDIGOS DE PAQUETE RADIUS. [HILL, 2001].....	35
TABLA 3.2 COMPARACIÓN ENTRE EL PRODUCTO Y LA SOLUCIÓN MSF. [MSF, 2002].....	41
TABLA 3.3 FORMATO PARA TABLA DE RIESGOS POSIBLES [MSF,2002]	45
TABLA 3.4 FORMATO PARA TABLA DE RIESGOS POSIBLES [MSF,2002]	46
TABLA 3.5 FORMATO PARA ANÁLISIS DE RIESGOS Y PRIORIDADES [MSF,2002].....	47
TABLA 3.6 PLANIFICACIÓN DE DESARROLLO DE COMPONENTES [MSF,2002].....	48
TABLA 3.7 PLANIFICACIÓN DE DESARROLLO DE COMPONENTES [MSF,2002].....	50
TABLA 4.1 DETALLE DE COSTOS TANGIBLES DEL PROYECTO.....	60
TABLA 4.2 DETALLE DE COSTOS TANGIBLES DEL PROYECTO.....	61
TABLA 4.3 IMPLICACIONES DE QUE EL PROYECTO NO SE LLEVE A CABO	62
TABLA 4.4 BENEFICIOS TANGIBLES DEL PROYECTO	63
TABLA 4.5 RETORNO DE INVERSIÓN DEL PROYECTO	64
TABLA 4.6 COMPARACIÓN DE SISTEMAS DE INTERNET PREPAGADO .	65
TABLA 4.7 RETORNO DE INVERSIÓN DE SISTEMA INTERNET PREPAGADO MARCATEL	66
TABLA 4.8 RETORNO DE INVERSIÓN PARA SISTEMA ADMINISTRADOR DE CUENTAS <i>DIAL-UP</i>	66
TABLA 4.9 ACTIVIDADES A REALIZAR.....	71
TABLA 4.10 DISTRIBUCIÓN DE RESPONSABILIDADES DEL 1-20	72
TABLA 4.11 DISTRIBUCIÓN DE RESPONSABILIDADES DEL 21-44	73
TABLA 4.12 PLAN DE TRABAJO DE ACTIVIDADES A REALIZAR..	74
TABLA 4.13 TABLA DE IDENTIFICACIÓN DE RIESGOS	75
TABLA 4.14 TABLA DE CLASIFICACIÓN Y DESCRIPCIÓN DE RIESGOS.	76
TABLA 4.15 ASIGNACIÓN DE PROBABILIDADES A RIESGOS.	77
TABLA 4.16 ASIGNACIÓN DE PLAN DE CONTINGENCIA A RIESGOS....	78
TABLA 4.17 TABLA PARA CASO DE CONEXIÓN DE CLIENTE.	81
TABLA 4.18 CASO PARA USO DE TARJETA FÍSICA.....	82
TABLA 4.19 CASO PARA COMPRA DE TARJETA VÍA INTERNET.....	83
TABLA 4.20 CASO PARA CONSULTA DE TARJETA POR INTERNET	84
TABLA 4.21 CASO PARA RECARGA DE TARJETA POR INTERNET.....	85
TABLA 4.22 CASO PARA TRASPASO DE SALDO DE TARJETA A TARJETA POR INTERNET	86
TABLA 4.23 CASO PARA EXPIRACIÓN DE TARJETAS.	87
TABLA 4.24 CASO PARA TARIFICACIÓN DE TARJETAS.	88

TABLA 4.25 COSTOS TANGIBLES PARA SISTEMA ADMINISTRATIVO DE TARJETAS	92
TABLA 4.26 COSTOS DE OPERACIÓN PARA SISTEMA ADMINISTRATIVO DE TARJETAS	93
TABLA 4.27 IMPLICACIONES PARA QUE PROYECTO NO SE LLEVE A CABO PARA SISTEMA ADMINISTRATIVO DE TARJETAS. .	93
TABLA 4.28 BENEFICIOS DE SISTEMA ADMINISTRATIVO DE TARJETAS	94
TABLA 4.29 RETORNO DE INVERSIÓN PARA SISTEMA ADMINISTRATIVO DE TARJETAS.....	94
TABLA 4.30 ACTIVIDADES A REALIZAR PARA SEGUNDA VUELTA MSF	95
TABLA 4.31 DISTRIBUCIÓN DE ACTIVIDADES PARA SEGUNDA VUELTA	96
TABLA 4.32 CASO PARA GENERACIÓN DE LOTES.....	101
TABLA 4.33 CASO PARA REEMPLAZO DE LOTES.....	102
TABLA 4.34 CASO PARA CANCELACIÓN DE LOTES.	103
TABLA 4.35 COSTOS TANGIBLES DE PROYECTO PARA PERFILES ...	104
TABLA 4.36 COSTOS DE OPERACIÓN PARA PERFILES	105
TABLA 4.37 IMPLICACIONES PARA QUE PROYECTO NO SE LLEVE A CABO PARA SISTEMA DE PERFILES.....	105
TABLA 4.38 BENEFICIOS DE SISTEMA DE PERFILES	106
TABLA 4.39 RETORNO DE INVERSIÓN PARA SISTEMA DE PERFILES.	106
TABLA 4.40 ACTIVIDADES A REALIZAR PARA TERCER VUELTA MSF	107
TABLA 4.41 DISTRIBUCIÓN DE ACTIVIDADES PARA TERCERA VUELTA.	108
TABLA 4.42 CASO PARA ACTIVACIÓN DE PERFILES.....	111
TABLA 4.43 CASO PARA ACTIVACIÓN DE PERFILES.....	112
TABLA 4.44 CASO PARA DEPURACIÓN DE PERFILES.....	113
TABLA 5.1 DESCRIPCIÓN DE ATRIBUTOS DE TABLA TARJETA_DIALUP	117
TABLA 5.2 DESCRIPCIÓN DE ATRIBUTOS DE TABLA ESTATUS.....	117
TABLA 5.3 DESCRIPCIÓN DE ATRIBUTOS DE TABLA TARJETAS.....	117
TABLA 5.4 DESCRIPCIÓN DE ATRIBUTOS DE TABLA SERIE	118
TABLA 5.5 DESCRIPCIÓN DE ATRIBUTOS DE TABLA TIPOTARJETA..	118
TABLA 5.6 DESCRIPCIÓN DE ATRIBUTOS DE TABLA DENOMINACIÓN	119
TABLA 5.7 DESCRIPCIÓN DE ATRIBUTOS DE TABLA TARIFA	119
TABLA 5.8 DESCRIPCIÓN DE ATRIBUTOS DE TABLA AREAGEOGRAFICA	119
TABLA 5.9 DESCRIPCIÓN DE ATRIBUTOS DE TABLA DOMINIO AREAGEOGRAFICA.....	120
TABLA 5.10 DESCRIPCIÓN DE ATRIBUTOS DE TABLA DEFHORARIO	120
TABLA 5.11 DESCRIPCIÓN DE ATRIBUTOS DE TABLA HORARIO	120
TABLA 5.12 DESCRIPCIÓN DE ATRIBUTOS DE TABLA TIPODIA.....	120
TABLA 5.13 DESCRIPCIÓN DE ATRIBUTOS DE TABLA DOMINIOTIPODIA	121

TABLA 5.14 DESCRIPCIÓN DE ATRIBUTOS DE TABLA LOGACTIVIDADES	121
TABLA 5.15 DESCRIPCIÓN DE ATRIBUTOS DE TABLA ACTIVIDADES	121
TABLA 5.16 TABLA TARJETA	122
TABLA 5.17 TABLA SERIE	122
TABLA 5.18 TABLA NIPS	122
TABLA 5.19 TABLA NIPS_DIALUP	123
TABLA 5.20 TABLA DENOMINACION	123
TABLA 5.21 TABLA DENOMINACION	123
TABLA 5.22 TABLA ACTIVIDADTC (ACTIVIDADES CON TARJETA DE CRÉDITO).....	124
TABLA 5.23 TABLA LOGACTIVIDADES	124
TABLA 5.24 TABLA AREAGEOGRAFICA.....	124
TABLA 5.25 TABLA DEFHORARIO	125
TABLA 5.26 TABLA DOMINIOAREAGEOGRAFICA	125
TABLA 5.27 TABLA DOMINIOTIPIODIA	125
TABLA 5.28 TABLA HORARIO	125
TABLA 5.29 TABLA TARIFA	125
TABLA 5.30 TABLA TIPODIAID	125
TABLA 5.31 TABLA RAIUSACCOUNTING	126
TABLA 5.32 TABLA LOGACTIVIDADES	126
TABLA 5.33 ICONO DE ESTADOS PARA MONITOREO DE SALDOS.....	129
TABLA 6.1 ESTÁNDARES DE PROGRAMACIÓN.....	135
TABLA 6.2 MÉTODOS UTILIZADOS POR EL OBJETO TARJETA	137
TABLA 6.3 PRUEBA DE CONSULTA DE SALDO PARA VALIDACIÓN DE FOLIO 1.....	156
TABLA 6.4 PRUEBA DE CONSULTA DE SALDO PARA VALIDACIÓN DE FOLIO 2.....	156
TABLA 6.5 PRUEBA DE CONSULTA DE SALDO PARA VALIDACIÓN DE FOLIO 3.....	157
TABLA 6.6 PRUEBA DE CONSULTA DE SALDO PARA VALIDACIÓN DE FOLIO 4.....	157
TABLA 6.7 PRUEBA DE CONSULTA DE SALDO PARA VALIDACIÓN DE FOLIO 5.....	158
TABLA 6.8 PRUEBA DE TRASPASO DE SALDO PARA FOLIOS DE TARJETA 1	159
TABLA 6.9 PRUEBA DE TRASPASO DE SALDO PARA FOLIOS DE TARJETA 2	160
TABLA 6.10 PRUEBA DE TRASPASO DE SALDO PARA FOLIOS DE TARJETA 3	161
TABLA 6.11 PRUEBA DE TRASPASO DE SALDO PARA FOLIOS DE TARJETA 4	161
TABLA 6.12 PRUEBA DE TRASPASO DE SALDO PARA FOLIOS DE TARJETA 5	162
TABLA 6.13 PRUEBA DE RECARGA PARA FOLIOS DE TARJETA 1.....	163
TABLA 6.14 PRUEBA DE RECARGA PARA FOLIOS DE TARJETA 2.....	164

TABLA 6.15 PRUEBA DE RECARGA PARA FOLIOS DE TARJETA 3.....	165
TABLA 6.16 PRUEBA DE RECARGA PARA FOLIOS DE TARJETA 4.....	165
TABLA 6.17 PRUEBA PARA VALIDACIÓN DE CANTIDAD A RECARGAR 1	166
TABLA 6.18 PRUEBA PARA VALIDACIÓN DE CANTIDAD A RECARGAR 2	166
TABLA 6.19 PRUEBA PARA VALIDACIÓN DE CANTIDAD A RECARGAR 3	167
TABLA 6.20 PRUEBA PARA VALIDACIÓN DE TIPO DE TARJETA DE CRÉDITO 1	167
TABLA 6.21 PRUEBA PARA VALIDACIÓN DE TIPO DE TARJETA DE CRÉDITO 2	168
TABLA 6.22 PRUEBA PARA VALIDACIÓN DE TIPO DE BANCO 1.....	168
TABLA 6.23 PRUEBA PARA VALIDACIÓN DE TIPO DE BANCO 2.....	169
TABLA 6.24 PRUEBA PARA NÚMERO DE TARJETA DE CRÉDITO 1	170
TABLA 6.25 PRUEBA PARA NÚMERO DE TARJETA DE CRÉDITO 2	170
TABLA 6.26 PRUEBA PARA NÚMERO DE TARJETA DE CRÉDITO 3	171
TABLA 6.27 PRUEBA PARA VALIDACIÓN DE FOLIO DE TARJETA	171
TABLA 6.28 PRUEBA PARA VALIDACIÓN DE CANTIDAD A RECARGAR 1	172
TABLA 6.29 PRUEBA PARA CANTIDAD A RECARGAR 2	172
TABLA 6.30 PRUEBA PARA VALIDACIÓN DE INVENTARIO 1.....	173
TABLA 6.31 PRUEBA PARA VALIDACIÓN DE INVENTARIO 2.....	174
TABLA 6.32 PRUEBA PARA ENVÍO DE CORREO ELECTRÓNICO.....	175
TABLA 6.33 PRUEBA DE MONITOREO DE SALDOS PARA MUESTRA DE SALDO CORRECTO.....	176
TABLA 6.34 PRUEBA DE MONITOREO DE SALDOS PARA CLIENTES MÚLTIPLES	177
TABLA 6.35 PRUEBA DE MONITOREO DE SALDOS PARA DESCONEXIÓN POR FALLA	177
TABLA 6.36 PRUEBA DE MONITOREO DE CONEXIONES Y SALDOS PARA SELCCIÓN DE TARIFA CORRECTA.....	178
TABLA 6.37 PRUEBA DE GENERACIÓN DE CUENTAS VÁLIDAS.....	179
TABLA 6.38 PRUEBA DE AMBIENTE MAXTNT	180
TABLA 6.39 PRUEBA DE AMBIENTE RADIUS	181
TABLA 6.40 PRUEBA DE GENERACIÓN DE ARCHIVOS A FACTURACIÓN	182
TABLA 6.41 PRUEBA PILOTO DE MONITOREO DE SALDOS POR CLIENTE PARA DEDUCIR TARIFA.....	183
TABLA 6.42 PRUEBA PILOTO DE MONITOREO DE CONEXIONES Y SALDOS PARA REGISTRAR CONEXIONES	184
TABLA 6.43 PRUEBA PILOTO DE MONITOREO DE CONEXIONES Y SALDOS PARA DESCONEJÓN FORZADA POR FALTA DE SALDO	184
TABLA 6.44 RESUMEN DE HERRAMIENTAS A UTILIZAR.	186

ÍNDICE DE FIGURAS

FIGURA 1.1 ARQUITECTURA MARCANET DEDICADO [MARCATEL, 2003]	6
FIGURA 1.2 ESQUEMA DE CONEXIÓN MARCANET DIAL UP [MARCATEL, 2003]	6
FIGURA 1.3 DIAGRAMA DE SERVICIO MARCANET VPL [MARCATEL, 2003]	7
FIGURA 1.4 ORGANIGRAMA DE MARCATEL INTERNATIONAL [BALBOA, 2003]	8
FIGURA 2.1 PROCESO PARA UTILIZAR PRODUCTO MARCANET DIAL-UP	12
FIGURA 2.2 SOLUCIÓN PROPUESTA PARA TARJETAS PREPAGADAS INTERNET	14
FIGURA 3.1 AMBIENTE CLR COMMON LANGUAGE RUNTIME [PLATT,2001]	23
FIGURA 3.2 CÓDIGO INTERMEDIO DE MICROSOFT [PLATT,2001]	23
FIGURA 3.3 PÁGINA INICIAL DE VISUAL STUDIO.NET [PERTUOSOS, 2002]	24
FIGURA 3.4 VENTANA PARA CREAR NUEVO PROYECTO [PERTUOSOS, 2002]	24
FIGURA 3.5 VENTANA DE DESARROLLO VB.NET [PERTUOSOS, 2002]	25
FIGURA 3.6 VENTANA DE COMPONENTES PARA VB.NET [PERTUOSOS, 2002]	26
FIGURA 3.8 PANTALLA DE CODIGO [PERTUOSOS, 2002]	27
FIGURA 3.9 PANTALLA DE DISEÑO ASP.NET [RESELMAN, 2001]	29
FIGURA 3.10 PANTALLA DE CODIGO ASP.NET [RESELMAN, 2001]	30
FIGURA 3.11 PANTALLA DE <i>QUERY ANALYZER</i>	31
FIGURA 3.12 PANTALLA DE <i>STORED PROCEDURES</i>	32
FIGURA 3.13 PANTALLA DE <i>ENTERPRISE MANAGER</i>	33
FIGURA 3.14 PANTALLA DE <i>JOBS</i>	34
FIGURA 3.15 PAQUETE DE PROTOCOLO RADIUS [HILL, 2001]	35
FIGURA 3.16 – CICLO DE VIDA DE PROYECTO EN CASCADA. [MSF, 2002]	38
FIGURA 3.17 – CICLO DE VIDA DE PROYECTO EN ESPIRAL [MSF, 2002]	39
FIGURA 3.18 – MODELO DE CICLO DE VIDA PARA <i>MSF</i> [MSF, 2002]	39
FIGURA 3.19 – CICLO DE VIDA PARA <i>MSF</i> [MSF, 2002]	43
FIGURA 3.20 – EJEMPLO DE ESCENARIO CON DIAGRAMA DE FLUJO	49
FIGURA 4.3 GRÁFICA COMPARATIVA ENTRE PROYECTOS	67
FIGURA 4.4 ARQUITECTURA DE SISTEMA	80
FIGURA 4.5 ESCENARIO PARA CONEXIÓN DE CLIENTE	81
FIGURA 4.6 ESCENARIO PARA USO DE TARJETA FÍSICA	82
FIGURA 4.7 ESCENARIO PARA COMPRA DE TARJETA VÍA WEB	83
FIGURA 4.8 ESCENARIO PARA CONSULTA DE TARJETA POR INTERNET	84

FIGURA 4.9 ESCENARIO RECARGA DE TARJETA POR INTERNET.	85
FIGURA 4.10 ESCENARIO RECARGA DE TARJETA POR INTERNET.	86
FIGURA 4.11 ESCENARIO PARA EXPIRACIÓN DE TARJETAS.	87
FIGURA 4.12 ESCENARIO PARA EXPIRACIÓN DE TARJETAS.	88
FIGURA 4.13 ARQUITECTURA DEFINIDA INICIALMENTE PARA PROYECTO.	98
FIGURA 4.14 ESCENARIO PARA GENERACIÓN DE LOTES	101
FIGURA 4.15 ESCENARIO PARA REEMPLAZO DE LOTES	102
FIGURA 4.16 ESCENARIO PARA CANCELACIÓN DE LOTES	103
FIGURA 4.17 ARQUITECTURA DEFINIDA PARA TERCERA VUELTA DEL CICLO MSF.	109
FIGURA 4.18 ESCENARIO PARA ACTIVACIÓN DE PERFILES	111
FIGURA 4.19 ESCENARIO PARA ACTUALIZACIÓN DE PERFILES.....	112
FIGURA 4.20 ESCENARIO PARA DEPURACIÓN DE PERFILES.....	113
FIGURA 5.1 MODELO ENTIDAD-RELACIÓN PREPAGO INTERNET	116
FIGURA 5.2 DISEÑO UTILIZADO EN PÁGINAS PRINCIPALES DE INTERNET PARA SERVICIO A CLIENTE	127
FIGURA 5.3 RESULTADO FINAL DE APLICACIÓN DE DISEÑO PARA COMPRA DE TARJETAS.	128
FIGURA 5.4 PANTALLA DE MONITOREO DE SALDOS.....	128
FIGURA 5.5 ICONO DE MONITOREO DE SALDO.....	129
FIGURA 5.6 DISEÑO DE PANTALLA PARA REPORTES ESTABLECIDOS	130
FIGURA 6.1 RESULTADO DE REPORTE PARA USUARIOS CONECTADOS AL SERVICIO PREPAGADO.	153
FIGURA 6.2 RESULTADO DE REPORTE PARA ACTIVIDADES CON TARJETA DE CRÉDITO EN INTERNET.....	154
FIGURA 6.3 RESULTADO DE REPORTE DE COBRO DIARIO DE TARIFAS A CUENTAS DE PREPAGO.	155

ABSTRACT

El desarrollo de un sistema de tarjetas de prepago Internet con herramientas .NET, SQL Server, equipos MaxTNT, Cisco ACS Radius utilizando la metodología MS Solution Framework fue desarrollado para la empresa de Marcatel. El objetivo de este sistema fue introducir el producto de tarjetas prepago Internet por teléfono al mercado por medio de Marcatel International.

Se utilizó la metodología Microsoft Solution Framework, que permite gran flexibilidad en la entrega de documentos, acuerdos y construcción del sistema. Con las herramientas de Visual Studio .NET y SQL Server se desarrolló el producto de tarjetas de prepago. Los equipos MAXTNT y ACS Radius se encargaron de recibir y autorizar las llamadas hechas. El cliente final puede interactuar con el producto por medio de servicios creados en ASP .NET. Mediante un servicio contratado por Banorte, los clientes pueden comprar o recargar su tarjeta prepagada en Internet con tarjeta de crédito.

INTRODUCCIÓN

La empresa de telecomunicaciones Marcatel International se dedica al área de servicios de voz y datos. Una de sus áreas importantes es el de servicio de Internet. Se sabe que el Internet ha vuelto el mundo de las comunicaciones más pequeño, rápido y eficiente.

El servicio de Internet puede ser obtenido por vía telefónica a baja velocidad a costo razonable a través de un dispositivo llamado MODEM, que traduce las señales de teléfono a señales que entienda la computadora. También puede ser por servicios de alta velocidad, que siempre están disponibles para el usuario pero a costo muy elevado.

Marcatel ofrece estos servicios a sus clientes, pero con la gran competencia de hoy y las fuertes demandas del cliente requieren otra manera para ofrecer este servicio de Internet. Hay clientes que no necesitan utilizar el servicio de Internet por mucho tiempo y quieren pagar solamente por lo que utilizan sin la necesidad de contratar un servicio con contratos y pagos mensuales. Las tarjetas prepagadas para uso de Internet permiten cumplir con estas demandas.

La empresa Marcatel International decide entrar al mercado de las tarjetas prepagadas, desarrollando un sistema que pueda administrar y

operar el producto. El servicio prepagado de Internet se enfoca al medio telefónico.

El presente proyecto utiliza la metodología Microsoft Solution Framework (MSF), herramientas de visual Studio .NET, equipos Max TNT y Radius y SQL Server 2000. Las herramientas son altamente recomendadas para servicios basados en Internet. Se presentan los resultados de las primeras dos fases de la metodología MSF, su planeación, escenarios creados para el producto y análisis de costos y beneficios. En la fase de desarrollo se encuentran los resultados del diseño. La mayoría de la tercera fase se dedica a construcción y pruebas. Se incluyen los modelos creados para la base de datos y diseño utilizado para los servicios que el cliente de prepagado utiliza. La cuarta fase incluye el plan de liberación del producto y futuras versiones a realizar.

1. EMPRESA

En este capítulo se describe la empresa Marcatel International. Aquí se realizó el proyecto de desarrollo del sistema de tarjetas de prepago Internet para clientes y usuarios administrativos. Se mencionan los datos generales de la empresa, productos y servicios que ofrece tanto de voz como datos, infraestructura computacional para los desarrolladores y organigrama de la empresa.

1.1 Datos Generales de la Empresa

Marcatel es una empresa dedicada al mercado de telecomunicaciones. Se dedica principalmente a telefonía de larga distancia para negocios, transmisión de datos e imágenes, Internet, líneas privadas, números 800 y tarjetas telefónicas de post pago y prepago. [Marcatel, 2003]

Marcatel nació de una alianza entre la empresa mexicana RadioBeep y las empresas americanas IXC.y Westel. En la actualidad los accionistas de

la empresa son RadioBeep y Progress International, ésta última estadounidense. [Marcatel, 2003]

La red está situada dentro de los mejores sitios estratégicos, como resultado se tienen tarifas más competitivas y atractivas para el mercado. Se cuenta con 5 centrales en esta red que están ubicadas en: Ciudad de México, Monterrey, Puebla, Guadalajara y Torreón; 6 repetidoras distribuidas entre la red y cruce fronterizo entre Reynosa, Tamaulipas y Mcallen, Texas [Marcatel, 2003] [Pereyra, 1999]

La empresa cuenta con una red de fibra óptica de 2500 Km., representa la primera etapa de un plan para permitir un servicio sin interrupciones y con 30,240 llamadas simultáneas. Mediante esta red se abarcan a más de 100 ciudades alrededor de la República Mexicana para originar llamadas de larga distancia.[Marcatel, 2003]

Todos los servicios con los que cuenta esta empresa se apoyan en su proveedor alemán Siemens que es una empresa líder en tecnología a nivel mundial. [Marcatel, 2003]

Con las tecnologías mencionadas anteriormente, no solo tiene la capacidad de proveer servicios de voz. Marcatel tiene la capacidad de proveer servicios de transmisión de datos vía Internet a través de sus líneas telefónicas (*Dial-Up*). Utiliza tecnologías como Lucent Serie TNT los cuales están instalados en Guadalajara, Monterrey y México. Recientemente, la empresa comenzó a proveer servicios de Internet dedicado por medio de ADSL (*Asynchronous Digital Subscriber Line*) [Marcatel, 2003]

1.2 Misión de la Empresa

“Ofrecer los mejores productos y servicios en telecomunicaciones contribuyendo en el proceso de diversificación tecnológica y globalización de las empresas mexicanas.” [Marcatel, 2003]

1.3 Productos y Servicios

Marcatel ofrece una amplia serie de productos para que los negocios puedan realizar llamadas de larga distancia. Entre estos, se encuentran los siguientes:

- *Marcatel Negocios*: Este servicio es para empresas que consumen entre \$500 y \$50,000 pesos en llamadas de larga distancia. El cliente

no paga renta mensual ni contratación. El objetivo de este producto es cobrar las llamadas de larga distancia y dar descuentos en cuanto al volumen de llamadas, darle opción de donde recibir sus facturas y pagar en centros bancarios como Banorte, Bancomer, Citibank o Serfin.

- *Tarjeta Extensión Marcatel*: Mediante esta tarjeta el cliente puede hacer llamadas de larga distancia en ubicaciones externas a su negocio y que se carguen a su cuenta empresarial.
- *Servicio con número 800*: Permite a la empresa contratar una línea telefónica la cual sus clientes se pueden comunicar desde cualquier estado de la República Mexicana sin costo, y ofrecer sus servicios al cliente.
- *Servicio de Control de Larga Distancia*: Este servicio permite hacer auditoria de todas las llamadas hechas por una empresa para tener un control de los gastos.

[Pereyra, 1999]

Marcatel también cuenta con una amplia serie de productos con servicios de Internet. Los servicios para la transmisión de datos son los siguientes:

- *MarcaNet Dedicado*: con este producto, la empresa puede estar conectada permanentemente a Internet sin la necesidad de un módem y línea telefónica. Este servicio cuenta con tres elementos:
 - Ruteador – Equipo administrador del tráfico de entre la red local de la empresa e Internet.
 - Acceso Local: Comunicación entre el equipo local y equipo utilizado por Marcatel
 - Puerto de Internet: Permite el acceso a Internet mediante un enlace entre el nodo central de Marcatel y el *backbone* que se encuentra en Estados Unidos.

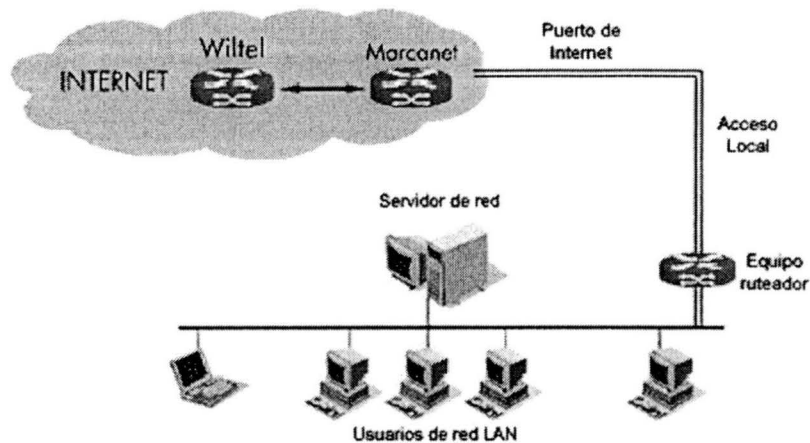


Figura 1.1 Arquitectura MarcaNet Dedicado [Marcatel, 2003]

- *MarcaNet Dial Up*: Este producto tiene como objetivo proveer a los clientes de un acceso a Internet vía telefónica. Este servicio cuenta con lo siguiente:
 - o Creación de Cliente: Definir el nombre de usuario y contraseña para tener acceso a Internet.
 - o Correo Electrónico de 10 MB.
 - o Espacio virtual de 5 MB para subir una página con dominio virtual.

[Marcatel, 2003]

La conexión a Internet se realiza a través de una computadora con sistema operativo Windows® o OS de Macintosh®. Con un módem se realiza una llamada a las instalaciones de Marcatel donde se le pedirá el nombre de usuario y contraseña para poder autorizar el acceso. La figura 1.2 describe este proceso más claramente.

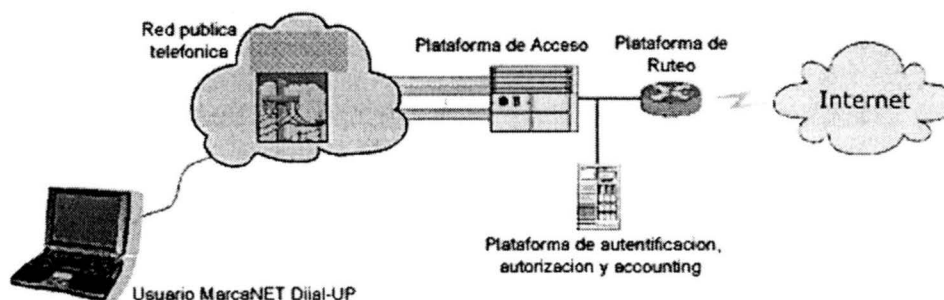


Figura 1.2 Esquema de Conexión MarcaNet Dial Up [Marcatel, 2003]

- *MarcaNet Burstable*: Este servicio permite cobrar a las empresas el uso de Internet sólo por lo que utilizan. Este uso se mide en mega bytes tanto recibidos y enviados. Se cobra mediante una cuota

establecida según el ancho de banda y cuenta con servicios de correo electrónico y páginas de Internet.

- **MarcaNet VPL:** Denominado Línea Virtual Privada como lo establecen sus siglas, este servicio permite a las empresas hacer el envío de datos directamente a través de las redes privadas virtuales de Marcatel. Esto crea una mayor seguridad en el envío de información entre unidades de la misma empresa. [Marcatel, 2003] A continuación se presenta un ejemplo del servicio VPL:

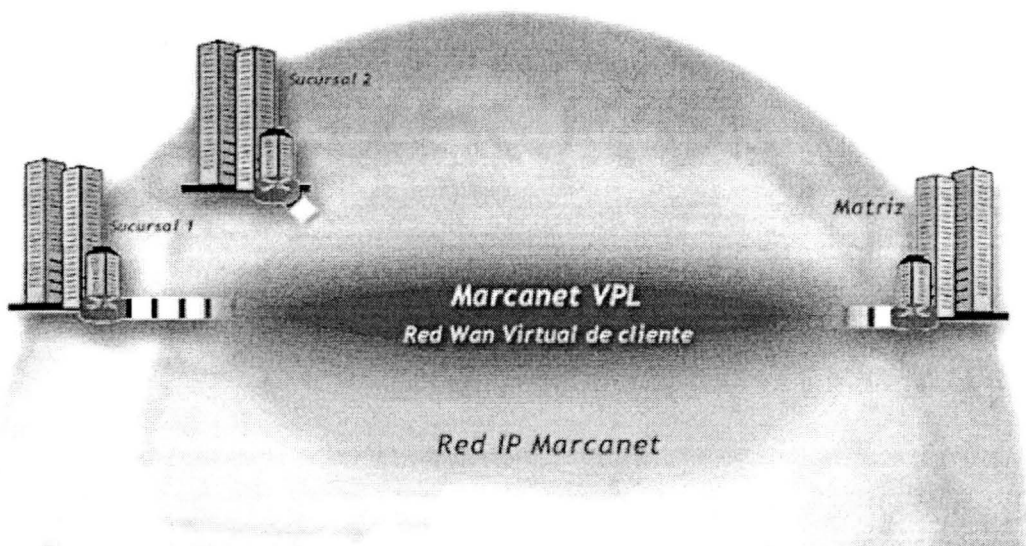


Figura 1.3 Diagrama de Servicio MarcaNet VPL [Marcatel, 2003]

1.4 Infraestructura Computacional de la Empresa

Marcatel cuenta con la siguiente infraestructura computacional para sus desarrolladores:

Hardware

- Maquinas desktop IBM®, Alaska® y Compaq® y portátiles IBM® con procesador Pentium® 2, 3 y IV Discos Duros con capacidad entre 3GB y 40GB, memoria RAM entre 128MB.

Software

- Windows 2000 Workstation o Advanced Server
- Visual InterDev 6.0
- Visual Basic 6.0
- Microsoft Office 2000
- MS Project 2000
- SQL Server 2000
- Microsoft Visual Studio .NET
- MS Framework. Service Pack 1.0, 1.1 o 1.2.

1.5 Organigrama de la Empresa

La figura 1.4 muestra el organigrama de la empresa donde se puede ubicar las diferentes divisiones que tiene la empresa.

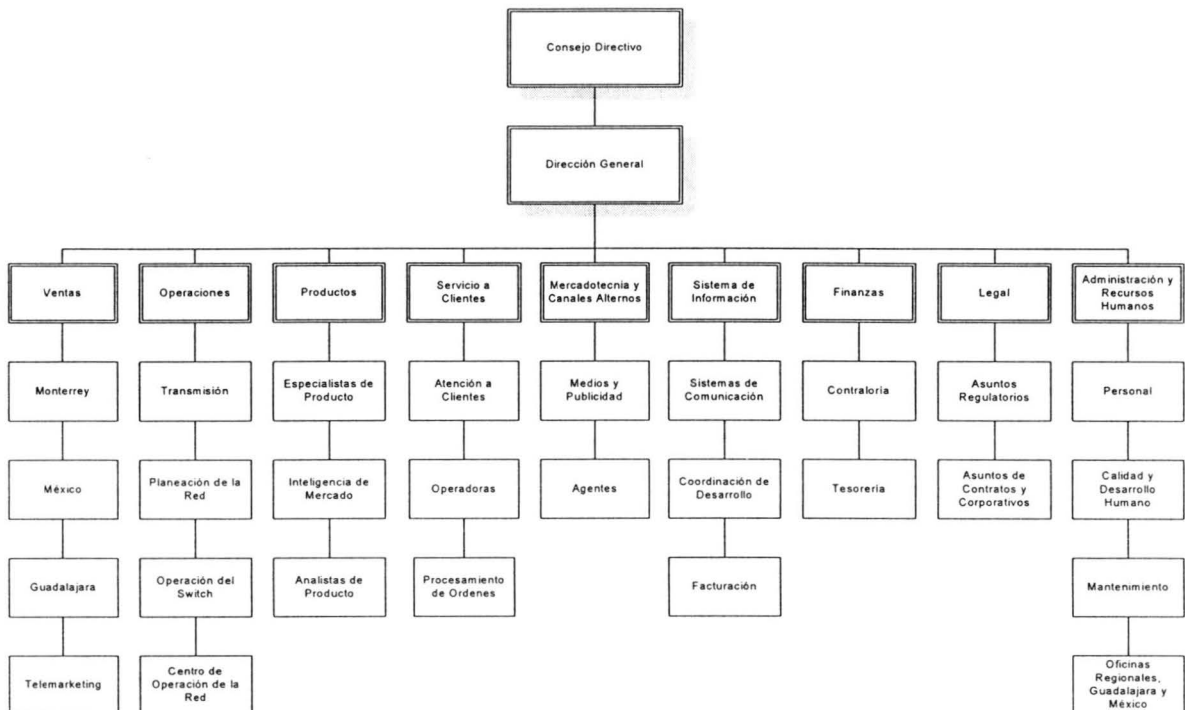


Figura 1.4 Organigrama de Marcatel International [Balboa, 2003]

A continuación se presenta una breve descripción de cada área.

Dirección General

Es responsable de establecer y lograr los objetivos de acuerdo a las estrategias planteadas por el Consejo de Administración de MARCATEL.

Administración y Recursos Humanos

Es responsable del ambiente organizacional y físico de MARCATEL, del reclutamiento y selección, del seguimiento de los logros del personal, de una equitativa compensación y de la planificación de su desarrollo.

Finanzas

Es responsable de la planeación financiera, control presupuestal y controles internos, cumplimiento de obligaciones fiscales, manejo y control de activos de la empresa, conseguir financiamiento del capital de trabajo con socios o mercados financieros, así como proponer a la dirección general y consejo del manejo apropiado de la estructura financiera.

Servicio a clientes

Es responsable de asegurar la satisfacción y permanencia de los clientes de MARCATEL, de la administración y procesamiento de información clientes nuevos y/o modificaciones a los ya existentes.

Legal

Es responsable de participar, asesorar, verificar y dar seguimiento al cumplimiento de los aspectos legales (Regulatorios, Contratos y Corporativos) inmersos en las actividades de MARCATEL con el fin de que existan una certidumbre jurídica en la operación de la empresa.

Operaciones

Es responsable del crecimiento, mantenimiento, optimización y administración de la red de fibra óptica de MARCATEL y sus elementos (*switches*, repetidoras, sitios, etc.), asegurando su continua operación.

Sistemas de Información

Es responsable de automatizar los procesos de MARCATEL, crear y soportar la infraestructura de comunicación interna todo esto con el fin de optimizar recursos y minimizar esfuerzos.

Mercadotecnia

Conocer las fuerzas, debilidades, amenazas y oportunidades de Marcatel, del mercado, la competencia y a la industria para desarrollar estrategias en conjunto con las áreas comerciales con el fin de desarrollar nuevos productos, incrementar las ventas y proyectar una imagen sólida, vanguardista y confiable en nuestros clientes y el mercado.

Ventas

Es responsable de aumentar el nivel de facturación y la penetración en el mercado de los productos de MARCATEL en la plaza asignada a través de los Ejecutivos de Venta de cada plaza y de la optimización de los recursos asignados a esta.

[Balboa, 2003]

1.6 Resumen y Conclusión del Capítulo

El capítulo actual dio a conocer la empresa Marcatel, productos que ofrece y estructura organizacional interna. Marcatel es una empresa de telecomunicaciones a nivel empresarial grande con fuerte actitud competitiva y productos enfocados a la transmisión de datos y telefonía de larga distancia.

2. JUSTIFICACION DEL PROYECTO

Este capítulo describe el problema actual que se presenta en Marcatel International, y explica la razón por la que esta empresa desea ingresar al mercado de tarjetas prepagadas Internet. Posteriormente se introduce la solución propuesta que especifica la funcionalidad y los alcances que tendrá este proyecto. Una vez planteada la solución se justifica el proyecto describiendo el aprendizaje adquirido y la importancia del producto desarrollado para la empresa.

2.1 Situación Actual y descripción del Problema

Actualmente Marcatel cuenta con un área de servicios llamada Dial-up, por medio de la cual ofrece una conexión a Internet de 56Kbps vía módem. Este servicio actualmente está dejando de ser atractivo a los clientes. En el mercado existen otros servicios muy parecidos y con más ventajas como lo son: Internet por cable, Internet de altas velocidades o Internet prepago. Los usuarios nuevos o actuales toman la opción de comprar su conexión con empresas de la competencia, descartando servicios como el que Marcatel ofrece, ya que no satisface sus necesidades. [Von Hauske, 2003] [Balboa, 2003]

La figura 2.1 presenta el proceso de conexión.



Figura 2.1 Proceso para Utilizar Producto MarcaNet Dial-Up

El Internet prepago es un servicio nuevo que ofrece a los usuarios llevar control de su cuenta de Internet, ya que permite comprar la cantidad necesaria en tiempo-Internet (mediante una tarjeta) para usarla cuando lo necesiten, sin pagar mensualidades y sin necesidad de contratos. Además permite la conexión a Internet desde lugares alejados de la ciudad, donde quizá no haya empresas que ofrezcan alguno de los servicios. Estas tarjetas tienen una cuenta relacionada en un servidor interno de la empresa de telecomunicaciones, que autoriza el acceso al servicio de Internet. [Von Hauske, 2003] [Balboa, 2003]

Por eso Marcatel decide lanzar al mercado una tarjeta de Internet prepago. Para ello necesita un sistema que administre el servicio desde la creación de tarjetas, hasta el cobro y recarga de las mismas. Este servicio tiene como mercado meta tanto a personas residenciales como empresariales. Este servicio se unirá en un futuro con el producto de Dial-up que ya está disponible en estos momentos. [Von Hauske, 2003] [Balboa, 2003]

2.2 Solución Propuesta

Se propone elaborar un sistema con herramientas como Visual .NET, SQL Server 2000 y Active Server Pages para que este servicio pueda operar. Para esto se deberá realizar lo siguiente:

- Elaborar una serie de servicios disponibles en Web para el cliente que permitan consultar saldos que se encuentran en una base de datos, compra de tarjetas virtuales y recarga de las mismas.
- La recarga consiste en aumentar el saldo disponible y puede ser de una tarjeta prepagada a otra o por tarjeta de crédito. Para recargas

con tarjeta de crédito el servicio debe validarse, registrar la actividad en la base de datos de transacciones y hacer la operación con el banco. Para recargas mediante otra tarjeta prepagada debe validarse la cuenta de cada tarjeta para permitir la transacción. [Von Hauske, 2003] [Balboa, 2003]

- El cliente puede comprar tarjetas físicas prepagadas para uso de Internet con los distribuidores autorizados que tiene Marcatel. Esto es administrado por un sistema ya elaborado para las tarjetas prepagadas de voz.
- Para las tarjetas prepagadas hay servicios administrativos que permitan asignar tarjetas a los agentes de ventas, imprimir cuentas en las tarjetas, consultar y tarificar. Como se mencionó en el punto anterior, se integró al sistema administrativo de tarjetas prepagadas para voz [Von Hauske, 2003] [Balboa, 2003]
- Un proceso automatizado para monitoreo de saldos que permite administrar los saldos y cortar la llamada a los usuarios que tengan un saldo agotado. [Von Hauske, 2003] [Balboa, 2003]
- Se le propuso a la empresa de telecomunicaciones permitir el acceso solamente a Marcatel cuando su saldo haya expirado. Marcatel investiga la viabilidad de esta sugerencia y en caso de desarrollarse, será una modificación posterior a la entrega de esta versión. [Von Hauske, 2003]

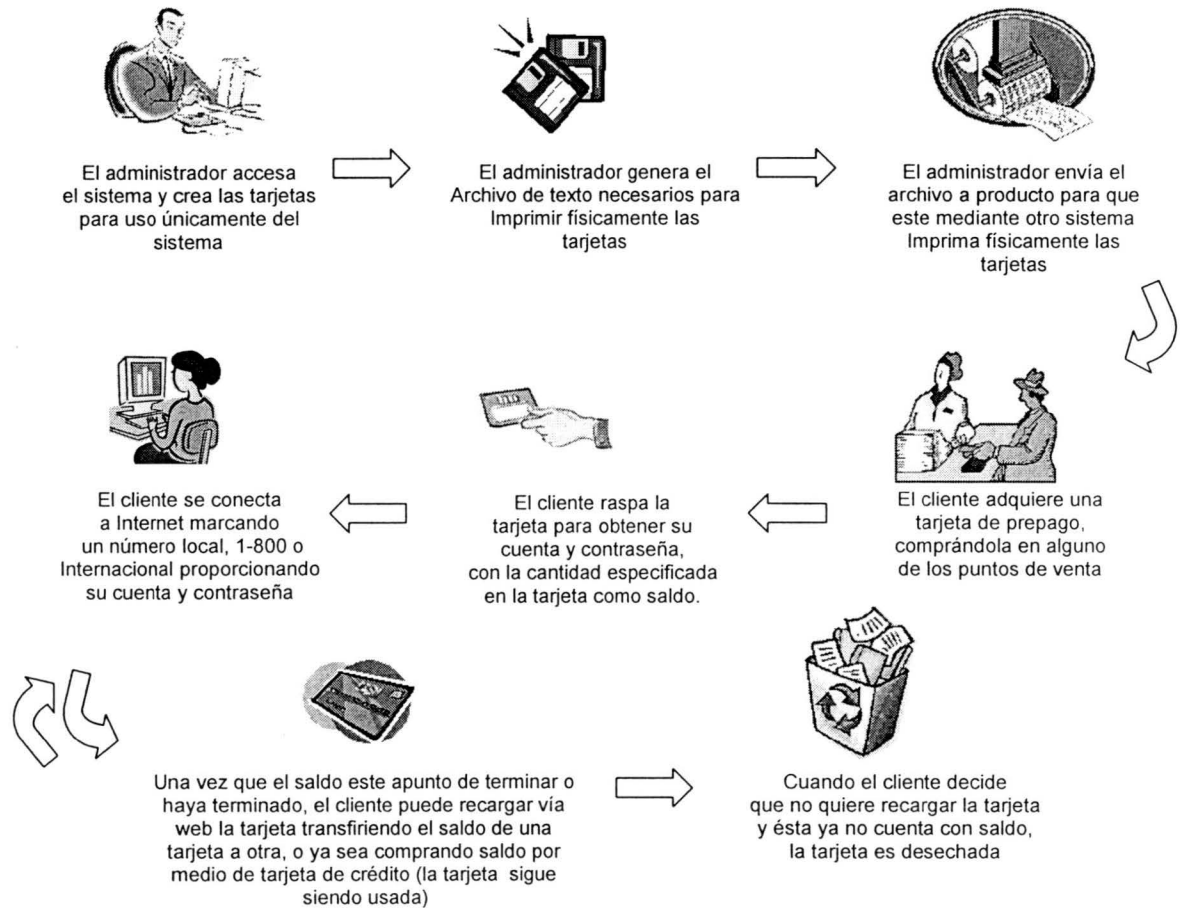
Proceso de Negocio Propuesto

Figura 2.2 Solución Propuesta para Tarjetas Prepagadas Internet

Los indicadores que Marcatel busca mejorar son los siguientes:

- Aumento de ventas en el servicio de Internet por vía telefónica.
- Mejor satisfacción de las necesidades del cliente.

Los indicadores que este proyecto busca mejorar son los siguientes:

- Seguridad en el acceso a los servicios de Internet mediante Radius (sistema encargado que autoriza a los usuarios para el uso de Internet vía telefónica)
- Rapidez en el cambio de tarifas, actualización de tarjetas y sus saldos.
- Buena administración de las tarjetas prepagadas.
- Uso eficiente de tecnologías de información bajo cargas pesadas al conectarse.

2.3 Alcances del Proyecto

Este proyecto fue desarrollo e integración a procesos implementados en sistemas actuales de Marcatel que sirven para su servicio de tarjetas prepagadas telefónicas. Entre estos se encuentra lo siguiente:

- La generación de tarjetas se integró a un proceso ya hecho para crear tarjetas de voz. No hubo desarrollo de ningún módulo para la impresión o administración de las tarjetas, sino una integración de este proceso con prepago Internet, su finalidad es hacer la tarjeta para uso de voz e Internet en un futuro. [Von Hauske, 2003]
- La generación de cuentas de Internet para las tarjetas se basó en un sistema que Marcatel tiene implementado para sus tarjetas de voz, es decir se generan las cuentas de las tarjetas para uso de Internet. [Von Hauske, 2003]
- El *log* de transacciones almacena únicamente información de los resultados de las validaciones de tarjeta (autorizaciones fallidas, compras de tarjetas y recargas) No se desarrollará ningún módulo o interfase para administrar dicha información.
- Las tarifas a utilizar ya están establecidas en otro sistema en base a los productos que tienen para Internet y voz, se van a integrar con el desarrollo de las tarjetas prepagadas para Internet para descontar el saldo necesario. [Von Hauske, 2003]
- Al integrar en un solo sistema los procesos de generación de tarjetas para voz e Internet, se pudieron realizar los siguientes reportes:
 - Tarjetas de prepago (solo tarjetas físicas)
 - Agentes de ventas (no incluye ventas en Internet)
 - Listado Cuentas
 - Reporte de ventas (no incluye Internet)

[Von Hauske, 2003]

2.4 Objetivo General

Elaboración de un sistema con herramientas Visual Basic .NET, SQL Server 2000 y Active Server Pages que permita la operación y administración del servicio de tarjetas prepagadas con infraestructura Radius y TNT como proveedor de Internet. El sistema debe tener aplicaciones para poder administrar las tarjetas prepagadas por medio de los empleados de Marcatel. El cliente puede consultar o y administrar (transferir y recargar saldos) sus tarjetas mediante una serie de aplicaciones.

2.5 Objetivos Particulares

El Administrador del sistema puede:

- Crear cuentas válidas en el servidor de dial-up, una por tarjeta.
- Borrar las cuentas vencidas.
- Deshabilitar cuentas por cuestión de seguridad.
- Desconectar el cliente cuando lo requiera.
- Establecer diferentes tarifas para los diferentes tipos de conexión.
- Editar las tarifas existentes.
- Adaptar el sistema al proceso de generación de folios de tarjetas.
- Asignar bloques de tarjetas a un proveedor.
- Administrar catálogos de tarjetas.
- Administrar los saldos de las tarjetas.
- Administrar catálogos de proveedores.
- Hacer consultas sobre los proveedores.
- Generar los archivos para imprimir las tarjetas.
- Guardar los archivos generados.
- Realizar la conciliación de la información en cuanto a ventas y pagos de tarjetas.
- Generar cualquiera de los reportes establecidos:
- Hacer consultas sobre la información de ventas.
- Enviar la información al sistema de facturación.

El cliente que usa el sistema puede:

- Consultar la página de recarga de tarjeta, para recargar su tarjeta mediante tarjeta de crédito.
- Consultar la página de recarga de tarjeta, para recargar su tarjeta mediante transferencia de saldo de una tarjeta a otra.
- Recargar su tarjeta vía Web por medio de tarjeta de crédito.
- Recargar su tarjeta vía Web por traspaso de saldo de tarjeta a tarjeta.
- Realizar la consulta del saldo de su tarjeta vía Web.

El Sistema debe:

- Valida los clientes que intenten conectarse, y a sea permitiéndole el acceso o negándosele.
- Monitorea las cuentas periódicamente, revisando los saldos, desconectando a los clientes cuyo saldo se haya terminado.
- Envía un mensaje al cliente cuando su saldo esté a punto de vencer o terminarse, esto previo a la instalación de un módulo cliente opcional.
- Interfase con el banco para pagar con tarjeta.

2.6 Lo que han hecho otras organizaciones

A continuación se analizan dos compañías que tienen este producto en el mercado, sus coberturas, tarifas y una comparación contra la solución propuesta en este documento.

2.6.1 Todito.com – Todito Card

La empresa Todito.com sacó al mercado una tarjeta de prepago para el uso de Internet para pagar solamente las horas que se utilizan por el usuario. Una de las ventajas de hacer este tipo de tarjeta es evitar el pago de mensualidades, elaboración de contratos que te pueden forzar a estar con una compañía por varios meses y configuraciones por CD que pueden maltratar la configuración personal de la computadora del usuario. [Todito.com, 2003]

Esta misma tarjeta cuenta con instrucciones para como activarla o el cliente puede llamar a un centro de atención donde se le asiste para la activación. [Todito.com, 2003]

Esta tarjeta se puede encontrar en más de 50 ciudades alrededor de la República Mexicana y distintos puntos de venta. La tarjeta cuenta con los siguientes precios: [Todito.com, 2003]

- \$100 pesos - 15 horas de navegación
- \$150 pesos - 30 horas de navegación
- \$300 pesos - 90 horas de navegación

El saldo se pueda consultar mediante la página www.toditocard.com, en este mismo sitio se puede volver a activar una tarjeta que tenga su saldo agotado. [Todito.com, 2003]

2.6.2 Terra Prepago

Terra prepago es el servicio de acceso a Internet sin el pago de mensualidades fijas. Este producto solo requiere comprar una tarjeta que contiene instrucciones de cómo conectarse y utilizar los servicios de Internet. Esta tarjeta se puede comprar en tiendas de autoservicio o en línea con una tarjeta de crédito. Esta empresa maneja dos tipos de horarios para hacer contabilizar su tarifa: [Terra, 2003]

- Horario Especial: de 9p.m. a 7a.m. por cada 3 minutos utilizados se deduce 1 minuto. [Terra, 2003]
- Horario Normal: de 7a.m. a 9p.m. tarifa normal por minuto, no hay promociones. [Terra, 2003]

La tarjeta cuenta con los siguientes precios:

Costo	Tiempo de conexión horario normal.	Tiempo de conexión horario especial.
\$90	750 minutos	2,250 minutos
\$170	1,500 minutos	4,500 minutos
\$290	4,000 minutos	12,000 minutos

Tabla 2.1 Tarifas de Terra Prepago

[Terra, 2003]

2.6.3 Comparación de productos existentes con el propuesto

Una de las ventajas del producto que ofrece Marcatel es su uso a nivel nacional e internacional a través de números 01-800 ó 1-800 respectivamente. Este servicio se podrá utilizar para su uso con el producto ya existente llamado Dial-up, que incluye productos como correo electrónico y espacio para publicar páginas en Internet.

Las funciones que tendrá el cliente (mencionadas en los objetivos particulares) tendrán la ventaja de recargar saldos con tarjeta de crédito o con otra tarjeta prepago. En la siguiente tabla se califica cada objetivo:

Objetivo	PESO	Terra Prepago		Todito.com		MarcaTel	
		Calificación	SUMA	Calificación	SUMA	Calificación	SUMA
Consultar la página de recarga de tarjeta	5%	9	4.50%	9	4.50%	9	4.50%
Recargar su tarjeta vía Web por medio de tarjeta de crédito.	15%	9	13.50%	9	13.50%	9	13.50%
Recargar su tarjeta vía Web por traspaso de saldo de tarjeta a tarjeta.	15%	0	0.00%	0	0.00%	10	15.00%
Realizar la consulta del saldo de su tarjeta vía Web.	15%	10	15.00%	9	13.50%	10	15.00%
Validará los clientes que intenten conectarse, ya sea permitiéndole el acceso o negándoselo.	10%	10	10.00%	9	9.00%	10	10.00%
Monitorear las cuentas periódicamente, revisando los saldos, desconectando a los clientes cuyo saldo se haya terminado.	10%	9.5	9.50%	9	9.00%	9	9.00%
Enviar un mensaje al cliente cuando su saldo esté a punto de vencer o terminarse, esto previo a la instalación de un módulo cliente opcional.	10%	0	0.00%	0	0.00%	6	6.00%
Interfase con el banco para pagar con tarjeta.	20%	10	20.00%	10	20.00%	10	20.00%
TOTAL	100%	7.1875	72.50%	6.875	69.50%	9.125	93.00%

Tabla 2.2 Comparación de Productos

2.7 Importancia del Proyecto

Este producto tiene como objetivo hacer que Marcatel sea más competitivo en el área de servicios de Internet por medio telefónico (Dial Up) debido a la fuerte competencia que presentan otras compañías dentro de este mercado. [Balboa, 2003] En un futuro este producto de tarjetas de prepago Internet se fusionará con las tarjetas de prepago telefónicas. El cliente no sólo tendrá la opción de hablar por teléfono, sino también utilizar Internet, todo en la misma tarjeta. El proyecto es prioritario a Marcatel. [Von Hauske, 2003]

2.8 Resumen y Conclusión del Capítulo

El capítulo presentó el producto que esta empresa quiere lanzar al mercado de prepago de Internet. Se definieron los objetivos generales y particulares del proyecto para la operación del producto.

El uso de Internet prepago es otra opción para el cliente para obtener acceso a Internet y tener un control del uso mediante un saldo. Esto permite a los clientes utilizar su tarjeta en las áreas en las que Marcatel provee su servicio.

3. Marco Teórico

Este capítulo explica las herramientas y metodología a usar en el proyecto. También se hace una comparación con otros productos similares al desarrollado.

3.1 Plataforma Microsoft .NET

El ambiente .Net es un conjunto de tecnologías para conectar información, gente y dispositivos. Permite un nivel de integración a través del uso de XML (Extensible Markup Language). La plataforma .NET permite acceder la información en todo momento, lugar y desde cualquier dispositivo. Esto permite simplificar el desarrollo y darle oportunidades nuevas a los negocios. La plataforma permite reunir varias tecnologías que pertenecen a .NET para, junto con los conocimientos del usuario, para formar una base de productos integrados unos a otros utilizando estándares de Internet abiertos. [Platt, 2001]

3.1.1 Herramientas Microsoft Visual Studio .NET y Framework

A continuación se explica el ambiente de *Visual Studio .NET*, el *Framework* y sus herramientas *VB.NET*, *ASP.NET* y *Web Services*.

3.1.1.1 .NET Framework

Visual Studio .NET automatiza a gran escala una aplicación a un servicio *Web XML*. Esto reduce el tiempo que utiliza el desarrollador para poder liberar soluciones. *NET Framework* provee lo necesario para asegurar que ninguna aplicación sea compartida con otra sin autorización de autor. Permite la integración segura de aplicaciones bajo una sola plataforma en sistemas operativos *Windows*. Esto permite evitar la instalación de varias librerías (llamadas *DLL*) para su operación adecuada. [Platt, 2001]

El Framework es un ambiente que permite a los programadores administrar fácilmente su código. Los componentes que se construyan pueden ser utilizados dentro del ambiente. Este ambiente cuenta con su propio administrador de memoria llamado "coleccionador de basura" o *garbage collector*. Los sistemas de servicio son más fáciles de acceder. Cuenta con más opciones de uso para el Internet y sistemas manejadores de bases de datos y un mecanismo más amigable para reutilizar código, más accesible que del ambiente COM que utiliza Visual Studio 6.0. [Platt, 2001]

"El ambiente *.NET Framework* proporciona soluciones prefabricadas a los problemas de programación." [Platt,2001] El ambiente se ejecuta a través de un ambiente llamado ambiente de lenguajes comunes o *Common Language Runtime (CLR)*. La siguiente figura muestra como sucede una ejecución de programa en el ambiente *CLR*:

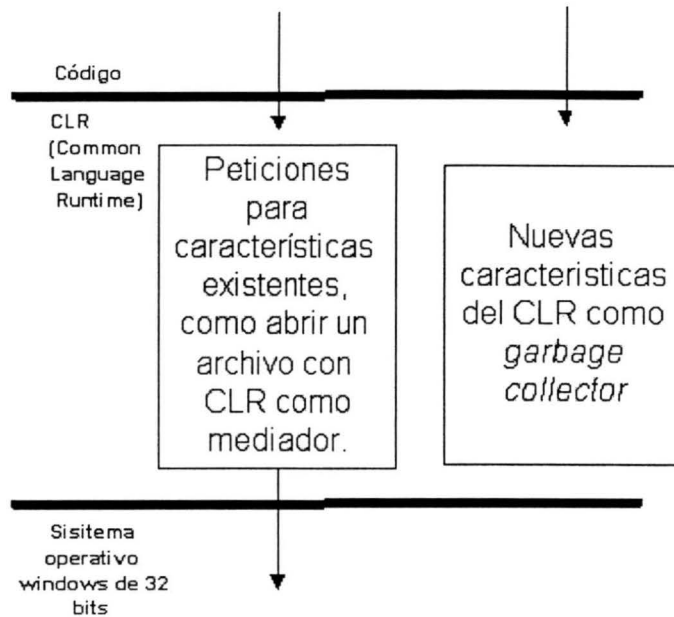


Figura 3.1 Ambiente CLR Common Language Runtime [Platt,2001]

El CLR bajo esta arquitectura no es capaz de soportar todos los lenguajes posibles, para esto se tiene el lenguaje intermedio de Microsoft o *Microsoft Intermediate Language (MSIL)*. Cada aplicación dentro de Visual Studio se convierte a código intermedio a través del *MSIL* y finalmente en su implementación se utiliza el CLR. La siguiente figura muestra el proceso de compilación para MSIL:



Figura 3.2 Código Intermedio de Microsoft [Platt,2001]

El código intermedio no puede ejecutarse directamente en cualquier computadora. Se requiere un segundo paso que es el compilador "justo a tiempo" o *Just-In-Time (JIT)*. El JIT convierte el MSIL en código para la plataforma específica, permitiéndole ser más independiente. [Platt,2001]

3.1.1.2 Visual Basic .NET

Visual Basic.NET es un lenguaje para crear aplicaciones. Liberado después del décimo aniversario de la primera versión del lenguaje, *Microsoft* tuvo que rediseñar el lenguaje antiguo debido a que no cumplía con los objetivos del paquete *.NET*. [Pertuosos, 2002]

Al abrir la aplicación de *Visual Studio.NET* se comienza en la página inicial de la aplicación:

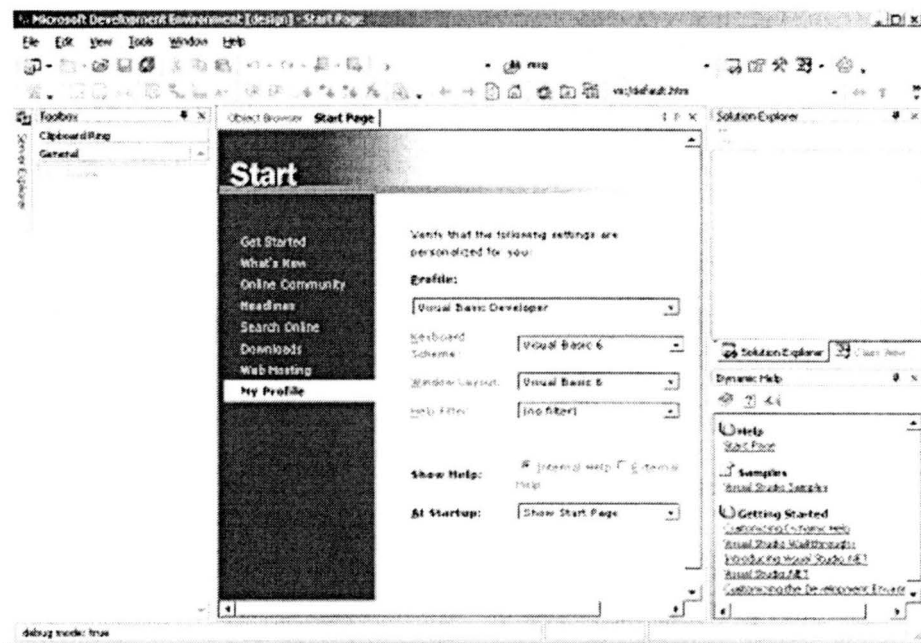


Figura 3.3 Página inicial de Visual Studio.NET [Pertuosos, 2002]

Una vez en esta página se selecciona del menú archivo, nuevo, proyecto y aparecerá una lista de proyectos que se pueden crear:

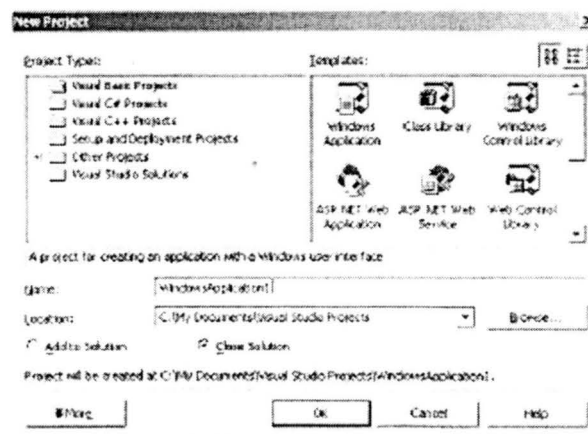


Figura 3.4 Ventana para crear nuevo proyecto [Pertuosos, 2002]

El formato de *Windows Application* es el adecuado para utilizar las herramientas de Visual Basic.NET. Una vez seleccionado este formato se define el nombre de la aplicación y donde se va guardar el proyecto. [Pertuosos, 2002]

Al oprimir el botón *OK* aparecerá la pantalla de desarrollo:

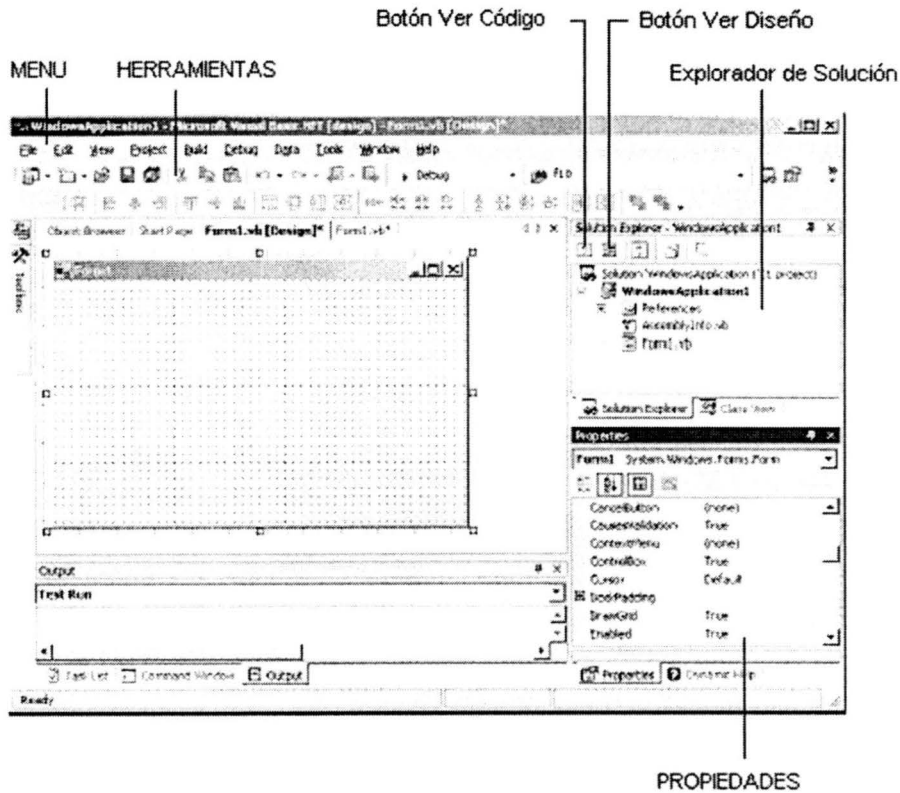


Figura 3.5 Ventana de desarrollo VB.NET [Pertuosos, 2002]

Igual como Visual Basic 6.0, se agregan componentes como botones o cajas de texto desde la barra de herramientas que se encuentra del lado izquierdo de la pantalla (botón se encuentra en forma vertical). Una vez que se ingresa a ese botón de herramientas aparece el siguiente menú con diferentes componentes:



Figura 3.6 Ventana de componentes para VB.NET [Pertuosos, 2002]

Una vez que el componente se selecciona y agrega a la pantalla de diseño se le pueden asignar propiedades al mismo (ver Figura 3.3), como ejemplos de propiedades están el color de fondo o color del texto. [Pertuosos, 2002]

Un ejemplo simple de Visual Basic.NET sería como se presenta a continuación:

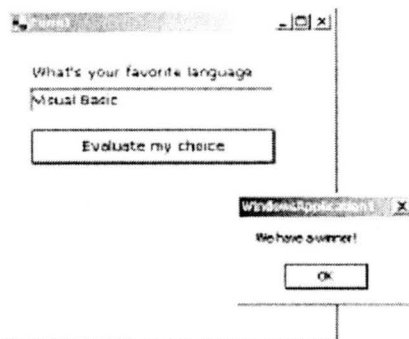


Figura 3.7 Ejemplo simple para VB.NET [Pertuosos, 2002]

El objetivo a realizar en la aplicación mostrada en la figura 3.5 será preguntar cual es tu lenguaje favorito, contestar en la caja de texto “Visual Basic” y presionar el botón para recibir la respuesta “Tenemos un Ganador”. En caso de no ingresar en la caja de texto “Visual Basic” indicar que ese lenguaje no es muy malo. El primer paso es dar doble click al componente del botón para ingresar a la ventana de código. Una vez aquí darle instrucciones al botón para dar la respuesta adecuada según lo ingresado en la caja de texto. La subrutina es creada automáticamente al darle doble click al componente. A continuación se muestra esta pantalla:

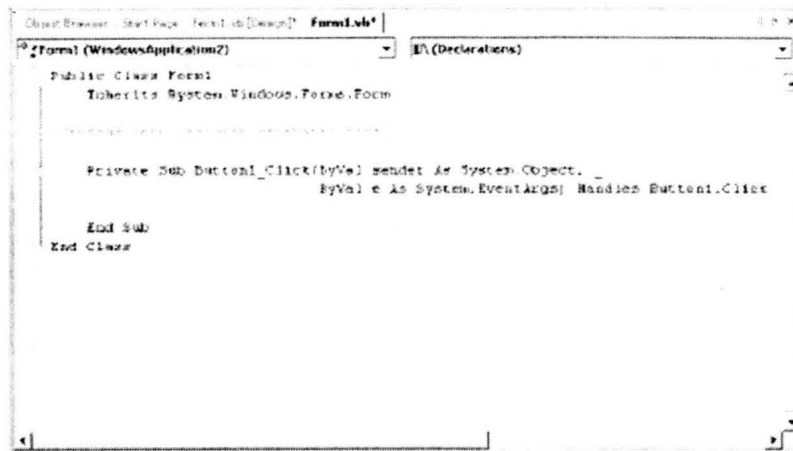


Figura 3.8 Pantalla de código [Pertuosos, 2002]

La subrutina que se creó es la siguiente:

```

Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
End Sub

```

En la subrutina se ingresará lo necesario para que se dé la respuesta necesaria para cumplir el objetivo de la aplicación a realizar:

```

Private Sub EvaluateLanguage_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    Dim language As String
    lenguaje = TextBox1.Text
    If lenguaje = "Visual Basic" Then
        MsgBox("Tenemos un Ganador!")
    Else
        MsgBox(lenguaje & " no es un lenguaje malo.")
    End If
End Sub

```

[Pertuosos, 2002]

El código que se agregó asigna lo capturado en la caja de texto (Textbox1.Text) a una variable llamada "lenguaje", se revisara en una condición If si la variable "lenguaje" es igual a "Visual Basic", si es así se mostrará un mensaje que diga "Tenemos un Ganador", en cambio si es cualquier otra serie de palabras (Ejemplo: "Pascal") se muestra el mensaje "Pascal no es un lenguaje malo". Esto se realiza con el operador & que permite juntar cadenas de texto (llamadas *strings*) tanto de variables o entre comillas. [Pertuosos, 2002]

Una vez terminado esto, se corre la aplicación para ver la página final, esto se puede hacer seleccionado del menú *Debug* la opción de *Start*. [Pertuosos, 2002]

3.1.1.3 ASP .NET

La mayoría de los programas simples creados se basará en Internet y utilizará un navegador para ser utilizada. Los programas requieren de un servidor que opere éstas aplicaciones. ASP.NET es una nueva manera de construir código y probarlo en un ambiente de Internet. Contiene el uso de forma de página o *web forms* que permite al programador utilizar un ambiente orientado a eventos para diseñar y construir su aplicación para navegadores de Internet. El ambiente de desarrollo es muy parecido al uso de Visual Basic 6.0 o .NET. Es más robusto y contiene una serie de mejoras en comparación al ASP original. [Platt, 2001]

ASP es un tipo de página dinámica que funciona mediante el Internet Information Server de Microsoft para mostrar contenido dinámico en las páginas Web mediante código denominado *VBScript (Visual Basic Script)*. Este mismo corre en un servidor y regresa los datos necesarios a la interfaz del cliente que esta mostrado y diseñado en *HTML (Hyper Text Markup Language)*. [Platt, 2001] [Reselman, 2001]

El *ASP.NET* permite al desarrollador separar el código de *HTML*, que es utilizado para diseñar y mostrar información, del código dinámico permitiendo al mismo tener más orden en su código. Las herramientas para *ASP.NET* se han vuelto más visuales y cuentan con soporte para ejecutar el código línea por línea en busca de errores, fallas o procesos repetitivos. [Platt, 2001]

Para desarrollar un ejemplo simple bajo *ASP.NET* se siguen pasos muy parecidos al *Visual Basic.NET*. El primer paso es crear un nuevo proyecto de la misma manera que *VB.NET*, pero en los formatos predeterminados (ver Figura 3.2) se selecciona "*ASP.NET Web Application*".

Una vez que el proyecto está creado aparece una ventana de desarrollo como se muestra en la siguiente figura:

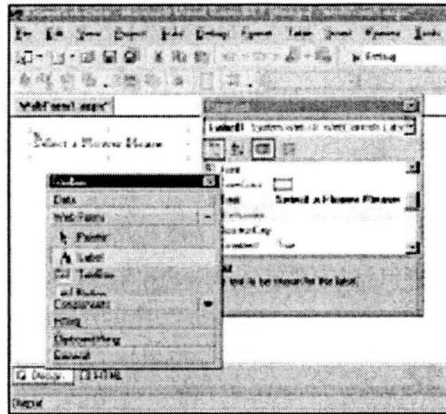


Figura 3.9 Pantalla de diseño ASP.NET [Reselman, 2001]

En este caso, el ASP.NET se divide en dos partes y una de estas en otras dos partes:

Nombre de Pagina: "X"

- Pagina web en HTML y ASP
 - Ventana de diseño: Vista gráfica de la página que se esta desarrollando.
 - Ventana HTML: Vista a nivel código de la pagina que se esta desarrollando.
- Administrador de Eventos de pagina "X" (code behind)

[Reselman, 2001]

El objetivo del ejemplo simple a desarrollar es crear una página que tenga una lista que inserte ciertos valores al momento de cargar la página y mostrar debajo un mensaje que diga que flor se seleccionó de la lista. El primer paso es seleccionar de la lista de herramientas (ver Figura 3.4) el componente de lista (*listbox*) y traerlo a la página de diseño gráfico. En la ventana de propiedades (Ver Figura 3.3) seleccionar los colores al gusto del usuario y darle el nombre (*ID*) "*lstflores*". Una vez completada la configuración de propiedades del componente se da doble click a la página (no al componente) esto trae la ventana de *code behind* de la página que se muestra en la siguiente figura: [Reselman, 2001]

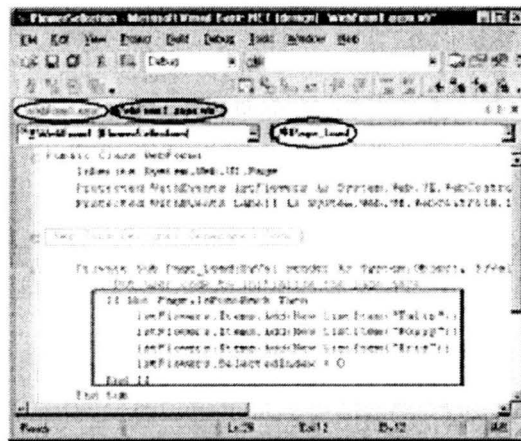


Figura 3.10 Pantalla de código ASP.NET [ReselMAN, 2001]

Al observar el código se nota que es idéntico al VB.NET, en la subrutina de "Page Load" se agrega el código necesario para llenar la lista como se muestra a continuación:

```
Private Sub Page_Load (By Val sender as System.Object ..... )
    "Poner código aquí para inicializar la página
    If Not Page.isPostBack Then
        LstFlores.Items.Add(New ListItem("Rosa"))
        LstFlores.Items.Add(New ListItem("Tulipan"))
        LstFlores.Items.Add(New ListItem("Girasol"))
    End If
End Sub
```

[Reselman, 2001]

Con este código se agregaran tres elementos a la lista y a asegurar que se haga una sola vez al iniciar la página (*IsPostBack*). Ahora hay que regresar a la página de diseño y agregar el componente *label* y darle en propiedades el nombre "*lblmensaje*". Después, se da doble click al componente de lista, aparecerá un código parecido al anterior con el nombre "*lstflores_SelectedIndexChanged*" y hay que agregar lo siguiente:

```
lblmensaje.Text="Haz Seleccionado: " + lstFlores.SelectedItem.Text
```

[Reselman, 2001]

Esto permitirá que el *label* agregado debajo de la lista muestre el mensaje "Haz Seleccionado" más el elemento que se seleccionó de la lista.

Una vez terminado esto, se corre la aplicación para ver la página final, esto se puede hacer seleccionando del menú "*Debug*" la opción de "*Start*". [Reselman, 2001]

Una vez presionado el botón *Start* la aplicación se compila a través del .NET Framework y muestra una página de Internet en el navegador. Aquí se podrá utilizar la aplicación simple que se acaba de construir. [Reselman, 2001]

3.1.2 Microsoft SQL Server 2000

SQL 2000 (denominado *Structured Query Language*) es una de las más recientes generaciones desde la versión 7.0. Esta nueva versión suporta procesamiento analítico en línea (OLAP) mejorado, soporte para plataformas 2000 en adelante. Este mismo es uno de los primeros productos disponibles para la plataforma .NET. [Patton, 2001] [Reselman, 2001] [Delgado, 2001] Es un manejador de bases de datos que ofrece los siguientes servicios:

- Almacenamiento y consulta de de datos: mediante la herramienta llamada “Analizador de consultas” (*Query Analyzer*) con lenguaje SQL (*Structured Query Language*) se pueden hacer consultas, almacenamientos y administrar tablas.

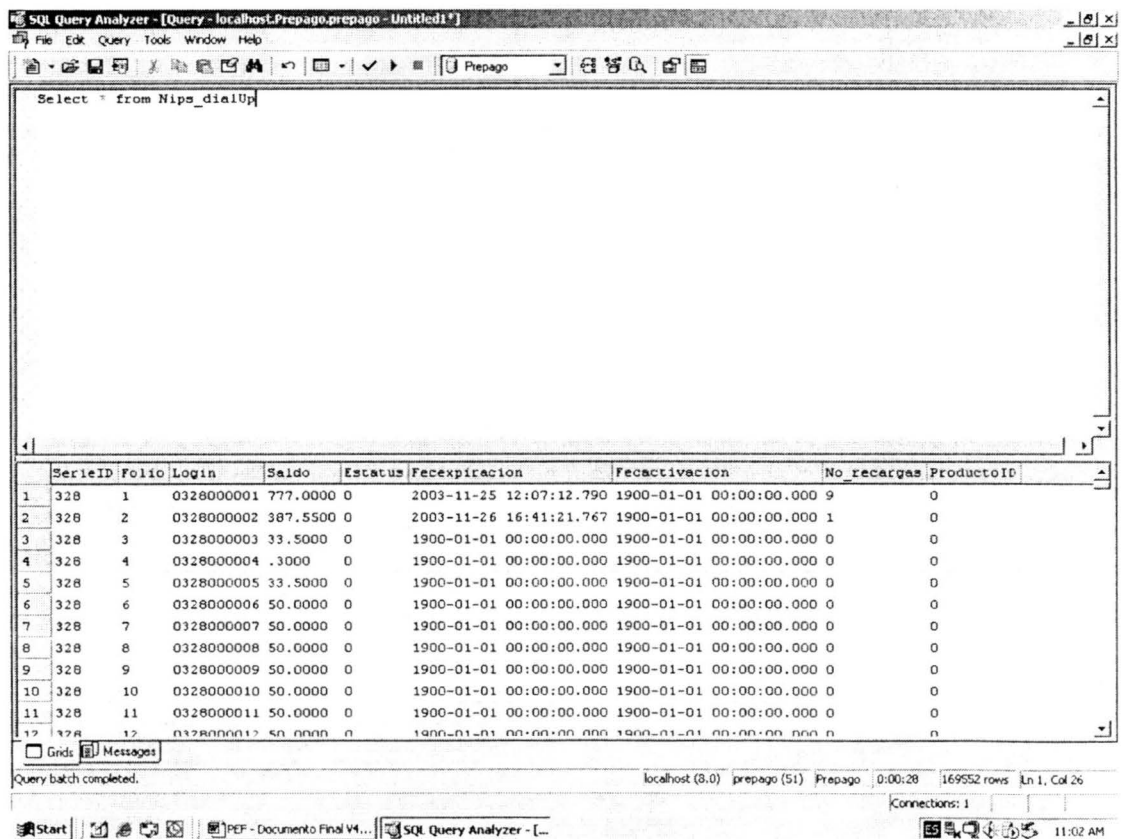


Figura 3.11 Pantalla de Query Analyzer

- Stored procedures: son funciones que contienen código estructurado y SQL combinado para dar un resultado o hacer una tarea específica.

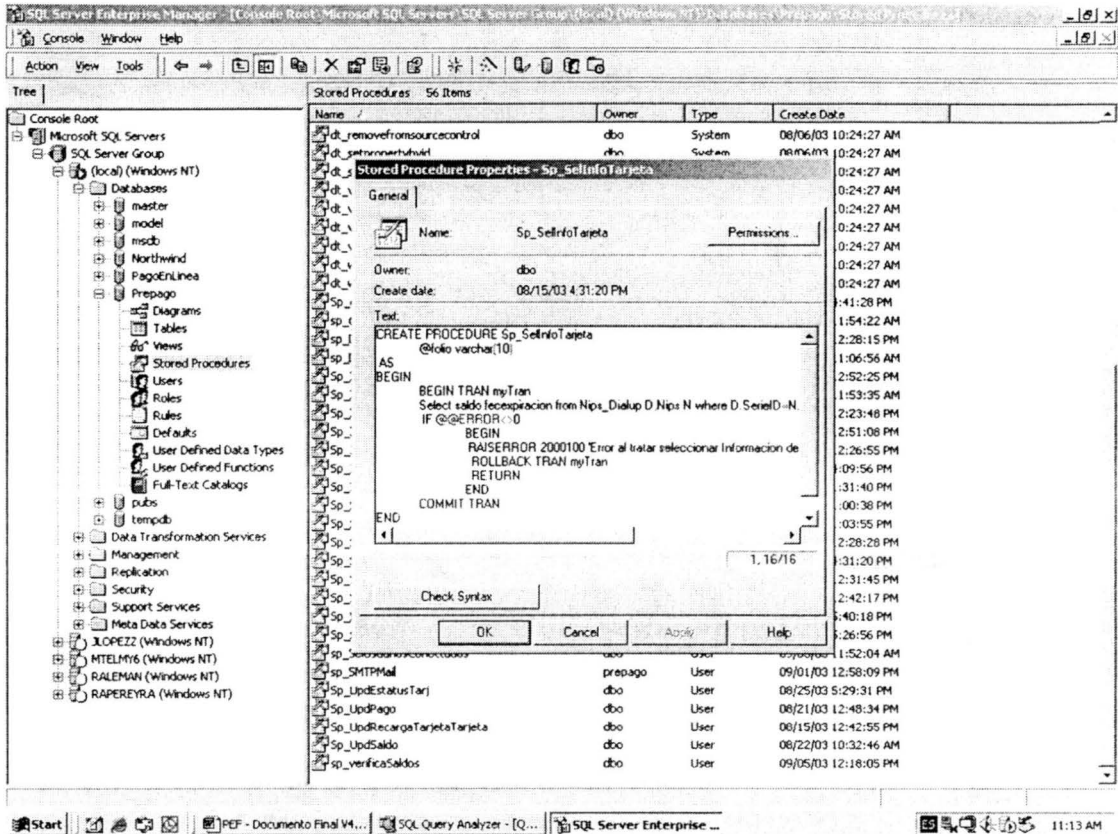


Figura 3.12 Pantalla de *Stored procedures*

- Administración del manejador: Se tiene una aplicación llamada *Enterprise Manager* que es utilizada para administrar las bases de datos que se encuentran en el servidor, acceso de usuarios y tareas que el analizador de consultas hace pero en forma gráfica.

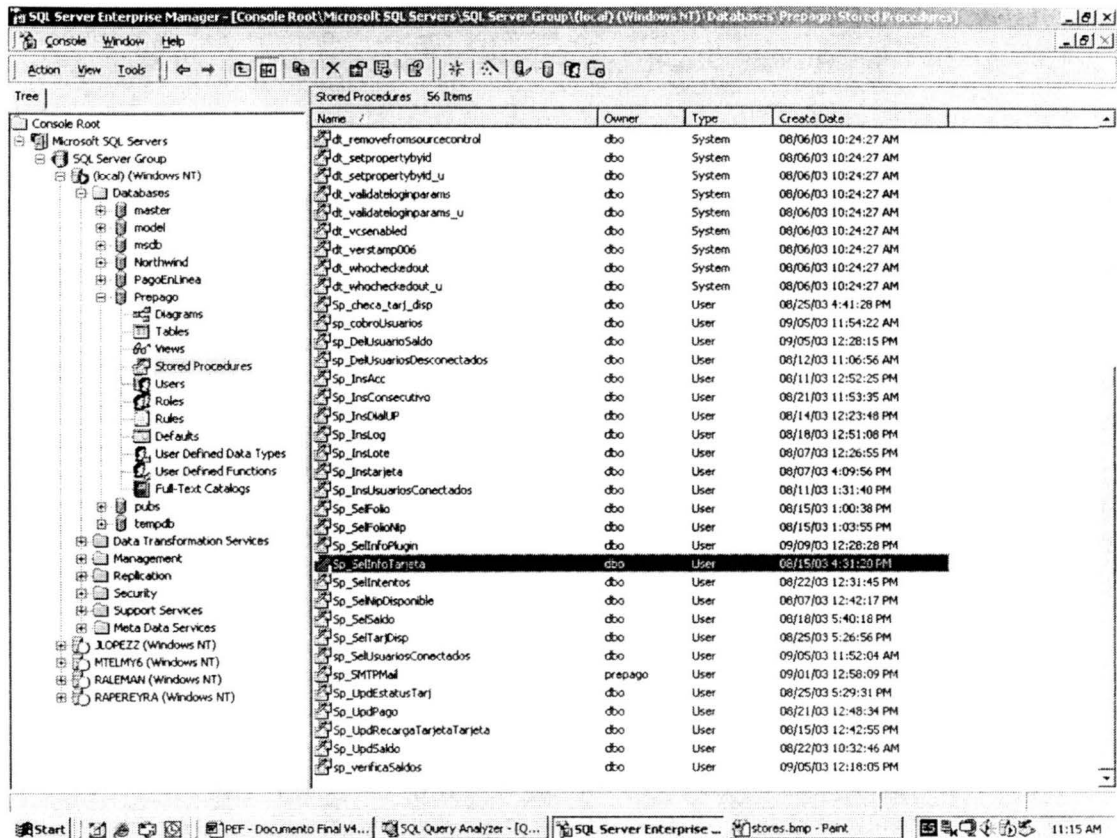


Figura 3.13 Pantalla de Enterprise Manager

- **Servicios:** Denominados como *jobs*, se utilizan para llamar tareas a realizar por el manejador de base de datos. Esto se mide en unidades de tiempo por fecha y hora. Un ejemplo de tareas a realizar son los *stored procedures* que se encuentran en la base de datos. [Delgado, 2001]

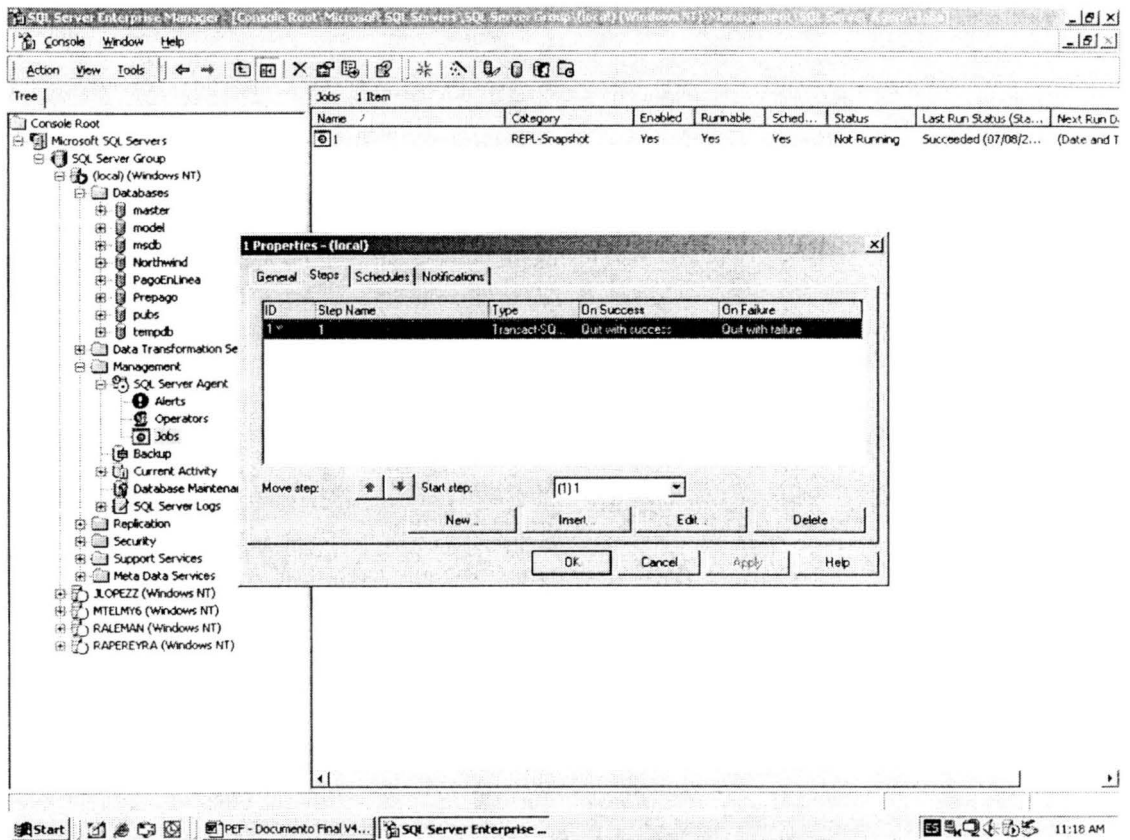


Figura 3.14 Pantalla de Jobs

Esta nueva versión de SQL es un refinamiento de los servicios para desarrolladores de aplicaciones. [Patton, 2001] [Reselman, 2001] [Delgado, 2001]

3.2 Equipo Lucent y Cisco

3.2.1 Cisco ACS Raduis

Cisco ACS Radius permite a un proveedor de Internet administrar su servicio y observar las operaciones mediante un manejador de base de datos compatible con el lenguaje SQL. Este producto es compatible con equipo de acceso remoto como el MaxTNT de Lucent, que será definido más adelante. RADIUS es una forma económica para administrar el servicio de Internet y su manejador de bases de datos permite acceso más rápido a la información mediante otras aplicaciones. [Cisco ACS Radius, 2003]

Este producto es una implementación del protocolo RADIUS que provee servicios de autorización, verificación y contabilidad de proveedores de servicios, en este caso de acceso para Internet. El protocolo es un estándar para la autorización de conexiones vía remota. El paquete se

compone 32 bits que permite hacer peticiones de autorización externas a la red con seguridad. [HIII, 2001] El siguiente diagrama muestra la estructura del paquete:

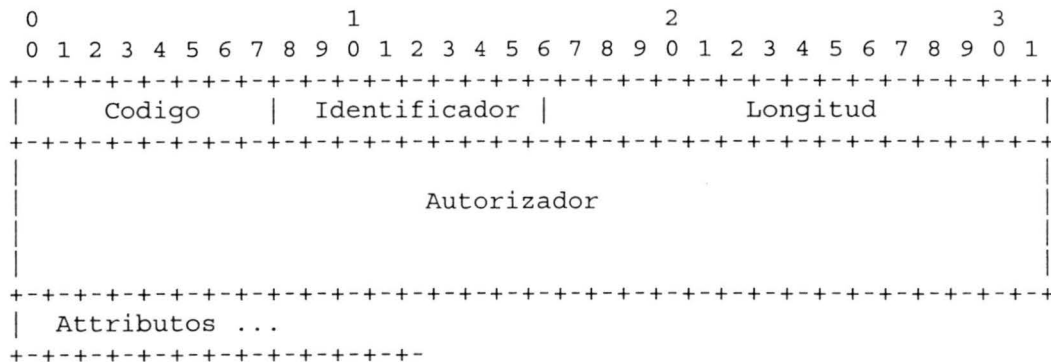


Figura 3.15 Paquete de protocolo Radius [HIII, 2001]

Este paquete de 32 bits se divide en tres tipos:

Paquete de código, identificador y longitud (32 bits totales)

Este representa el paquete inicial del protocolo Radius. El código define que tipo de paquete se enviará, la siguiente tabla muestra los diferentes tipos de paquetes:

Valor	Descripción
1	Petición para Acceso
2	Aceptación de Acceso
3	Negación de Acceso
4	Petición Administrativa
5	Respuesta Administrativa
11	Desafío de acceso
12	Estado de Servidor (experimental)
13	Estado de Cliente (experimental)
255	Reservado

Tabla 3.1 Valor y descripción de códigos de paquete Radius. [HIII, 2001]

El identificador sirve para juntar la respuesta con la petición que hizo el cliente. La longitud es el número de bits a transmitir con los datos necesarios de la petición o respuesta. [HIII, 2001]

Paquete de código autorizador (32 bits totales)

El código autorizador sirve para verificar si el paquete que se esta enviando es auténtico o verdadero, esto para efectos de seguridad. [Hill, 2001]

Atributos (32 bits totales)

Los atributos representan los datos a autorizar como el nombre de usuario y contraseña. [Hill, 2001]

3.2.2 MAX TNT Universal Gateway

Este es un dispositivo que permite conectar a usuarios a Internet que llamen por vía telefónica. Puede manejar hasta 720 llamadas de forma concurrente sobre varios tipos de enlaces, estos pueden ser: [Lucent, 2003]

- Línea análoga (Teléfono) [Lucent, 2003]
- Línea *ISDN* (enlace con ancho de banda de 128 kilo bits por segundo) [Lucent, 2003]
- Línea *T1/E1/T3* (enlaces con ancho de banda entre 1.4 mega bits por segundo y 4.2 mega bits por segundo) [Lucent, 2003]

Este sistema funciona bajo un sistema operativo llamado Sistema Operativo *True Access* (denominado TAOS por sus siglas en inglés) que permite ser proveedor de acceso a Internet y la expansión en un futuro al mercado de telefonía IP. [Lucent, 2003]

El max TNT permite:

- Configuración de la infraestructura de red a las necesidades de ancho de banda y aplicaciones del usuario.
- Alto rendimiento en capacidades de hardware y software.
- Reducir el equipo de red a utilizar como *modems*, servidores para terminal, *routers* y líneas de acceso.
- Eliminar el alto costo de operación por uso de tecnología de *modem* análoga.

Características:

- Ancho de banda dinámica según la demanda.
- Conecta oficinas remotas y usuarios vía *dial-up* con servicios que provee una empresa.

- Su diseño modular permite enlaces con varios tipos de enlaces (mencionado anteriormente)
- Alta confiabilidad, permite cambio de dispositivos sin necesidad de apagar el hardware.
- Compatible con protocolo RADUIS para autorización y seguridad.

3.4 Metodología *Microsoft Solution Framework* aplicada según las necesidades de MarcaTel

A continuación se explica la Metodología de desarrollo *Microsoft Solution Framework* (*MSF por sus siglas*) esta dividida en cuatro fases: [Microsoft, 2003]

1- Estrategia y alcance:

Esta fase es fundamental, se definen los requerimientos y el equipo completo debe tener una idea clara de los objetivos a realizar. [Microsoft, 2003] [MSF, 2002]

2- Planificación y prueba de concepto

Cuando los requerimientos están aprobados, se pasa a la fase de planeación, donde se especifican las funciones del sistema a través de escenarios, el proceso de diseño, análisis de costo y beneficio y el plan de trabajo

[Microsoft, 2003] [MSF, 2002]

3- Desarrollo

En la fase de desarrollo se cumplen con la mayoría de los componentes a construir. En la siguiente fase será necesario hacer cambios en respuesta a pruebas sobre estos componentes.

[Microsoft, 2003] [MSF, 2002]

4- Estabilización

Esta fase permite hacer pruebas sobre los componentes construidos. Se realiza bajo condiciones reales de trabajo, se resuelven fallas y se prepara la liberación.

[Microsoft, 2003] [MSF, 2002]

5-Despliegue

La última fase se dedica a liberar los componentes construidos y corregidos durante la fase de liberación y obtiene aprobación del cliente.

[Microsoft, 2003] [MSF, 2002]

3.4.1 Introducción

Como introducción se repasan los ciclos de vida de proyecto que están basados en la metodología MSF.

3.4.1.1 El Modelo en Espiral y Cascada

Los ciclos de vida más populares de un proyecto de tecnologías de información han sido el proceso en cascada y el proceso en espiral. [MSF, 2002]

A continuación se presenta el modelo en cascada:

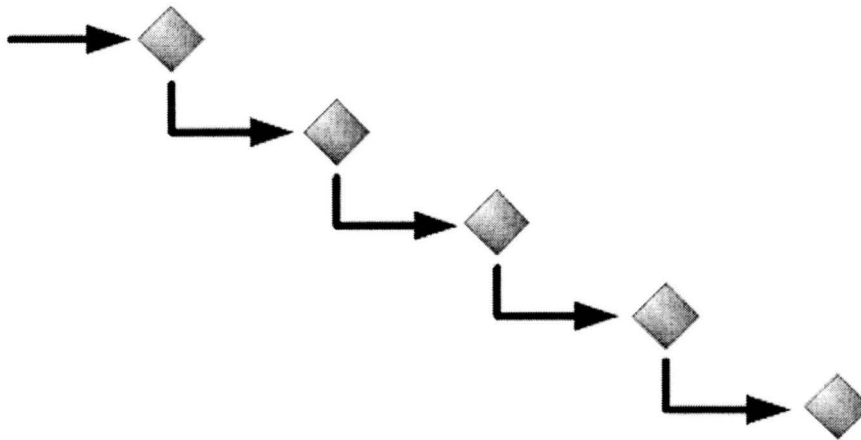


Figura 3.16 – Ciclo de Vida de Proyecto en Cascada. [MSF, 2002]

Este modelo usa metas para cumplir cada fase de un proyecto. Cada meta debe ser cumplida antes de pasar a la siguiente fase establecida. Este ciclo es muy usado para elaborar proyectos que no tiene muchos cambios en sus requerimientos. Las transiciones de meta a meta facilitan la asignación de responsabilidades y asignación de recursos humanos y económicos. [MSF, 2002]

A continuación se presenta el modelo en espiral:

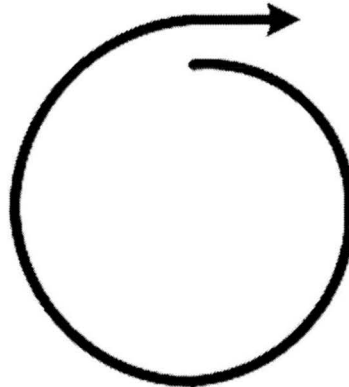


Figura 3.17 – Ciclo de Vida de Proyecto en Espiral [MSF, 2002]

Este modelo se enfoca principalmente en cambios continuos en los requerimientos y estimaciones en un proyecto. Este ciclo de vida es muy útil en un proyecto pequeño. Fue desarrollado en respuesta a críticas porque los ciclos de vida tradicionales no tienen prototipo, es decir revisiones constantes para mejorar la solución. [Behforooz,1996] Permite una fuerte relación entre el cliente y el desarrollador a través de retroalimentación constante por parte del cliente. Una desventaja del ciclo es su falta de metas durante el desarrollo que pueden causar que el proyecto sea problemático. [MSF, 2002]

3.4.1.2 El ciclo de vida de *MSF*

El ciclo de vida *MSF* ofrece una combinación entre el ciclo de vida de proyecto en cascada y espiral. Esto permite una mejor planeación del proyecto y retroalimentación por parte del cliente. A continuación se muestra el modelo del ciclo de vida para *MSF*:

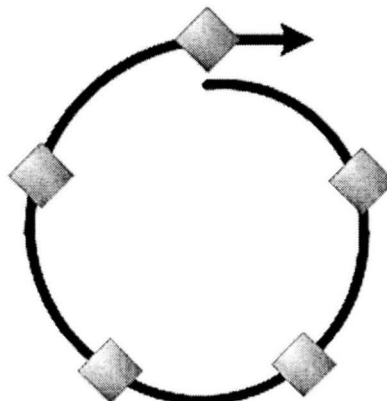


Figura 3.18 – Modelo de ciclo de vida para *MSF* [MSF, 2002]

3.4.1.3 Los principios que utiliza MSF

El modelo *MSF* está directamente asociado a los siguientes principios:

- **Tener una visión compartida:** Los integrantes del equipo de desarrollo y los clientes deben tener la misma visión del producto a desarrollar. Se dan a conocer las ideas de cómo esta actividad va beneficiar a la organización. Esto permite que todos los integrantes trabajen para cumplir la misma meta.
- **Ser ágil, esperar cambios:** Los ciclos de vida tradicionales establecen que los requerimientos han sido definidos claramente y no requieren de cambios. *MSF* espera que haya cambios durante el ciclo de vida del proyecto.
- **Enfoque en importancia empresarial:** Las soluciones exitosas, deben satisfacer las necesidades y beneficiar al cliente.

[MSF, 2002]

3.4.1.4 Conceptos a Conocer de *MSF*

A continuación se definen algunos conceptos para entender el ciclo de vida de la metodología. [MSF, 2002]

3.4.1.4.1 Clientes o Consumidores

La metodología *MSF* distingue entre el cliente y el usuario. Por ejemplo, para soluciones que involucran un producto de software, juegos, aplicaciones para uso en Internet o Intranet y cliente puede ser el mismo usuario. Para soluciones de negocio, el cliente es el individuo que patrocina y espera un beneficio del producto y los usuarios son los que interactúan con la solución desarrollada. [MSF, 2002]

La participación del cliente es fundamental para el éxito de la solución. *MSF* permite al usuario varias oportunidades para formar y modificar la solución con su retroalimentación. Estos clientes pueden ser internos o externos a la organización. [MSF, 2002]

3.4.1.4.2 Inversionistas

Son los individuos o grupos que tienen un interés del resultado del proyecto. Las metas que establecen no son siempre las mismas a las que establece el cliente. Cada inversionista tiene requerimientos importantes para la solución. [MSF, 2002]

3.4.1.4.3 ¿Qué es una solución?

Una solución es una estrategia o método que se utiliza para resolver un problema. Para MSF, una solución tiene un significado específico. “Es el envío coordinado de elementos necesitados (como tecnologías, documentación, entrenamiento y soporte) para responder satisfactoriamente al problema de negocio de un cliente específico.” MSF se enfoca en dar soluciones a un cliente en particular para desarrollar un producto comercial para un mercado masivo. [MSF, 2002]

En la siguiente tabla se muestra la diferencia entre el producto requerido y la solución a dar:

Producto	Solución para <i>MSF</i>
Diseñado par alas necesidades del Mercado masivo.	Diseñado para satisfacer las necesidades del cliente.
Enviado como un paquete de bienes o bits (CD-ROM, etc.)	Enviado como un proyecto.

Tabla 3.2 Comparación entre el producto y la solución MSF. [MSF, 2002]

3.4.1.4.4 Lineamientos

Los lineamientos son medidas o estados conocidos para comparar o medir algo. Establecer lineamientos es un proceso recurrente en la metodología MSF. Ejemplos de productos entregables durante un lineamiento son código fuente, configuración de servidores, horarios, especificaciones, manuales de usuario y presupuestos. [MSF, 2002]

3.4.1.4.5 Alcance

El alcance es la suma de todos los servicios y productos entregables para el proyecto. Debe establecer lo que debe tener y soportar la visión compartida. Integra la visión compartida contra la realidad y refleja lo que el cliente considera esencial para la liberación. [MSF, 2002]

Los beneficios de establecer alcances son:

- Dividir visiones a largo plazo en pedazos pequeños que se puedan cumplir.
- Definir las características que va tener cada versión final.
- Permitir una flexibilidad para cambios.
- Permitir un lineamiento para compensaciones de manejo.

[MSF, 2002]

El alcance tiene dos aspectos. El primero es el alcance de la solución que describe las características de la solución y sus entregables, incluyendo cosas que no involucren código fuente. El segundo es el alcance del proyecto que describe el trabajo a ser elaborado por el equipo para permitir la entrega de cada elemento descrito en el alcance de solución. [MSF, 2002]

3.4.2 Fases del Modelo *MSF* y Metas

MSF versión 3.0 integra los dos procesos ancestros del MSF, desarrollo de aplicaciones e infraestructura para liberación. La siguiente figura muestra el proceso del MSF sus metas y fases a seguir:

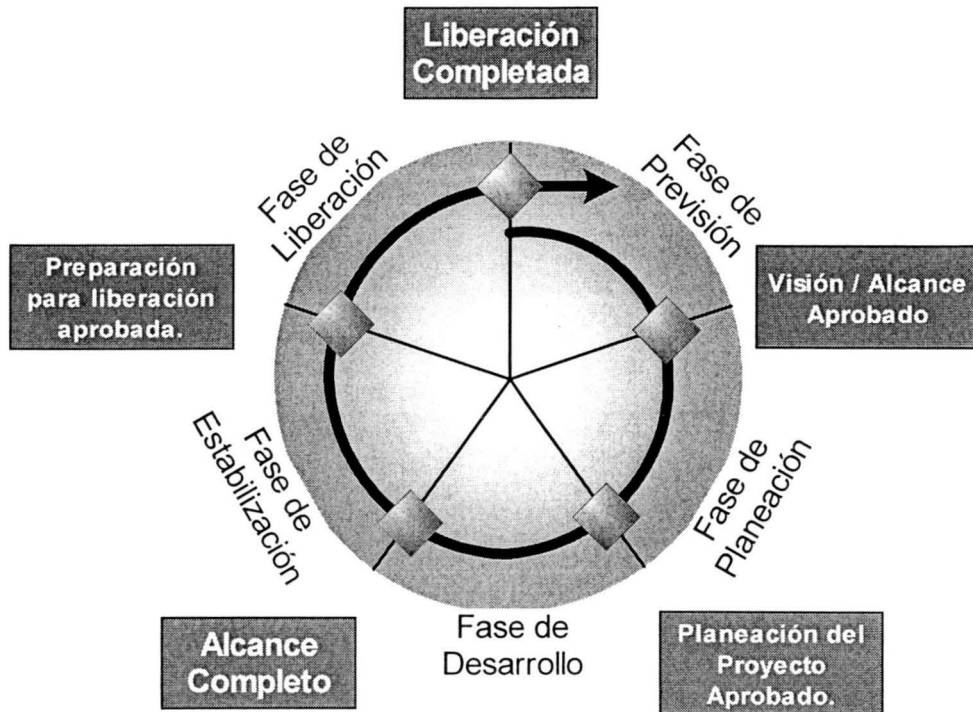


Figura 3.19 – Ciclo de vida para MSF [MSF, 2002]

3.4.3 Fase de Previsión - Estrategia y Alcance

En esta fase deben tener lugar los siguientes trabajos:

- ❖ Elaboración y aprobación del Documento de Alcance y Estrategia definitivo: debe ser un documento de consenso con la participación del número de agentes implicados en el proyecto. En este documento quedarán definitivamente reflejados las funcionalidades y servicios que debe ofrecer la solución a implantar. [Microsoft, 2003]

El documento debe contener las siguientes secciones:

- Declaración de oportunidad del negocio: Aquí se define el estado actual del cliente en lenguaje de negocios, en ningún momento se define algo técnico. Esto permite entender el estado actual del negocio desde el punto de vista del cliente.
- Declaración de visión: La sección describe de forma clara y concisa el estado futuro deseado de la empresa una vez que el proyecto haya concluido. Esta escrito como si el proyecto ya terminó. Este documento permite que todas las entidades entiendan el problema y cumplan con los objetivos.

- Análisis de Beneficios: Esta sección describe como el cliente va obtener valor de la solución propuesta. Relaciona las metas del negocio y objetivos con las expectativas a realizar en el proyecto. Esta sección debe ser en términos numéricos. El análisis costo-beneficio será utilizado en este caso.
- Análisis de Usuarios: Define los usuarios involucrados de la solución. Se especifican los usuarios propuestos que utilizarán este sistema. Normalmente se especifican por su área funcional y son de áreas de tecnologías de información y de negocios.
- Requerimientos: Debe identificar lo que la solución debe hacer. Se pueden describir en términos funcionales. Estos requerimientos pueden ser los siguientes:
 - Requerimientos del negocio.
 - Requerimientos del usuario.
 - Requerimientos operacionales.
 - Requerimientos del sistema.
- Alcances: Esta sección pone un límite alrededor de la solución detallando las especificaciones y funciones del sistema. En esta sección se incluyen:
 - Lista de funciones.
 - El “Fuera de Alcance” que define las funciones que se excluyen de la solución.

[Microsoft, 2003] [MSF, 2002]

❖ Formación del Equipo de Trabajo y distribución de competencias y responsabilidades:

Se especifica en una tabla los involucrados durante el desarrollo de la solución definida en el documento de alcance y estrategia. Generalmente se definen como áreas principales las que se describen en el diseño de arquitectura, modelo de base de datos, pruebas de laboratorio. [Microsoft, 2003]

❖ Elaboración del Plan de Trabajo:

Deben marcarse fechas y contenidos para esta fase y las siguientes. Los mecanismos y protocolos de intercambio de información y coordinación deben quedar suficientemente bien establecidos y consensuados. [Microsoft, 2003]

❖ Elaboración de la matriz de Riesgos y Plan de Contingencia:

Los principales riesgos detectados deben tener un plan de mitigación y actuación y revisarse con periodicidad. [Microsoft, 2003] A continuación se muestran los formatos a utilizar para definir los riesgos y planes de contingencia de cada uno:

- Tabla de riesgos posibles:

Organización: Nombre de Proyecto: Fecha:		
Posibles Riesgos	Designación	Acción a preocupaciones, asuntos y problemas.
Lista de posibles riesgos identificados haciendo lluvias de ideas, entrevistas o conocimientos propios por experiencia.	Especificar una preocupación, asunto o problema con una X y un riesgo con una R .	Definir la acción apropiada para cada asunto que se designó con una X en la columna anterior. Acciones pueden ser “consolidar”, “aumentar”, “referir”, “redefinir” o “tratar como fuera de alcance” y asignar una en columna anterior una X.

Tabla 3.3 Formato para Tabla de Riesgos Posibles [MSF,2002]

- Identificación de riesgos posibles (para uso de análisis de riesgos y prioridades):

Organización: Nombre de Proyecto: Fecha:				
Posibles Riesgos	Número identificador	Clasificación/ Categoría	Condiciones	Consecuencias
Los riesgos con designación "R" de la tabla de posibles riesgos se agregan aquí.	Usar un número único para identificar cada riesgo.	<ul style="list-style-type: none"> • Gente <ul style="list-style-type: none"> - Consumidor - Usuario Final - Patrocinador - Inversionistas - Personal - Organización - Habilidades - Política - Moral • Proceso <ul style="list-style-type: none"> - Misión y Metas - Toma de Decisiones - Características de Proyecto - Presupuesto, costos y calendario. - Requerimientos - Diseños - Construcción - Pruebas • Tecnología <ul style="list-style-type: none"> - Seguridad - Desarrollo y ambiente de pruebas - Herramientas - Liberación - Soporte - Ambiente Operacional - Disponibilidad • Ambiente <ul style="list-style-type: none"> - Legal - Regulaciones - Competencia - Económico - Tecnología - Negocios 	Identificar la causa del riesgo.	Definir el resultado potencial del riesgo, puede haber múltiples consecuencias para cada condición y algunas consecuencias pueden ser condiciones.

Tabla 3.4 Formato para Tabla de Riesgos Posibles [MSF,2002]

- Análisis de riesgos y prioridades:

Organización:					
Nombre de Proyecto:					
Fecha:					
Número de Identificación	Prioridad/ Orden	Probabilidad (Numérico)	Impacto (Numérico)	Exposición (Numérico)	Receptor de Exposición
		Probabilidad de que suceda el riesgo: 1%-30%: Bajo 31%-70%: Medio 71%-100%: Alto	Effect of the Risk Consequence, with 1 - 3 for low; 4 - 7 for medium, 8 - 10 for high, and 100 for catastrophic ratings Efecto del riesgo: 1-3: Bajo 4-7: Medio 8-10: Alto 100: Catastrófico	Calculado multiplicando Probabilidad por Impacto y usado como base para ordenar y dar prioridades a riesgos.	Identificar quien va ser afectado por la exposición del riesgo. Ejemplo: socio, cliente, compañía.

Tabla 3.5 Formato para análisis de riesgos y prioridades [MSF,2002]

3.4.4 Fase de Planeación - Planificación y Pruebas de Concepto

Deben alcanzarse los siguientes objetivos:

❖ Documento de Planificación y Diseño de Arquitectura:

Es el documento principal, donde se describen los aspectos funcionales y operativos de la nueva plataforma. La aprobación de este documento es el primordial de esta fase y deben ser consistentes con esta Guía. Si en el curso de las fases sucesivas fuera necesario revisar estos contenidos, se deberá hacer por acuerdo y conocimiento de todo el equipo de trabajo y se llevará un registro de versiones que permita hacer un seguimiento adecuado de estas revisiones. [Microsoft, 2003]

Sobre este plan se incluye la siguiente tabla con lo siguiente:

Componente	Descripción	Fecha Crítica de Entrega	Responsable
Nombre del componente a realizar	Breve descripción del componente.	Fecha para la entrega de este componente	La persona responsable del éxito de este componente.

Tabla 3.6 Planificación de desarrollo de componentes [MSF,2002]

❖ Documento de Plan de Laboratorio - Prueba de Concepto:

La descripción del contenido del laboratorio de prueba de concepto y los diversos escenarios a simular son objetivos a cubrir en este documento. Es un documento dinámico, en el que se recoge la idea y la experiencia práctica al llevarla a cabo en entorno controlado y aislado. La etapa de prueba de laboratorio concluye cuando la maqueta ofrece todos los servicios y funciones descritos en el Documento de Alcance y Estrategia, y su grado de estabilidad y rendimiento es considerado como "suficiente". [Microsoft, 2003]

El plan de laboratorio se compone por escenarios, estos se utilizan para simular las diferentes funciones del sistema. Para elaborar los escenarios se utiliza los diagramas de flujo como se muestra a continuación:

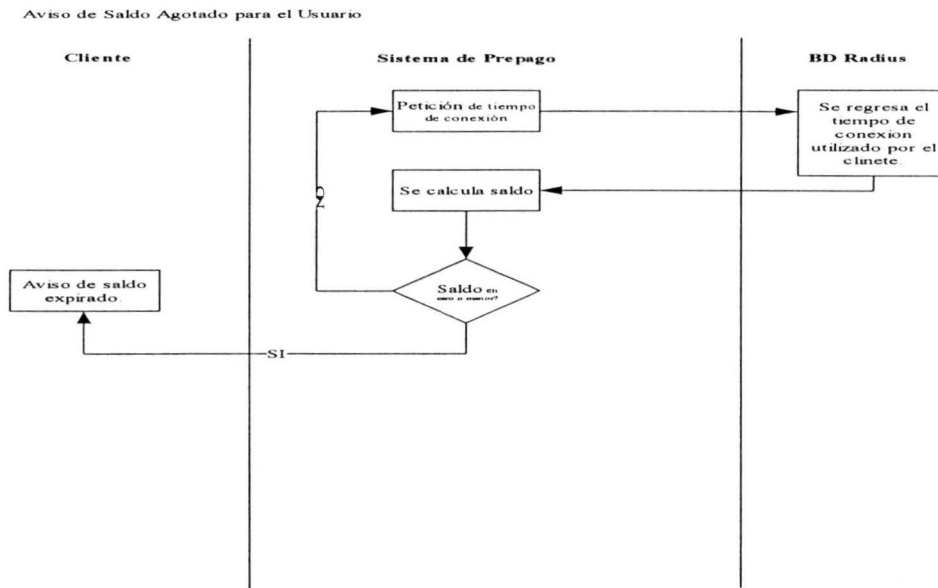


Figura 3.20 – Ejemplo de escenario con diagrama de flujo.

Cada ambiente o entidad está separado por una línea vertical. Cada proceso o paso del diagrama de flujo que está en esta área delimitada es responsabilidad del ambiente de tecnología de información o entidad.

Después de diseñar el escenario con diagramas de flujo se pasa a llenar el siguiente formato para definir claramente el caso del escenario:

Nombre del caso

Descripción	Breve descripción del caso.
identificador de escenario	Utilizar un identificador para el escenario. Ejemplo: "Escenario Básico 1" utilizar EB1. Esto se utiliza para rastreo dentro de los escenarios.
Autor	Persona(s) que contribuye(n) al texto.
Fecha	Última fecha de actualización
Revisión	Persona que hizo ultima revisión
Actores	Lista de actores (involucrados en proceso)
PRE-condiciones	Condiciones que deben existir antes de que el escenario pueda comenzar.
Acciones	Lista de acciones que se utilizan en este caso.
Post-condiciones	Lista de condiciones que deben ser cumplidas antes de terminar el escenario.
Agregados	Hacer referencia a otros casos que use este escenario.
Extensiones	Lista de casos que este escenario extiende o se construye sobre.
Generalizaciones	Casos donde este escenario sea una generalización de otros.

Tabla 3.7 Planificación de desarrollo de componentes [MSF,2002]

3.4.5 Fase de Desarrollo

Durante la fase de desarrollo el equipo cumple con la mayoría del diseño y construcción de los componentes de la solución propuesta. Será necesario en la fase de estabilización rediseñar procesos debido a las pruebas que se harán sobre los componentes. [MSF, 2002]

Los objetivos en ésta fase son los siguientes:

- Documento de pruebas y reporte de fallas (Gestión de Incidencias): Este documento contiene un reporte resumido de pruebas y fallas, los métodos y herramientas utilizadas y pruebas que faltan realizar. A continuación se deben describir las pruebas hechas a cada componente o área del producto de la siguiente forma:
 - Metas de la prueba: Lo que debe cumplir la prueba.
 - Criterio de evaluación: Criterio utilizado para evaluar la prueba.
 - Resultados de la prueba.
 - Recomendaciones.

- Entrega de código fuente y ejecutables.
- Instalación y configuración de maquinas para liberación.

[MSF, 2002]

3.4.6 Fase de Estabilización

La solución implantada en la maqueta se pasa a un entorno real de explotación, restringido en número de usuarios y en condiciones tales que se pueda llevar un control efectivo de la situación. Esto se hace cuando el desarrollo de todos los componentes ha sido terminado. [Microsoft, 2003]
[MSF, 2002]

Los objetivos fundamentales de esta fase son:

Selección del entorno de prueba piloto

Se acordará la composición y ubicación del conjunto de máquinas y usuarios que entrarán en la prueba. Esta selección se recomienda que se haga atendiendo a la mayor variedad posible de casos, de manera que puedan aflorar el máximo de incidentes potenciales en el menor tiempo posible. La dimensión de la muestra tiene también que calcularse, sin perder de vista que la prueba piloto no es el despliegue propiamente, sino una fase de observación en la que es absolutamente crítico establecer unos cauces efectivos de tratamiento de los errores. [Microsoft, 2003]

Gestión de Incidencias

Aunque esta labor se habrá iniciado en la fase anterior, el éxito de la prueba piloto dependerá de que se forme una de captura de incidentes y de resolución de problemas y documentación de los mismos. [Microsoft, 2003]
[MSF, 2002]

Se elabora un documento adicional donde se especifican los casos de pruebas donde se identifica cada prueba que se hará del sistema. El formato a seguir es el siguiente:

- Elemento de prueba (numerados): Se presenta cada elemento a probar del sistema, expresa el área donde se enfocarán las pruebas.
 - Característica A
 - Comportamiento esperado.
 - Desempeño esperado.
 - Confiabilidad esperada.
 - Característica B
 - Comportamiento esperado.
 - Desempeño esperado.
 - Confiabilidad esperada.
 -
 -
 - Característica Z
 - Comportamiento esperado.
 - Desempeño esperado.
 - Confiabilidad esperada.

Revisión de la documentación final de Arquitectura

El documento de Planificación y Diseño de Arquitectura se puede ver alterado parcialmente como resultado de esta fase. El documento final, aprobado por consenso, supone el principal documento del Proyecto y la culminación de los trabajos de diseño, al menos en sus líneas principales. Este documento se considerará definitivo cuando la solución puesta en marcha se muestre estable y el número de incidencias graves (de intervención o de resolución) sea nulo y la cantidad de las consideradas leves sean pocas. [Microsoft, 2003]

Elaboración del Plan de Despliegue

Se debe acordar la fecha de finalización de la fase Piloto, y las condiciones de calidad que debe cumplir la solución final para iniciar el despliegue. En el Plan deben identificarse las fases, estrategia de implantación, fechas, tareas a realizar, procedimientos de validación y método de control de incidencias. [Microsoft, 2003]

El tiempo necesario para abordar esta fase es variable y depende en parte de factores ajenos a la complejidad de la propia solución, como es la adecuada selección del entorno de prueba y el momento del año en que tenga lugar (evitando que coincida con periodos de vacaciones o puntas de trabajo críticas como Fin de Año). [Microsoft, 2003]

3.4.7 Fase de Liberación - Despliegue

Se llevarán a cabo en esta fase los planes diseñados en la anterior, principalmente el de despliegue. Los principales trabajos a conseguir son, en este caso, además de los obvios (implantación de la plataforma, puesta en servicio de todas las funciones los siguientes: [Microsoft, 2003]

- Continuación con las labores de recepción de incidencias, clasificación, tratamiento, resolución y distribución de arreglos o intervención en el sitio. [Microsoft, 2003]
- Registro de mejoras y sugerencias, funcionalidades no cubiertas y novedades a incorporar en sucesivas versiones de la plataforma, incluyendo mejoras aportadas por los fabricantes de software (nuevas versiones o *Service Packs*, por ejemplo) [Microsoft, 2003]
- Entrega de los documentos definitivos acordados como entregables. [Microsoft, 2003]
- Finalmente, entrega del Proyecto y cierre del mismo, con o sin apertura de nuevo proyecto en base a la información y experiencia obtenidas. [Microsoft, 2003]

La duración fase de despliegue, puesto que debe planificarse, no puede establecerse a priori. Depende de numerosos factores externos al propio proyecto (incluyendo factores de oportunidad política o de negocio) que pueden retardar o acelerar la conclusión. [Microsoft, 2003]

La experiencia demuestra que no hay una relación directa entre número de máquinas y tiempo necesario para el despliegue. Los factores más relevantes en el cálculo suelen ser la dispersión o concentración geográfica, la complejidad del proceso de migración, el grado de automatización alcanzado, la experiencia y nivel de los técnicos que realizan la operación y

condicionantes de calendario, a menudo con restricciones no técnicas, sino de otros tipos (las fechas-objetivo suelen marcarse por criterios de oportunidad de negocio). [Microsoft, 2003]

En esta fase se deberá entregar una captura de segunda incidencia, este documento es un cierre al proyecto que contiene los siguientes puntos:

- Resumen de cierre de proyecto: Aquí se incluye un breve resumen.

- Aproximación a la visión establecida: Esta sección describe como el proyecto cumplió por completo, se aproximó o fallo en cumplir con la visión establecida. Se incluye la visión reestructurada y diferencias que hubo con la visión original.

- Cambios que impactaron en la visión: Se describen los eventos clave que causaron el cambio en la visión original. Este punto se separa en varias categorías que describen los cambios que hubo dentro de cada una:
 - Cambios en el Negocio
 - Cambios en la Visión
 - Cambios en el equipo de trabajo
 - Cambios en los procesos del cliente.
 - Cambios en el proceso del proyecto
 - Cambios en especificaciones
 - Cambios en la itinerario de actividades

- Próximos pasos - Siguiete versión: Se incluyen comentarios para una futura visión del proyecto actual. Se puede incluir una descripción de nuevas características para una segunda versión del proyecto. El enfoque principal de esta sección es establecer una idea inicial de cómo será la siguiente visión a establecer para comenzar el proceso *MSF* de nuevo.

3.5 Resumen y Conclusión de Capitulo

Se cubren en este capitulo las herramientas y metodología que serán utilizadas durante el desarrollo. También se mencionaron algunos productos parecidos y su comparación con el que se desarrolló.

El uso de la metodología *Microsoft Solution Framework* permite planificar, diseñar, construir y liberar sistemas a tiempo con un uso controlado de recursos económicos y humanos. Esta metodología permite una mejor flexibilidad a cambios en la solución propuesta durante el ciclo de vida.

4. ANÁLISIS

4.1 Alcance y Estrategia

A continuación se explican los resultados del documento de alcance y estrategia según la metodología aplicada *MSF*.

4.1.1 Oportunidad de Negocio

Como se definió en el capítulo dos, la oportunidad de negocio es el mercado de Internet Prepagado vía *dial-up*. Se busca ingresar a este mercado para ofrecerle un mejor producto al cliente. Una ventaja de las tarjetas de prepago es su portabilidad, puede ser utilizada en varios estados de la República Mexicana a través de números locales o 01-800 nacional y Estados Unidos por 1-800 internacional.

4.1.2 Visión

El sistema de Tarjetas de Internet prepago le permitirá a Marcatel administrar las tarjetas de Internet desde su creación hasta el consumo y expiración. Además, por medio de servicios *web* el sistema brindará a los usuarios la posibilidad de comprar y recargar sus tarjetas de Internet prepago, además de consultar su saldo e información de la tarjeta.

4.1.3 Solución

Se propuso e laborar un sistema que permite a dministrar las tarjetas prepagadas para Internet desde su creación física o virtual en Internet y herramientas que el cliente pueda utilizar para la administración personal de estas mismas.

4.1.4 Beneficios

Como se mencionó en el capítulo dos Marcatel buscó un aumento de ventas en el área de servicios de Internet vía telefónica y una mejor satisfacción del cliente. A continuación se muestra el análisis costo-beneficio y retorno de inversión estimados para este producto:

Detalle los costos tangibles del Proyecto:

	Costo en unidades de medición apropiadas	Explicación	Traducido a Dinero	Relación con Indicadores
Tiempo:	720 Horas	Tiempo abarcado por el proyecto tomando en cuenta un horario de 4.5 horas diarias durante 16 semanas y realizadas por 2 personas, dentro de este costo se toman en cuenta los gastos de transportación y alimento	\$3200	Tiempo en el que se desarrollarán los módulos del sistema.
Personal Marcatel	2 personas	Basado en el tiempo dedicado de las personas encargadas del proyecto en Marcatel(20 horas al mes en promedio)	\$30000	Se deberán tomar en cuenta los puntos clave para cumplir con el desarrollo integro del sistema
Materiales Oficina	2 Paquetes de hojas de máquina y 2 cartuchos de tinta para la impresora	Tomando en cuenta el material utilizado para el análisis , diseño y desarrollo del sistema	\$1200	Los recursos son asignados por la empresa para el uso de los desarrolladores

	Costo en unidades de medición apropiadas	Explicación	Traducido a Dinero	Relación con Indicadores
Hardware	4 Servidores, 1 Radius Server, 2 computadoras	En los servidores se alojarán el sistemas y las bases de datos ocupados por el mismo, además del espacio ocupado por las cuentas en el Radius Server, también se destinaron 2 computadoras para el desarrollo del sistema, de las cuales una fue comprada y la otra ya se contaba con ella.	\$10800	En los servidores se concentrará la información de las tarjetas prepagadas
Software	Windows 2000 Server, SQL Server 2000, Visual Studio .NET, API de Banco, ACS Radius, API Radius, Microsoft Office 2000 , Microsoft Visio 2000, Erwin	El desarrollo será basado en .Net (ASP.NET y VB.NET) con base de datos en SQL 2000. La documentación y diagramas serán realizados con Office, Edwin y Visio 2000. Todo el software antes mencionado se encuentra disponible y con licencias dentro de la empresa	\$0	Con el uso de ASP.NET y VB.NET se permitirá tener accesos vía Web y el desarrollo de objetos y servicios utilizados por el sistema

	Costo en unidades de medición apropiadas	Explicación	Traducido a Dinero	Relación con Indicadores
Servicios	Agua, Luz e instalaciones	Tomando en cuenta el porcentaje de servicios ocupados por el sistema y el personal del proyecto	\$1200	Los servidores sen encontrarán en el edificio de Marcatel Internacional
Materiales	Impresión de tarjetas plásticas	Impresión de tarjetas de Internet prepago según estadísticas de venta de Marcatel	\$2000	Las tarjetas son mandada a imprimir con un proveedor de Marcatel

Costo Total : \$ 48,400

TABLA 4.1 Detalle de costos tangibles del proyecto

Detalle de los costos de Operación(Operación semestral)

	Descripción	Cantidad
Servicios	Cantidad derivada de los servicios ocupados para que el sistema opere(Luz)	\$2,000
Operadores	Sueldo de las personas encargadas de operar el sistema	\$20,000
Módulo de servicio de tarjeta de crédito	Renta del servicio de tarjeta de crédito Banorte tomando en cuenta un promedio de venta de 60 tarjetas en 6 meses	3.15% por cada transacción(\$180)
Mantenimiento	Costo de mantenimiento del sistema	\$5,000

TABLA 4.2 Detalle de costos tangibles del proyecto

Costo Total : \$27,180.00

Explique la implicación de que el proyecto no se lleve a cabo:

	Explicación
Tiempo:	No existiría algún cambio ya que actualmente no existe el servicio de Internet prepagado.
Personas:	Se tendría que destinar 3 personas para la instalación y operación del sistema a comprar.
Materiales:	Se utilizarían los mismos materiales.(impresión de tarjetas)
Hardware:	Se tendría que comprar un sistema que administre las cuentas de DialUp para ofrecer el servicio de Internet Prepagado
Software:	El software a utilizar sería el software que incluyera el sistema comprado.
Redes:	Seguiría existiendo la misma infraestructura con la que actualmente se cuenta.
Dinero:	Se observaría una gran pérdida de clientes, traducida a dinero, además de no contar con las ganancias que les generaría el servicio de Internet prepagado.

TABLA 4.3 Implicaciones de que el proyecto no se lleve a cabo

Detalle los beneficios tangibles del proyecto:

	Beneficio en unidades de medición	Explicación del Cálculo	Traducido a Dinero
Tiempo:	0	No existe ningún beneficio ya que el sistema no involucra ahorro de tiempo	\$0
Personas:	3 personas	Ahorro en sueldo de las personas que instalarán y pondrán a funcionar el sistema Administrador de cuentas DialUp	\$20,200
Materiales:	0	No existe ningún beneficio ya que el sistema sigue utilizando los mismos materiales	\$0
Hardware:	Accounting Manager	Ahorro en la compra del Hardware necesario para el funcionamiento del sistema administrador de cuentas DialUp	\$816,960*
Software:	0	No existe ningún beneficio en Software ya que se utilizará el actual.	\$0
Redes:	0	No existe ningún beneficio en Redes ya que se utilizará la actual.	\$0

Beneficio Total : \$ 837,160

*Tomando en cuenta el valor del dólar en \$11.04

TABLA 4.4 Beneficios Tangibles del Proyecto

Detalle los beneficios intangibles del proyecto:

Los beneficios intangibles que este proyecto ofrece, es el proveer a Marcatel de un nuevo servicio que le permitirá aumentar su cartera de clientes, además de ofrecerle una competitividad en el mercado, a actualmente no la tiene para este tipo de productos.

Resta al Beneficio el Costo:

$$837,160 - 48,400 = \$788,760$$

Retorno sobre la Inversión ROI del proyecto:

Período (6 meses)	Costos del período	Beneficios del período	Beneficio Neto
1 semestre	\$48,400	\$0	-\$48,400
2 semestre	\$27,180	\$30,000	\$2,820
3 semestre	\$27,180	\$35,000	\$7,820
4 semestre	\$27,180	\$40,000	\$12,820
5 semestre	\$27,180	\$50,000	\$22,820
6 semestre	\$27,180	\$60,000	32,820
Total			\$30,700

TABLA 4.5 Retorno de Inversión del Proyecto

*El retorno de la inversión se espera dentro del tercer año de operación

Comparación de los sistemas de Internet Prepagado

Comparación de costos

Concepto	Sistema Internet prepago Marcatel	Administrador de cuentas DialUp
Personal Marcatel	30,000	20,200
Personal Udem	3,200	0
Material de Oficina	1,200	0
Hardware	10,800	816,960
Software	0	0
Servicios	1,200	2,500
Materiales	2,000	2,000
Totales	48,400.00	841,660.00

TABLA 4.6 Comparación de sistemas de Internet Prepagado

Retorno de inversión - Sistema de Internet Prepagado Marcatel

Semestres	Beneficio
1 Semestre	-\$48,400
2 Semestre	-\$45,580
3 Semestre	-\$37,760
4 Semestre	-\$24,940
5 Semestre	-\$2,120
6 Semestre	\$30,700

TABLA 4.7 Retorno de inversión de Sistema Internet Prepagado Marcatel

Retorno de inversión - Sistema Administrador de cuentas DialUp

Semestres	Beneficio
1 Semestre	-\$837,160
2 Semestre	-\$834,340
3 Semestre	-\$826,520
4 Semestre	-\$813,700
5 Semestre	-\$790,880
6 Semestre	-\$758,060

TABLA 4.8 Retorno de Inversión para Sistema Administrador de Cuentas *DIAL-UP*

Gráfica comparativa entre proyectos

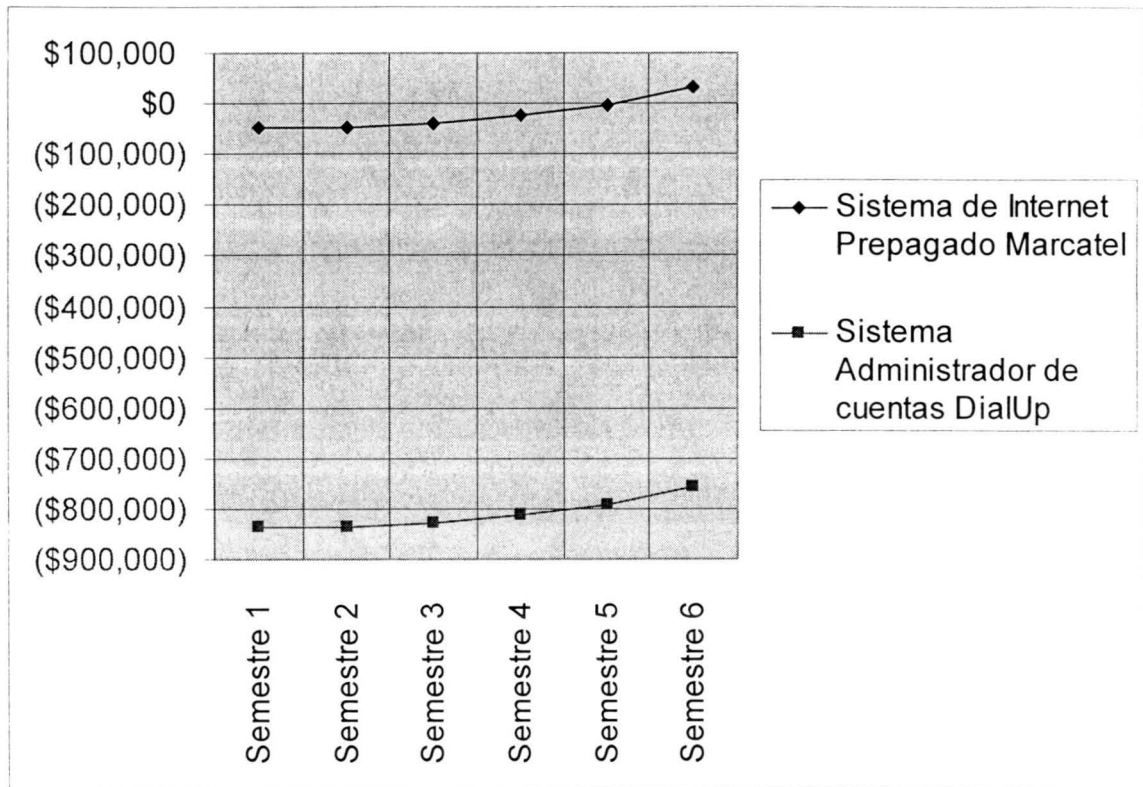


FIGURA 4.3 Gráfica comparativa entre proyectos

4.1.5 Análisis de Usuarios

A continuación se especifican los tipos de usuarios involucrados en términos generales:

Cliente: El cliente es el usuario final, quien compra las tarjetas de prepago y el que se conecta a Internet.

Administrador: Es el usuario (personal de Marcatel) quien hace uso del sistema administrativo de prepago.

4.1.6 Requerimientos

A continuación se describen los requisitos operacionales para el sistema de Internet prepagado:

Servicios Globales del Sistema

Max TNT Server:

Permite la conexión de los clientes a Internet por vía telefónica.

Radius Server:

Servidor que administra las cuentas y el acceso a Internet.

Router:

Enlaza la red en donde se encuentra el TNT Server con la red que tiene acceso a Internet.

Base de Datos SQL Log de Radius:

Base de datos que almacena, lee o borra la información de las cuentas del Radius.

Base de Datos SQL Tarifas:

Base de datos que almacena, lee o borra la información de las diferentes tarifas del sistema.

Base de Datos SQL Tarjetas:

Base de datos que almacena, lee y borra la información de las tarjetas de prepago, los agentes de ventas e información relacionada con las tarjetas de prepago.

Base de Datos SQL Transacciones:

Base de datos que almacena, lee y borra la información de las transacciones realizadas con el banco, durante la recarga de tarjetas por medio de tarjeta de crédito.

Monitoreo de saldos:

Proceso que se encarga de evaluar periódicamente las cuentas activas, revisando los saldos y desconectando a los clientes cuyo saldo se ha terminado.

Servicios al Cliente

Acceso:

Válida y permite el uso al cliente del sistema (únicamente los módulos visibles al cliente).

Interfaz:

Grupo de pantallas que permiten al administrador crear, editar o borrar la información del sistema.

Consulta de saldos:

Por medio de éste módulo el cliente puede consultar el saldo actual de su tarjeta d prepago.

Recarga de tarjeta:

Interfaz que permite al cliente la opción de recargar su tarjeta ya sea por medio de traspaso de saldos de tarjeta a tarjeta o por recarga vía tarjeta de crédito.

Recarga de tarjeta a tarjeta:

Módulo que permite al cliente recargar su tarjeta transfiriendo el saldo de una o más tarjeta a otra.

Recarga por tarjeta de crédito:

Módulo que permite al cliente recargar su tarjeta comprando y abonando saldo por medio de su tarjeta de crédito.

Sistema de Administración

Acceso:

Válida y permite el uso al administrador del sistema.

Interfaz:

Grupo de pantallas que permiten al administrador crear, editar o borrar la información del sistema.

Conciliaciones:

Módulo que por el cual se verifica que la información recibida de los pagos coincida con las cantidades vendidas, descartando la posibilidad de tener errores en la información final de ventas.

Administración de tarjetas:

Permite crear, editar y borrar la información de las tarjetas de prepago dentro del sistema.

Administración de agentes de ventas:

Permite crear, editar y borrar la información de los agentes de ventas.

Generación de tarjetas a impresos:

Módulo que permite generar el archivo con la información necesaria para imprimir físicamente las tarjetas de prepago.

Consultas:

Permite al administrador hacer consultas sobre la información contenida en el sistema, como son las tarjetas, agentes de ventas, cuentas generadas.

Tarificación:

Permite crear, editar o borrar las diferentes tarifas utilizadas por el sistema.

Banco:

El banco proporciona toda la información y configuración necesaria para validar las cuentas de tarjeta de crédito en su base de datos y pagos a realizar.

4.1.7 Equipo de Trabajo, distribución de competencias y responsabilidades

Las actividades realizadas definidas fueron las siguientes:

No. Actividad	Actividad
1	Junta para definir funciones del proyecto
2	Junta para definir el producto
3	Aprendizaje de lenguajes VB.NET y ASP.NET
4	Elaboración del Documento de Alcance y Estrategia
5	Aprobación del Documento de Alcance y Estrategia
6	Diseño de la Arquitectura del Sistema
7	Diseño del Modelo de Base de datos(Entidad relación)
8	Elaboración de las Pruebas de Laboratorio
9	Elaboración del Plan de trabajo
10	Elaboración de la Matriz de riesgos
11	Elaboración del Plan de contingencia
12	Aprobación de la Planificación y Diseño de Arquitectura
13	Construcción de base de datos de tarjetas
14	Construcción de base de datos de Log de transacciones
15	Desarrollo de página principal de clientes
16	Desarrollo de módulo de acceso de clientes
17	Desarrollo de módulo de consulta de saldo
18	Desarrollo de módulo de recarga de tarjeta a tarjeta.
19	Desarrollo de modulo de compra y recarga con tarjeta de crédito.
20	Módulo cliente para monitoreo de saldos.
21	Desarrollo de tarifas para tarjetas.
22	Desarrollo de expiración de tarjetas.
23	Servicio para monitoreo de conexiones y saldos.
24	Desarrollo de estatus de tarjetas.
25	Adaptar tarjetas prepagadas a módulo de agentes de venta.
26	Adaptar tarjetas prepagadas a módulo de administración de tarjetas.
27	Generación de cuentas válidas para tarjetas.
28	Generación y adaptación de reportes necesarios
29	Desarrollo y pruebas sobre ambiente Radius.
30	Generación de archivo de pagos para Facturación
31	Selección de entorno de pruebas piloto
32	Realización de pruebas sobre la maqueta
33	Captura de incidencias durante la prueba
34	Aprobación de la implementación de Arquitectura Final del sistema.
35	Elaboración del Plan de despliegue
36	Despliegue
37	Captura de Incidencia 2ª versión
38	Elaboración de Documento de mejoras
39	Entrega de documentos establecidos
40	Entrega del proyecto
41	Documentación de proyecto
42	Presentaciones
43	Juntas con asesor
44	Revisiones con Erich Von Hauske

Tabla 4.9 Actividades a realizar

A continuación se define como quedaron establecidas las responsabilidades:

No. Actividad	Actividad	Asignación
1	Junta para definir funciones del proyecto	Gerardo Canepa Fabián Orta Ing Erich Von Hauske Ing Juan Carlos Balboa
2	Junta para definir el producto	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske Ing. Juan Carlos Balboa
3	Aprendizaje de lenguajes VB.NET y ASP.NET	Gerardo Canepa Fabián Orta Asesor: Ing. Jorge López
4	Elaboración del Documento de Alcance y Estrategia	Fabián Orta
5	Aprobación del Documento de Alcance y Estrategia	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske Ing. Juan Carlos Balboa Ing. Marta Sylvia del Río
6	Diseño de la Arquitectura del Sistema	Gerardo Canepa
7	Diseño del Modelo de Base de datos(Entidad relación)	Gerardo Canepa Fabián Orta
8	Elaboración de las Pruebas de Laboratorio	Gerardo Canepa Fabián Orta
9	Elaboración del Plan de trabajo	Gerardo Canepa
10	Elaboración de la Matriz de riesgos	Fabián Orta
11	Elaboración del Plan de contingencia	Gerardo Canepa
12	Aprobación de la Planificación y Diseño de Arquitectura	Gerardo Canepa Fabián Orta Ing Erich Von Hauske Ing. Marta Sylvia del Río
13	Construcción de base de datos de tarjetas	Gerardo Canepa
14	Construcción de base de datos de Log de transacciones	Fabián Orta
15	Desarrollo de página principal de clientes	Fabián Orta
16	Desarrollo de módulo de acceso de clientes	Fabián Orta
17	Desarrollo de módulo de consulta de saldo	Gerardo Canepa Fabián Orta
18	Desarrollo de módulo de recarga de tarjeta a tarjeta.	Fabián Orta Ing. Jorge López
19	Desarrollo de módulo de compra y recarga con tarjeta de crédito.	Gerardo Canepa Ing: Jorge López
20	Módulo cliente para monitoreo de saldos.	Gerardo Canepa Fabián Orta

Tabla 4.10 Distribución de responsabilidades del 1-20

No. Actividad	Actividad	Asignación
21	Desarrollo de tarifas para tarjetas.	Fabián Orta Ing. René Pereyra Ing. Erich Von Hauske
22	Desarrollo de expiración de tarjetas.	Gerardo Canepa
23	Servicio para monitoreo de conexiones y saldos.	Gerardo Canepa Fabián Orta
24	Desarrollo de estatus de tarjetas.	Gerardo Canepa
25	Adaptar tarjetas prepagadas a módulo de agentes de venta.	Gerardo Canepa Ing. Ruth Leal
26	Adaptar tarjetas prepagadas a módulo de administración de tarjetas.	Gerardo Canepa Ing. Ruth Leal
27	Generación de cuentas válidas para tarjetas.	Gerardo Canepa Fabián Orta
28	Generación y adaptación de reportes necesarios	Gerardo Canepa Fabián Orta Ing. Ruth Leal
29	Desarrollo y pruebas sobre ambiente Radius.	Fabián Orta Ing. Erich Von Hauske
30	Generación de archivo de pagos para Facturación	Gerardo Canepa Fabián Orta Ing. Jorge López Ing. René Pereyra
31	Selección de entorno de pruebas piloto	Fabián Orta
32	Realización de pruebas sobre la maqueta	Gerardo Canepa Fabián Orta
33	Captura de incidencias durante la prueba	Gerardo Canepa Fabián Orta
34	Aprobación de la implementación de Arquitectura Final del sistema.	Gerardo Canepa Fabián Orta Ing: Erich Von Hauske Ing. Marta Sylvia del Río
35	Elaboración del Plan de despliegue	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske
36	Despliegue	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske Ing. Jorge Lopez Ing. Ruth Leal Ing. René Pereyra
37	Captura de Incidencia 2ª versión	Gerardo Canepa Fabián Orta
38	Elaboración de Documento de mejoras	Gerardo Canepa
39	Entrega de documentos establecidos	Gerardo Canepa Fabián Orta
40	Entrega del proyecto	Gerardo Canepa Fabián Orta
41	Documentación de proyecto	Gerardo Canepa Fabián Orta
42	Presentaciones	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske Ing. Marta Sylvia del Río
43	Juntas con asesor	Gerardo Canepa Fabián Orta Ing. Marta Sylvia del Río
44	Revisiones con Erich Von Hauske	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske

Tabla 4.11 Distribución de responsabilidades del 21-44

4.1.8 Plan de trabajo

A continuación se presenta el plan de trabajo que se observó para este proyecto. El proyecto comenzó el 4 de agosto del 2003 y terminó el 28 de noviembre del 2003.

Actividades Proyecto	SEMANA																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Junta para definir el proyecto	■	■															
Junta para definir el producto	■																
Aprendizaje de lenguajes VB.NET y ASP.NET	■	■	■														
Elaboración del Documento de Alcance y Estrategia				■	■	■											
Aprobación del Documento de Alcance y Estrategia				■	■	■	■										
Diseño de la Arquitectura del Sistema				■	■	■	■	■									
Elaboración del Plan de trabajo				■	■	■	■	■									
Elaboración del Plan de contingencia				■	■	■	■	■									
Diseño del Modelo de Base de datos(Entidad relación)				■	■	■	■	■									
Elaboración de las Pruebas de Laboratorio				■	■	■	■	■									
Elaboración de la Matriz de riegos				■	■	■	■	■									
Aprobación de la Planificación y Diseño de Arquitectura				■	■	■	■	■									
Construcción de base de datos de tarjetas								■	■								
Construcción de base de datos de Log de transacciones								■	■								
Desarrollo de página principal de clientes								■	■	■							
Desarrollo de módulo de acceso de clientes								■	■	■							
Desarrollo de módulo de consulta de saldo								■	■	■							
Desarrollo de módulo de recarga de tarjeta a tarjeta.								■	■	■							
Desarrollo de modulo de compra y recarga con tarjeta de crédito.								■	■	■							
Módulo cliente para monitoreo de saldos.											■	■	■	■			
Desarrollo de tarifas para tarjetas.											■	■	■	■			
Desarrollo de expiración de tarjetas.											■	■	■	■			
Servicio para monitoreo de conexiones y saldos.											■	■	■	■			
Desarrollo de estatus de tarjetas.											■	■	■	■			
Adaptar tarjetas prepagadas a módulo de agentes de venta.											■	■	■	■			
Adaptar tarjetas prepagadas a módulo de administración de tarjetas.											■	■	■	■			
Generación de cuentas válidas para tarjetas.											■	■	■	■			
Generación y adaptación de reportes necesarios											■	■	■	■			
Desarrollo y pruebas sobre ambiente Radius.											■	■	■	■			
Generación de archivo de pagos para Facturación											■	■	■	■			
Revisiones con Erich Von Hauske								■	■	■	■	■	■	■	■	■	■
Elaboración de Documento de mejoras																	
Elaboración del Plan de Laboratorio																	
Selección de entorno de pruebas piloto																	
Realización de pruebas sobre la maqueta																	
Captura de incidencias durante la prueba																	
Aprobación de la implementación de Arquitectura Final del sistema																	
Elaboración del Plan de despliegue																	
Despliegue																	
Captura de Incidencia 2ª versión																	
Entrega de documentos establecidos																	
Entrega del proyecto																	
Documentación de proyecto																	
Presentaciones																	
Juntas con asesor																	

Tabla 4.12 Plan de trabajo de actividades a realizar.

4.1.9 Matriz de Riesgos y Planes de Contingencia

Se presenta la tabla de identificación de riesgos:

Riesgos Posibles - Identificación de riesgos	
Marcatel international	
Internet prepago	
Fecha: 19-Sep-2003	
Posibles Riesgos	Designaciones
Las fechas estipuladas no se cumplan	X
El API de Radius no sea establecido a tiempo	R
El módulo de creación de tarjeta no sea terminado a tiempo para acoplarlo	X
La computadora pueda perder información	R
La Liberación se retrase por problemas Legales	R

Tabla 4.13 Tabla de identificación de riesgos

X- Preocupación

R- Riesgo

A continuación se le asignó una clasificación al riesgo, condición y consecuencias:

Clasificación de Riesgos Marcatel International Internet prepago Fecha: 19-Sep-2003				
Posible riesgo	No. de Riesgo	Clasificación	Condición	Consecuencias
El Api de Radius no sea establecido a tiempo	1	Proceso - Toma de decisión	API no entregado a tiempo	Si no se cumple la fecha el proyecto se va a retrasar.
La computadora pueda perder información	2	Tecnología - Ambiente de trabajo	Virus o falla de sistema	Pérdida de la información de la computadora
La Liberación se retrase por problemas Legales	3	Ambiente Laboral - Legal	Proceso de prepago registrado por Todito.com	No se pueda liberar el sistema en el tiempo establecido.

Tabla 4.14 Tabla de clasificación y descripción de riesgos.

A continuación se asigno una probabilidad a cada riesgo:

Análisis de Riesgo y prioridad					
Marcatel International					
Internet Prepago					
Fecha: 19-Sep-2003					
No. De riesgo	Prioridad	Probabilidad de que ocurrencia	Impacto	Exposición	Afectado
1	1	35%	7	245	Desarrollo
3	2	65%	7	455	Liberación, Marcatel
2	3	1%	10	10	Sistema Entero

Tabla 4.15 Asignación de probabilidades a riesgos.

A continuación se asignó un plan de contingencia por cada riesgo:

Plan de Riesgo										
Marcatel International Internet prepago Fecha: 19-Sep-2003										
No. de Riesgo	Prioridad	Plan de Mitigación	Plan de contingencia	Triggers	Responsable	Entregables	Actividades	Dependencias	Fechas	Requerimientos
1	1	Presionar/Recordar al responsable de la toma de decisión para que lo haga a tiempo	Realizar junta inmediata o tener un API cualquiera para hacer pruebas	1 semana antes de la fecha establecida	Fabian Orta, Gerardo Canepa	Minutas	Juntas		1 día después de la fecha establecida	
3	2	Mandar aviso al departamento de legal para solucionar el problema	Liberar Internamente o cambiar la forma del producto	1 mes antes de Liberación	Depto. De Sistemas	Minutas	Juntas			
2	3	Respaldos diarios	Tener una segunda maquina disponible	Pérdida de Información	Fabian Orta, Gerardo Canepa		Respaldos	Copias de Proyectos y Documentos	Diario	Disquetes, CD's, dispositivos USB

Tabla 4.16 Asignación de plan de contingencia a riesgos.

4.2 Planeación

A continuación se presenta la fase de planeación para el proyecto de tarjetas prepagadas. Se presenta la arquitectura de sistema y los escenarios para el producto.

4.2.1 Arquitectura de Sistema

Una vez definidos los requerimientos operacionales en el punto 4.1.6 definimos la arquitectura de sistema del proyecto:

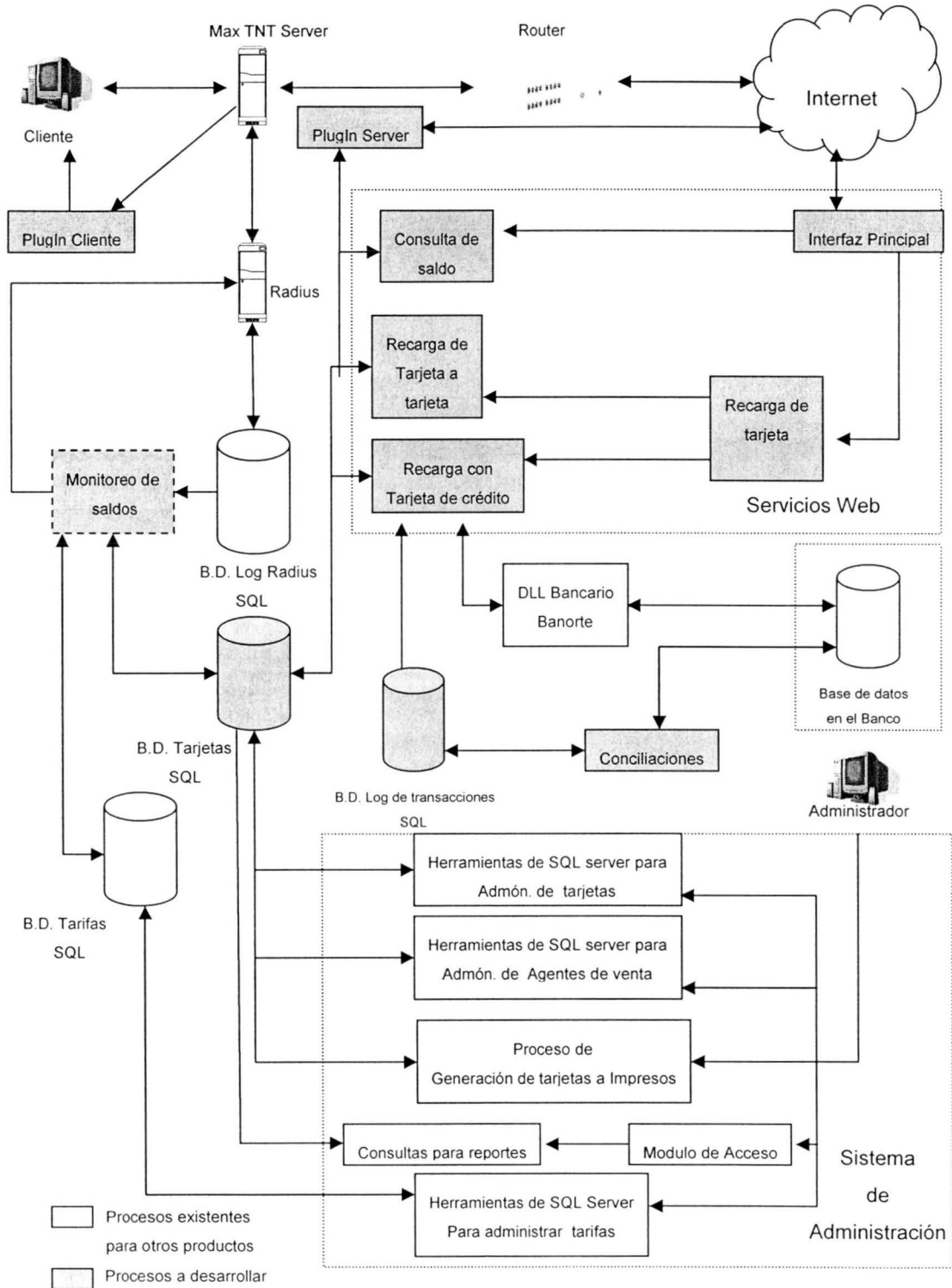


Figura 4.4 Arquitectura de Sistema

4.2.2 Plan de Laboratorio – Escenarios

A continuación se presentan los escenarios desarrollados:

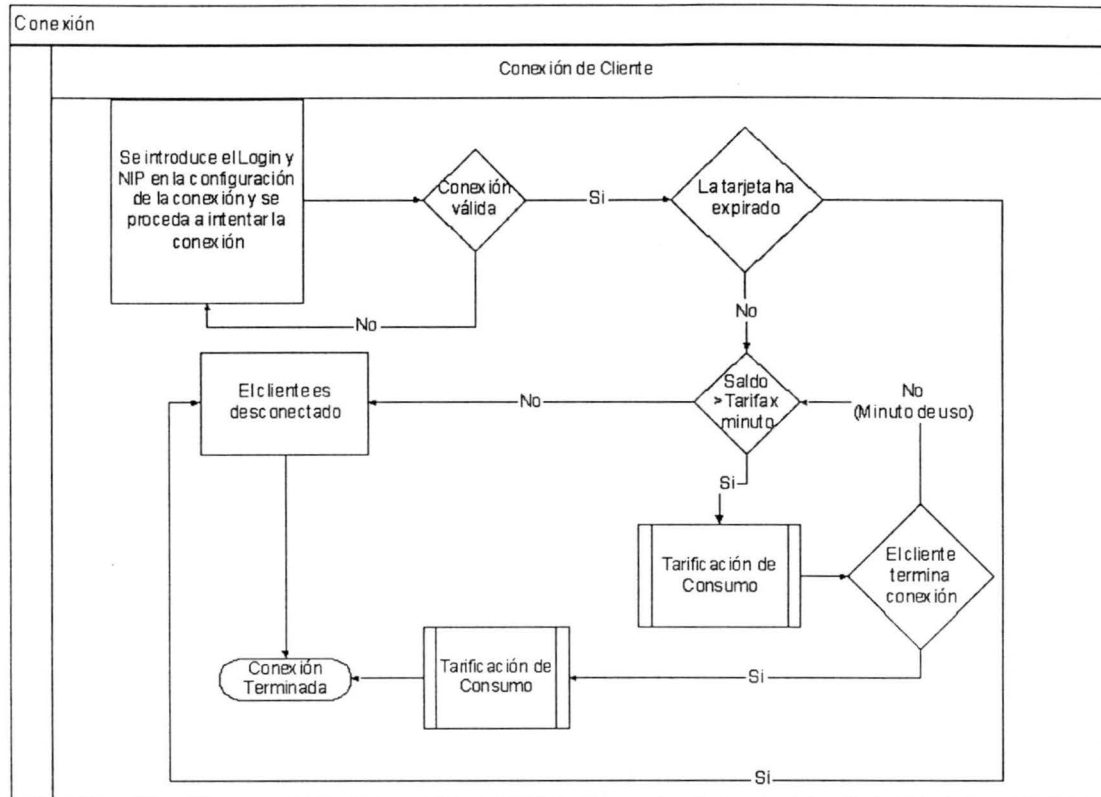


Figura 4.5 Escenario para conexión de cliente

Caso 1: conexión

Descripción	Muestra la forma en que el usuario realiza su conexión con su tarjeta de Internet prepago
Identificador	A.1
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Sistema Internet Prepago
Pre-condiciones	Tener una tarjeta de Internet prepago Equipo de cómputo con <i>modem</i> /línea telefónica
Acciones	Introducir datos Validar datos Tarificar Desconectar
Post-condiciones	Saldo = 0 Tarjeta Expirada Desconexión por parte del usuario
Incluye	C.1 Tarificación de consumo
Extensión	B.1.1 Tarjeta Física
Generalización	Ninguno

Tabla 4.17 Tabla para caso de conexión de cliente.

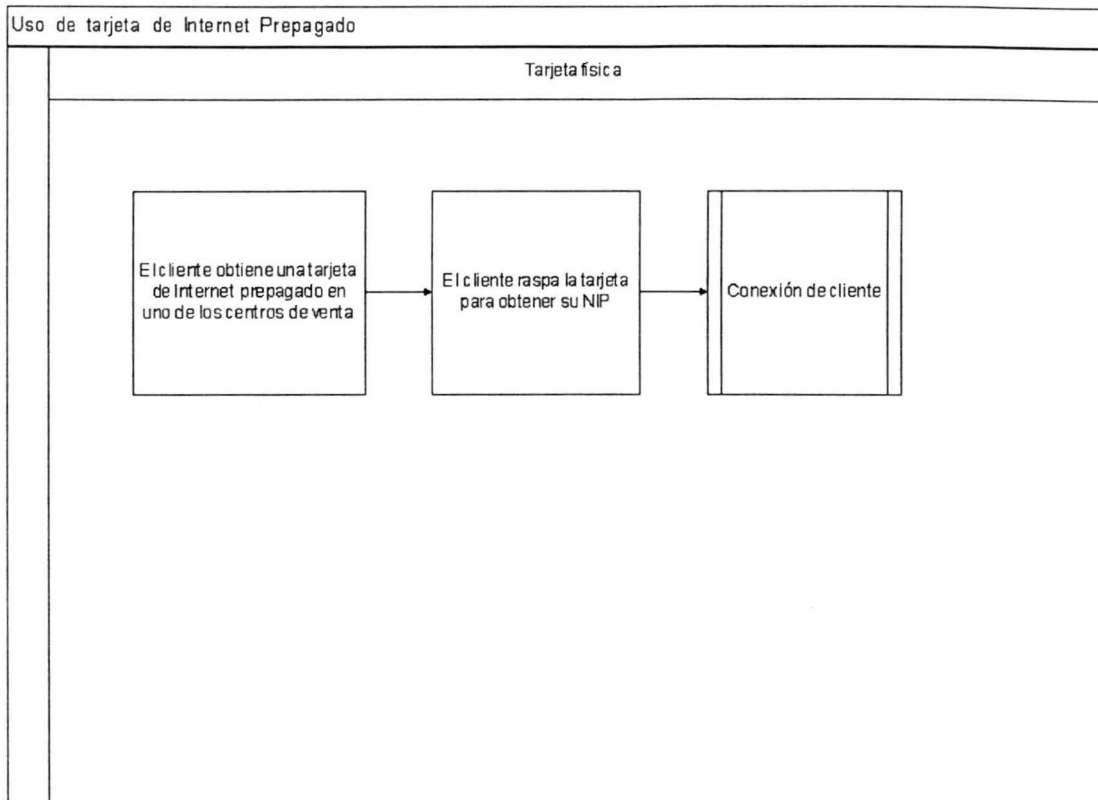


Figura 4.6 Escenario para uso de tarjeta física

Caso 2: Tarjeta Física

Descripción	Muestra como el usuario obtiene su tarjeta de Internet prepagado y realiza su conexión
Identificador	B.1.1
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Sistema Internet Prepagado
Pre-condiciones	Ninguna
Acciones	Comprar tarjeta Raspar tarjeta Conectar
Post-condiciones	Desconexión por parte del usuario
Incluye	A.1 Conexión
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.18 Caso para uso de tarjeta física.

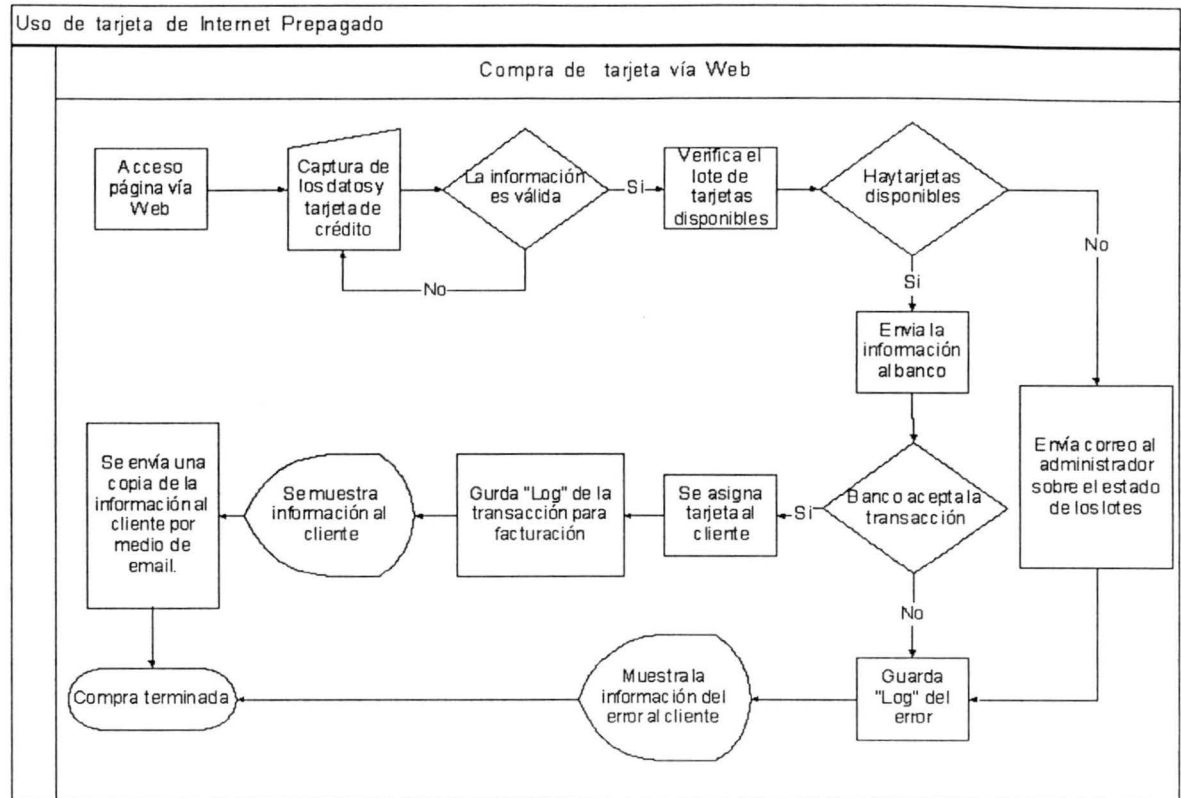


Figura 4.7 Escenario para compra de tarjeta vía web.

Caso 3: Compra de tarjeta vía Web

Descripción	Muestra como comprar una tarjeta de Internet prepagado vía Web con tarjeta de crédito
Identificador	B.1.2
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Servicio Web de Internet prepagado
Pre-condiciones	Tener una tarjeta de crédito
Acciones	Capturar Datos Validar datos Validar tarjeta de crédito Comprar tarjeta de Internet prepagado Desplegar Información de la compra
Post-condiciones	Tarjeta de crédito Inválida No existen tarjetas de Internet prepagado disponibles Compra realizada exitosamente
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.19 Caso para compra de tarjeta vía Internet.

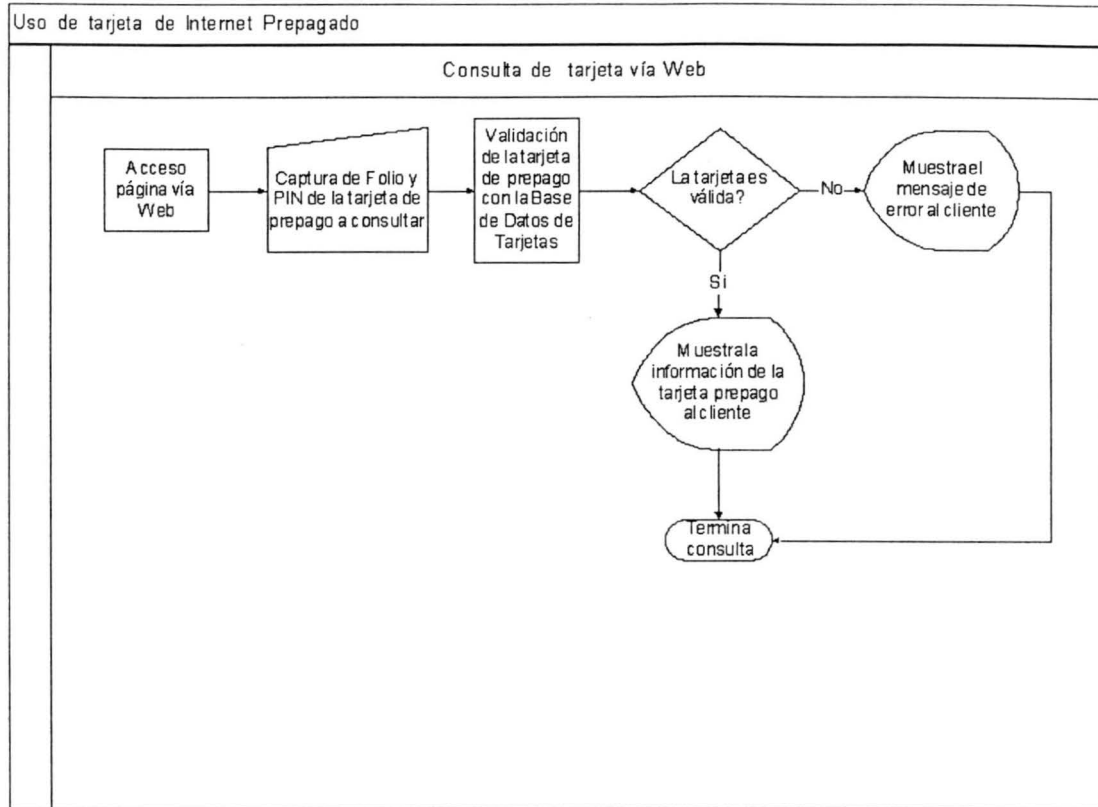


Figura 4.8 Escenario para consulta de tarjeta por Internet.

Caso 4: Consulta de tarjeta vía Web

Descripción	Muestra como consultar la información de una tarjeta de Internet prepago vía Web
Identificador	B.1.3
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Servicio Web de Internet prepago
Pre-condiciones	Tener una tarjeta de Internet Prepago
Acciones	Capturar Datos Validar datos Desplegar Información de la tarjeta
Post-condiciones	Tarjeta de Internet prepago Inválida Consulta realizada exitosamente
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.20 Caso para consulta de tarjeta por Internet

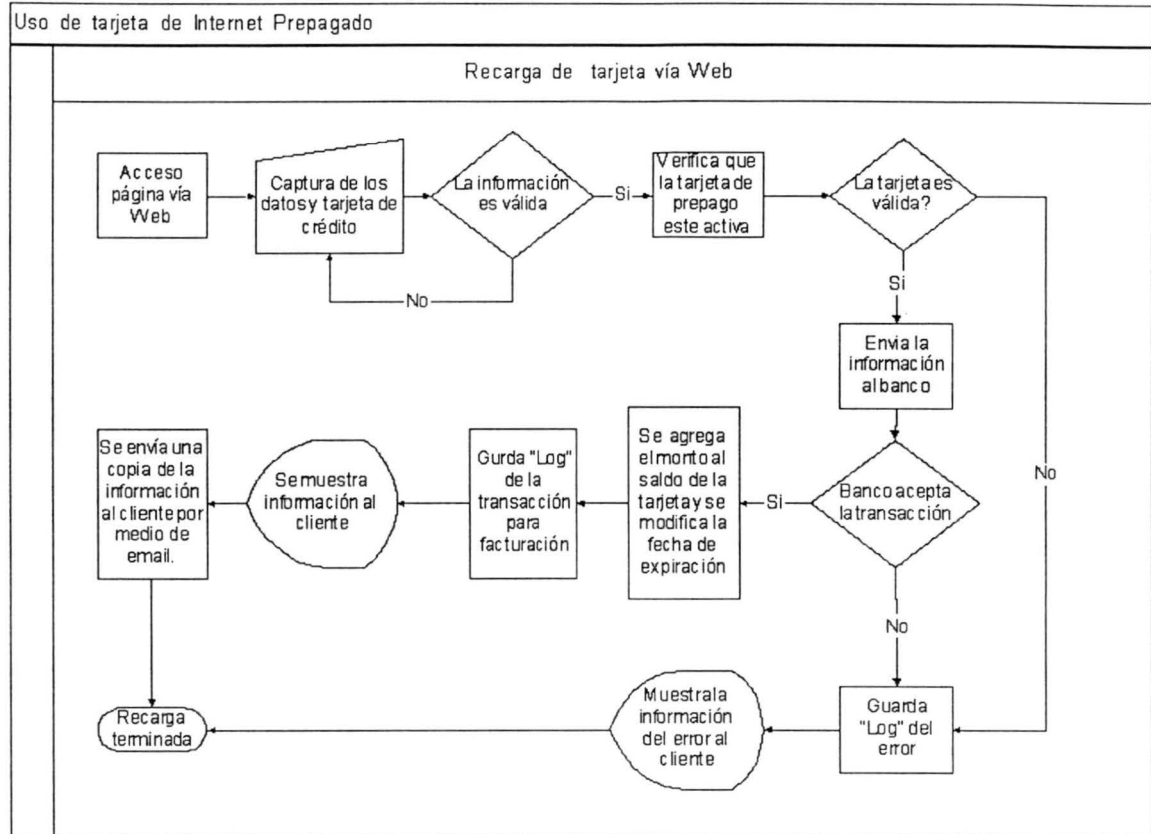


Figura 4.9 Escenario recarga de tarjeta por Internet.

Caso 5: Recarga de tarjeta vía Web

Descripción	Muestra como recargar una tarjeta de Internet prepago vía Web con una tarjeta de crédito
Identificador	B.1.4
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Servicio Web de Internet prepago
Pre-condiciones	Tener una tarjeta de Internet Prepago
Acciones	Capturar Datos Validar datos Desplegar Información de la tarjeta
Post-condiciones	Tarjeta de Internet prepago Inválida Consulta realizada exitosamente
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.21 Caso para recarga de tarjeta por Internet.

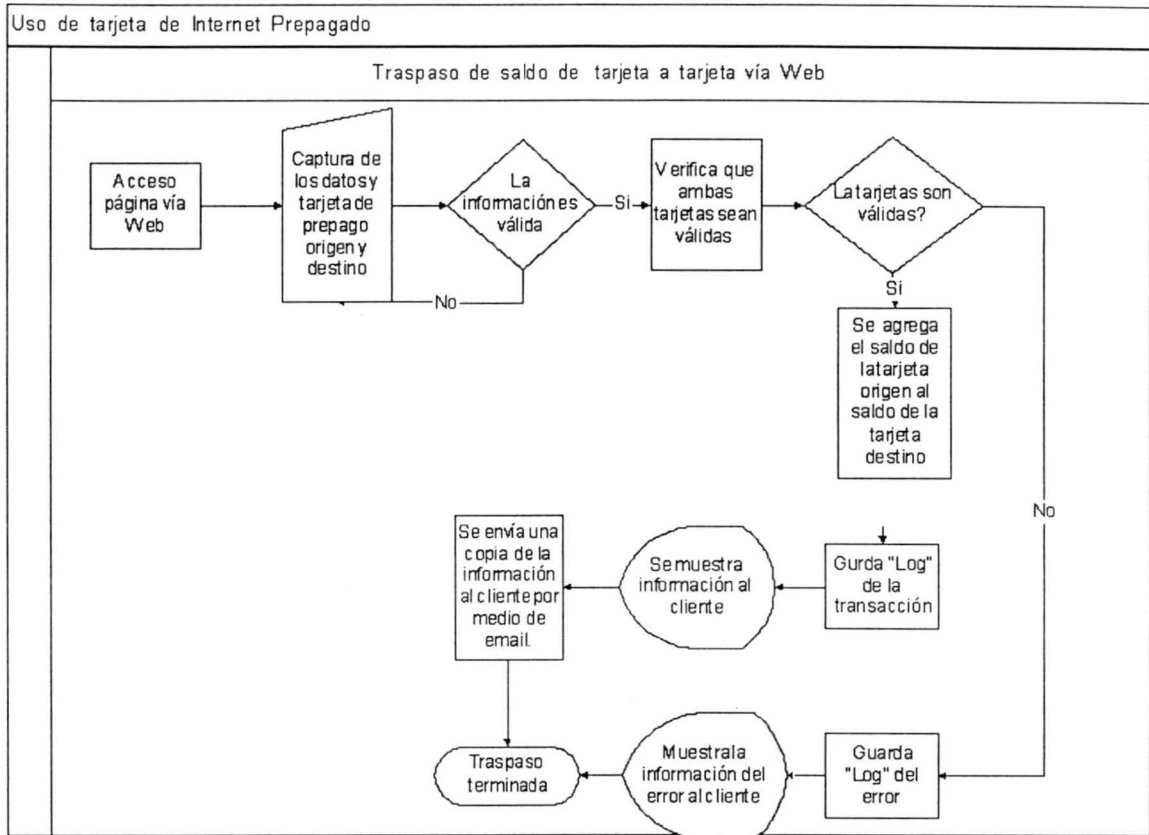


Figura 4.10 Escenario recarga de tarjeta por Internet.

Caso 6: Traspaso de saldo de tarjeta a tarjeta vía Web

Descripción	Muestra como traspasar el saldo de una tarjeta de Internet prepago a otra tarjeta vía Web
Identificador	B.1.5
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Servicio Web de Internet prepago
Pre-condiciones	Tener 2 tarjetas de Internet prepago
Acciones	Capturar Datos Validar datos Traspasar saldo Desplegar Información de la tarjeta
Post-condiciones	Tarjeta de Internet prepago Inválida Tarjeta expirada Consulta realizada exitosamente
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.22 Caso para traspaso de saldo de tarjeta a tarjeta por Internet

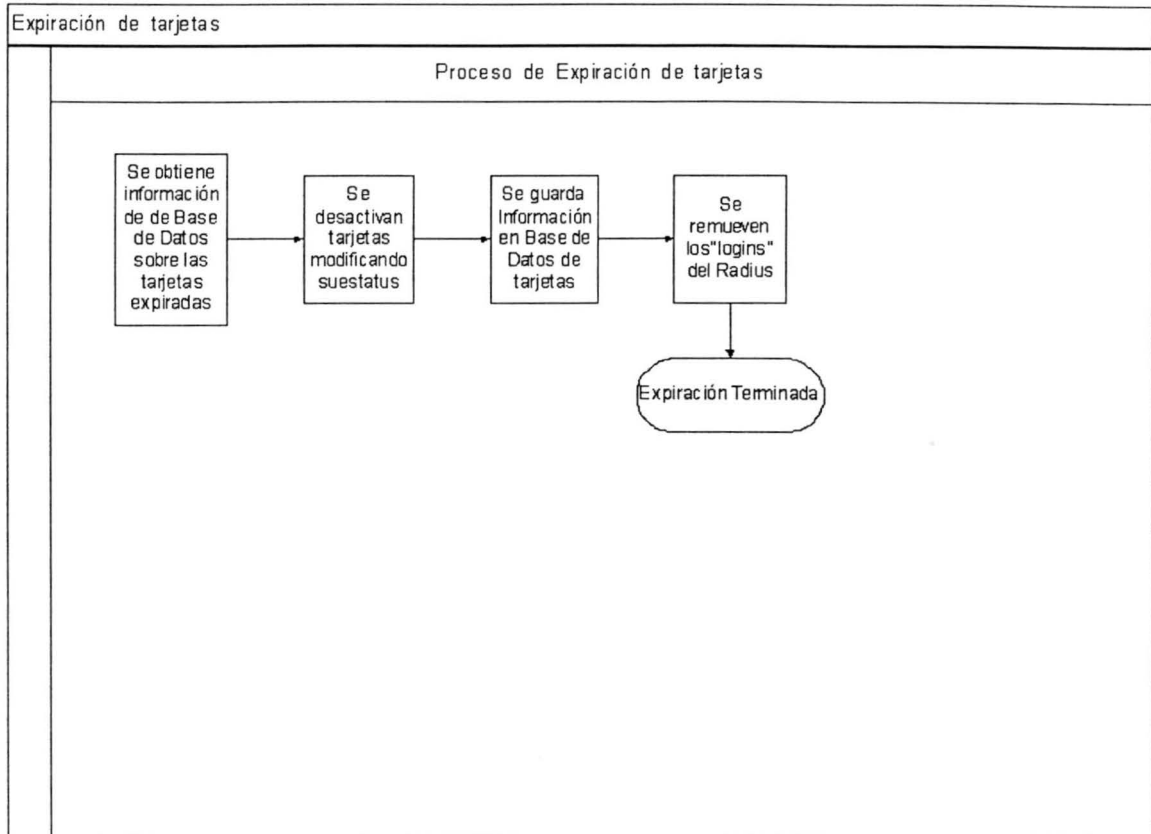


Figura 4.11 Escenario para expiración de tarjetas.

Caso 7: Expiración de tarjetas

Descripción	Muestra como el estatus de las tarjetas con fecha de vencimiento es modificado a Expiradas
Identificador	B.1.6
Autor	Fabián Orta
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Servicio Web de Internet prepago
Pre-condiciones	Tener 2 tarjetas de Internet prepago
Acciones	Obtener tarjetas Modificar estatus Remover logias
Post-condiciones	Tarjeta expiradas No existan tarjetas a modificar
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.23 Caso para expiración de tarjetas.

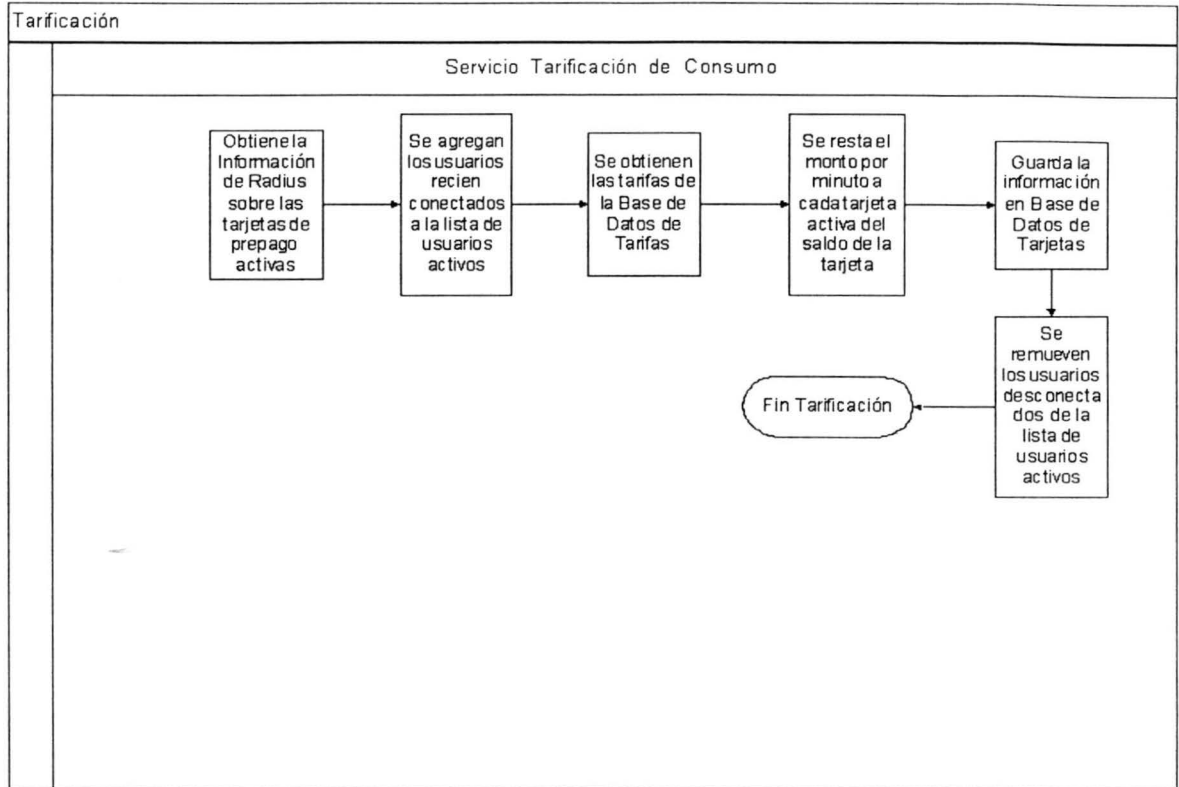


Figura 4.12 Escenario para expiración de tarjetas.

Caso 8: Tarificación

Descripción	Muestra como las tarjetas se les va deducir el saldo cuando se utilicen.
Identificador	B.1.7
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	02/Oct/2003
Actores	Servicio Web de Internet prepago
Pre-condiciones	Tener 1 tarjeta de Internet prepago con saldo.
Acciones	<ul style="list-style-type: none"> -Obtener información de Radius de conexiones y desconexiones de usuarios. -El usuario que se conecte se agrega a la lista de usuarios conectados. -Se obtiene información de la BD de tarifas necesaria para tarificación. -Monto se restado por minuto y se actualiza saldo de tarjeta. -Si el usuario se desconecta, se remueve de la lista de usuarios conectados. -Se repite proceso.
Post-condiciones	<ul style="list-style-type: none"> Tarjeta expiradas No existe una recarga durante su uso.
Incluye	Ninguna
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.24 Caso para tarificación de tarjetas.

4.3 Oportunidad de Negocio, Visión y Solución para 2do y 3er Ciclo de MSF

Antes de proceder al inciso 4.1.5, que explica los beneficios del proyecto, se explica las versiones anteriores del producto y la arquitectura de sistema a través de juntas realizadas en el mes de agosto 2003.

Se definió inicialmente en mayo 2003 un producto de tarjetas prepagadas de Internet que tuviera las siguientes características:

1. Generación de Lotes
2. Venta de Lotes
3. Cancelación de Lotes
4. Cancelación de Tarjetas
5. Generación Archivos para Impresor
6. Activación de Lotes vendidos
7. Asignación de Lotes para Venta en Internet
8. Asignación de Lotes para Venta en SAC
9. Monitoreo de Inventario
10. Asignación de tarjeta a un perfil vía SAC
11. Asignación de tarjeta a un perfil vía Internet
12. Expiración de Lotes
13. Expiración de Tarjetas (asignadas para venta en Internet y SAC)
14. Monitoreo de Saldos para desconexión
15. Reemplazo de Lotes y tarjetas
16. Aviso de saldo agotado para el usuario
17. Bonificación de Saldos por reclamos
18. Atención a quejas
19. Compra de Tarjetas vía Internet
20. Compra de Tarjetas vía SAC (Servicio de Atención a Clientes)
21. Auditoria de ventas vs. Pagos
22. Aprovisionamiento Financiero de los pagos
23. Creación de Perfiles
24. Actualización de información de perfiles vía Internet
25. Actualización de información de perfiles vía SAC
26. cancelación de perfiles
27. Depuración de perfiles por inactividad
28. Recarga de Tarjeta
29. Consulta de Saldos de Perfil

[Balboa, 2003]

Se definió un producto de tarjetas prepagadas Internet que requiere de un sistema administrativo y servicios para el cliente. El servicio administrativo requiere de generación de lotes de tarjetas, su impresión, venta y activación para su uso. Todo lo mencionado estuvo planeado hacerse desde su inicio, sin ninguna adaptación a otro proceso.

En la figura 4.1, los módulos y bases de datos que están sombreados, iban a ser desarrollados como parte de este proyecto. Lo que se encuentra en blanco eran bases de datos del banco y del servidor Radius, que ya están disponibles y se relacionan con otros procesos.

En agosto 2003 se tuvo una junta para presentar al departamento de ventas el proyecto de prepago Internet y recibir retroalimentación. La creación de perfiles requería más tiempo de lo esperado debido a la asignación de tarifa al perfil. Marcatel maneja una tarifa determinada para cada producto que ofrece. Al tratar de relacionar la tarjeta con un perfil, se observó que se contaba con tres diferentes tarjetas con diferente tarifa cada una. Si el cliente decide comprar una tarjeta de \$100 con tarifa \$0.70. Al asignarlo al perfil se le daba la tarifa \$0.70. El cliente podía comprar otra tarjeta de \$500 con tarifa de \$0.40 y ligarla a su perfil. El problema se presentó al tratar de determinar que tarifa se le asigna. No se podía dar la más barata o cara debido a problemas de facturación. Al final, perfiles se removió del producto y se decidió mantener una cuenta con usuario y contraseña numérico sin ligar a ningún perfil. También, se concluyó que este producto de Internet prepago sería más aprovechado si se junta con la tarjeta prepagada de voz [Ventas, 2003]

Con los perfiles asignados a un proyecto próximo, se estableció otra junta con el personal administrativo y responsable del sistema de tarjetas prepagadas de voz Ing. Ruth Leal para dar una solución a la fusión del producto de Internet y voz prepago. Durante la junta se determinó que los lotes y tarjetas que se venden para voz, se manejan de la misma manera que los objetivos establecidos para Internet prepago. Para fusionar los productos en uno solo, se decidió reutilizar la información de tarjetas voz para tarjetas Internet. Esto evitó inconsistencias de datos para definir el producto nuevo de voz e Internet. [Leal, 2003]

Finalmente, se determinó que las tarifas a utilizar para prepago Internet no diferían mucho de las utilizadas para cobrar el uso del producto *Dial-Up*. (Ver inciso 1.2 Productos y Servicios en capítulo 1, Figura 1.2) No era necesario determinar otra serie de tarifas para Internet prepago, se podían utilizar las mismas tarifas que utiliza el producto *Dial-Up* y aprovecharlo para los dos sistemas, en vez de uno por producto. [Pereyra,2003]

Se establecen dos ciclos más para la fase previsión y planeación del *MSF*:

- Segundo ciclo: Creación del sistema administrativo de lotes para tarjetas de voz y datos.
- Tercer ciclo: Creación de perfiles para usuarios

4.3.1 Segundo Ciclo MSF, Sistema administrativo de tarjetas

A continuación se presenta las partes restantes del documento de alcance y estrategia para el sistema administrativo de tarjetas.

4.3.1.1 Beneficios

A continuación se muestra el análisis de beneficios para segundo ciclo de fases *MSF* que representa el sistema administrativo de tarjetas:

Detalle los costos tangibles del Proyecto:

	Costo en unidades de medición apropiadas	Explicación	Traducido a Dinero	Relación con Indicadores
Tiempo:	700 Horas	Se toma en cuenta personal con título profesional, no a nivel practicante.	\$20,000	Tiempo en el que se desarrollarán los módulos del sistema.
Personal Marcatel	2 personas	Basado en el tiempo dedicado de las personas encargadas del proyecto en Marcatel(20 horas al mes en promedio)	\$15,000	Se deberán tomar en cuenta los puntos clave para cumplir con el desarrollo del sistema
Materiales Oficina	2 Paquetes de hojas de máquina y 2 cartuchos de tinta para la impresora	Tomando en cuenta el material utilizado para el análisis , diseño y desarrollo del sistema	\$500	Los recursos son asignados por la empresa para el uso de los desarrolladores
Hardware	1 Servidor, 1 Radius Server, 2 computadoras	En los servidores se alojarán el sistemas y las bases de datos ocupados por el mismo.	\$4,000	En los servidores se concentrará la información de la información de los tarjetas prepagadas para su administración.
Software	Windows 2000 Server, SQL Server 2000, Visual Studio .NET, Microsoft Office 2000 , Microsoft Visio 2000, Erwin	El desarrollo será basado en .Net (ASP.NET y VB.NET) con base de datos en SQL 2000. La documentación y diagramas serán realizados con Office, Edwin y Visio 2000. Todo el software antes mencionado se encuentra disponible y con licencias dentro de la empresa	\$0	Con el uso de ASP.NET y VB.NET se permitirá tener accesos vía Web y el desarrollo de objetos y servicios utilizados por el sistema
Servicios	Agua, Luz e instalaciones	Tomando en cuenta el porcentaje de servicios ocupados por el sistema y el personal del proyecto	\$900	Los servidores sen encontrarán en el edificio de Marcatel Internacional

Costo Total \$40,400

Tabla 4.25 Costos tangibles para sistema administrativo de tarjetas

Detalle de los costos de Operación(Operación semestral)

	Descripción	Cantidad
Servicios	Cantidad derivada de los servicios ocupados para que el sistema opere(Luz)	\$4,000
Operadores	Sueldo de las personas encargadas de operar el sistema de lotes	\$12,000
Mantenimiento	Costo de mantenimiento del sistema	\$7,000
Costo Total		\$23,000

Tabla 4.26 Costos de operación para sistema administrativo de tarjetas

Explique la implicación de que el proyecto no se lleve a cabo:

	Explicación
Tiempo:	No haya problemas durante el desarrollo del sistema administrativo de lotes.
Personas:	Se tendría que destinar 1 persona para la instalación y operación del sistema administrador.
Redes:	Seguiría existiendo la misma infraestructura con la que actualmente se cuenta.

Tabla 4.27 Implicaciones para que proyecto no se lleve a cabo para sistema administrativo de tarjetas.

Detalle los beneficios tangibles del proyecto:

	Beneficio en unidades de medición	Explicación del Cálculo	Traducido a Dinero
Tiempo:	200	Ahorro en horas de trabajo en la generación de lotes por semestre.	\$100,000
Materiales:	0	No existe ningún beneficio ya que el sistema sigue utilizando los mismos materiales	\$0
Redes:	0	No existe ningún beneficio en Redes ya que se utilizará la actual.	\$0
Software:	0	No existe ningún beneficio en Software ya que se utilizará el actual.	\$0

Beneficio Total: \$100,000

Tabla 4.28 Beneficios de sistema administrativo de tarjetas

Retorno sobre la Inversión ROI del proyecto:

Período (6 meses)	Costos del período	Beneficios del período	Beneficio Neto
1 semestre	\$40,400	\$0	-\$40,400
2 semestre	\$23,000	\$100,000	\$77,000
3 semestre	\$23,000	\$100,000	\$77,000
4 semestre	\$23,000	\$100,000	\$77,000
5 semestre	\$23,000	\$100,000	\$77,000
6 semestre	\$23,000	\$100,000	\$77,000
Total	\$155,400	\$500,000	\$344,600

Tabla 4.29 Retorno de Inversión para sistema administrativo de tarjetas

4.3.1.2 Análisis de Usuarios

Para la segunda vuelta del ciclo *MSF* que involucra el sistema administrativo se considera un solo usuario, en este caso el administrador que maneja las tarjetas prepagadas.

4.3.1.3 Requerimientos

Los requerimientos para la segunda vuelta del ciclo *MSF* son las mismas. Las actividades excluidas del proyecto actual de todos los requerimientos que se encuentran en el inciso 4.1.6.

4.3.1.4 Equipo de trabajo, distribución de competencias y responsabilidades

Las actividades para el segundo ciclo del *MSF* de administración de tarjetas son las siguientes:

No. Actividad	Actividad
1	Junta para definir funciones del proyecto
2	Junta para definir el producto
3	Diseño de la Arquitectura del Sistema
4	Diseño del Modelo de Base de datos(Entidad relación)
5	Elaboración de las Pruebas de Laboratorio
6	Elaboración del Plan de trabajo
7	Elaboración de la Matriz de riesgos
8	Elaboración del Plan de contingencia
9	Aprobación de la Planificación y Diseño de Arquitectura
10	Administración de lotes de tarjetas
11	Cancelación de inventarios.
12	Creación de catálogos de proveedores
13	Construcción de catálogos de tarjetas físicas
14	Asignación de tarjetas para departamento de Servicio de Atención a Clientes
15	Selección de entorno de pruebas piloto
16	Realización de pruebas sobre la maqueta
17	Captura de incidencias durante la prueba
18	Aprobación de la implementación de Arquitectura Final del sistema.
19	Elaboración del Plan de despliegue
20	Despliegue
21	Captura de Incidencia 2ª versión
22	Elaboración de Documento de mejoras
23	Entrega de documentos establecidos
24	Entrega del proyecto

Tabla 4.30 Actividades a realizar para segunda vuelta MSF

A continuación se define la distribución de actividades para el segundo ciclo:

No. Actividad	Actividad	Asignación
1	Junta para definir funciones del proyecto	Gerardo Canepa Fabián Orta
2	Junta para definir el producto	Gerardo Canepa Fabián Orta
3	Diseño de la Arquitectura del Sistema	Gerardo Canepa Fabián Orta
4	Diseño del Modelo de Base de datos(Entidad relación)	Gerardo Canepa Fabián Orta
5	Elaboración de las Pruebas de Laboratorio	Gerardo Canepa Fabián Orta
6	Elaboración del Plan de trabajo	Gerardo Canepa Fabián Orta
7	Elaboración de la Matriz de riegos	Gerardo Canepa Fabián Orta
8	Elaboración del Plan de contingencia	Gerardo Canepa Fabián Orta Gerardo Canepa Fabián Orta
9	Aprobación de la Planificación y Diseño de Arquitectura	Gerardo Canepa Fabián Orta
10	Administración de lotes de tarjetas	Fabián Orta
11	Cancelación de inventarios de tarjetas físicas por robo o pérdida.	Gerardo Canepa Fabián Orta
12	Creación de catálogos de proveedores	Gerardo Canepa
13	Construcción de catálogos de tarjetas físicas	Fabián Orta
14	Asignación de tarjetas para departamento de Servicio de Atención a Clientes	Gerardo Canepa Fabián Orta
15	Selección de entorno de pruebas piloto	Gerardo Canepa Fabián Orta
16	Realización de pruebas sobre la maqueta	Gerardo Canepa Faiban Orta
17	Captura de incidencias durante la prueba	Gerardo Canepa Fabián Orta
18	Aprobación de la implementación de Arquitectura Final del sistema.	Gerardo Canepa Fabián Orta
19	Elaboración del Plan de despliegue	Gerardo Canepa Fabián Orta
20	Despliegue	Fabián Orta
21	Captura de Incidencia 2ª versión	Gerardo Canepa
22	Elaboración de Documento de mejoras	Gerardo Canepa Fabián Orta
23	Entrega de documentos establecidos	Gerardo Canepa Fabián Orta
24	Entrega del proyecto	Gerardo Canepa Fabián Orta

Tabla 4.31 Distribución de actividades para segunda vuelta

4.3.1.5 Arquitectura de Sistema

La arquitectura de sistema quedó definida de la siguiente forma para la segunda vuelta del ciclo *MSF*:

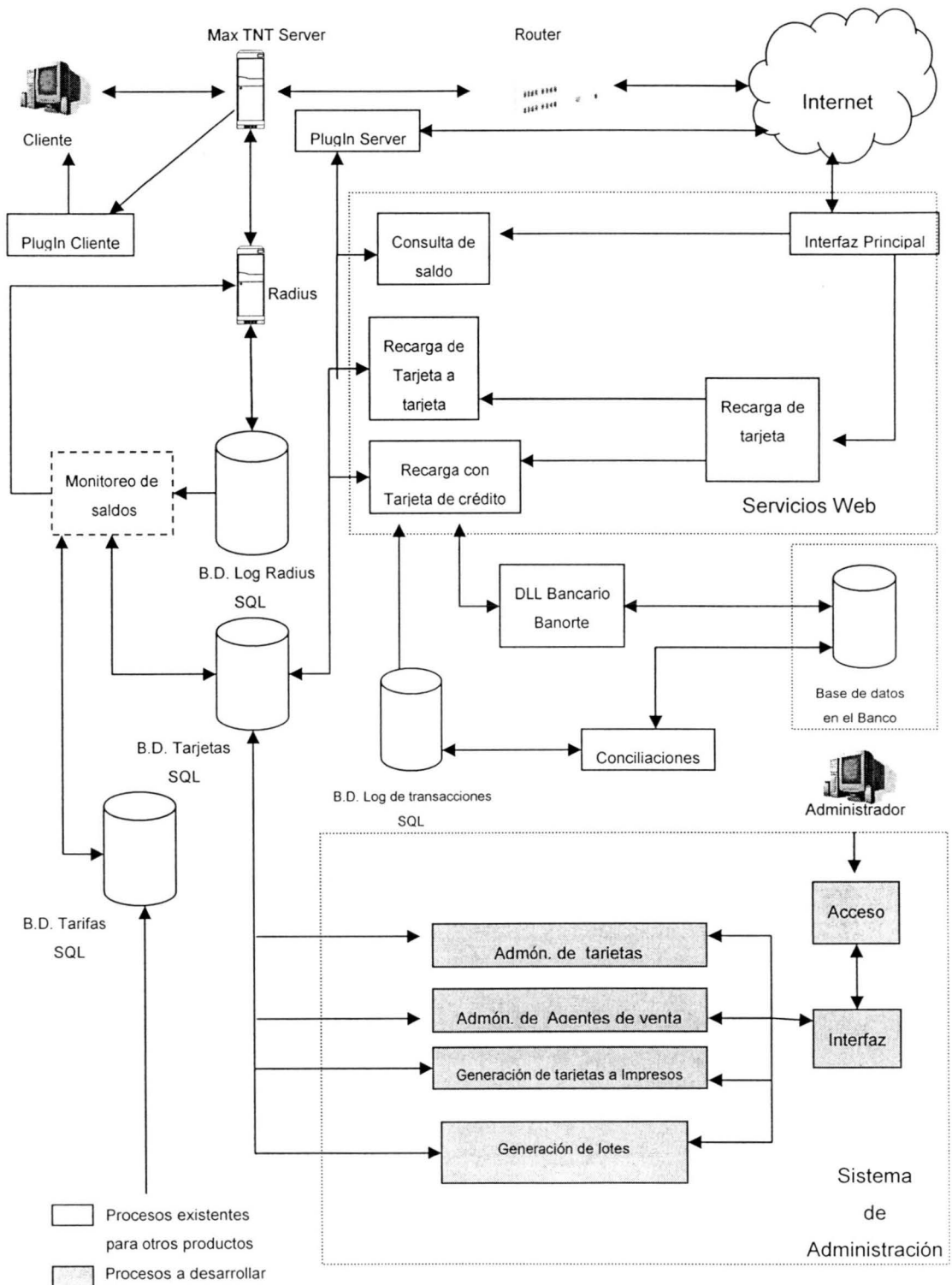


FIGURA 4.13 Arquitectura definida inicialmente para proyecto.

Solo se describen los procesos a desarrollar en el segundo ciclo del *MSF*:

Acceso:

Válida y permite el uso al administrador del sistema.

Interfaz:

Grupo de pantallas que permiten al administrador crear, editar o borrar la información del sistema.

Administración de tarjetas:

Permite crear, editar y borrar la información de las tarjetas de prepago dentro del sistema.

Administración de agentes de ventas:

Permite crear, editar y borrar la información de los agentes de ventas.

Generación de tarjetas a impresos:

Módulo que permite generar el archivo con la información necesaria para imprimir físicamente las tarjetas de prepago para voz y Internet

Generación de lotes:

Módulo que permite lotes de tarjetas para ser vendidas a los agentes de venta.

4.3.1.6 Plan de Laboratorio

A continuación se presentan los escenarios para la segunda vuelta del ciclo MSF, administración de tarjetas:

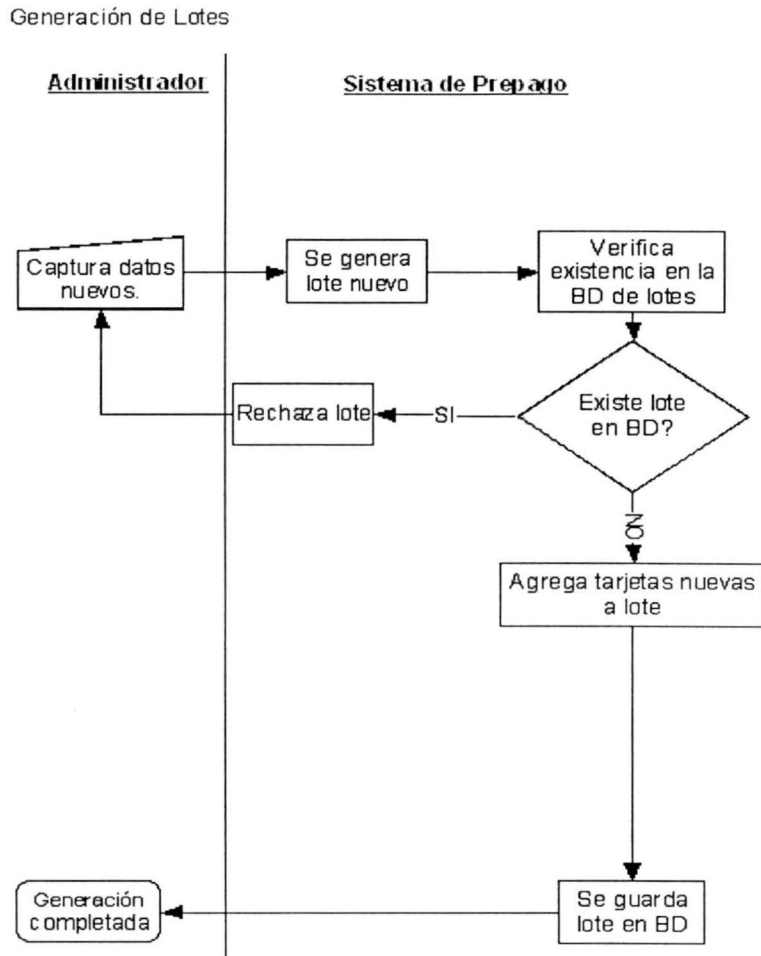


Figura 4.14 Escenario para Generación de Lotes

Caso 1: Generación de Lotes

Descripción	Muestra la forma en que los lotes de dan de alta en la Base de Datos de Tarjetas
Identificador	L.1
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Administrador, Sistema Internet Prepagado
Pre-condiciones	Ninguna
Acciones	Introducir datos Validar datos Generar Lote Terminar
Post-condiciones	Lote Generado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.32 Caso para Generación de Lotes.

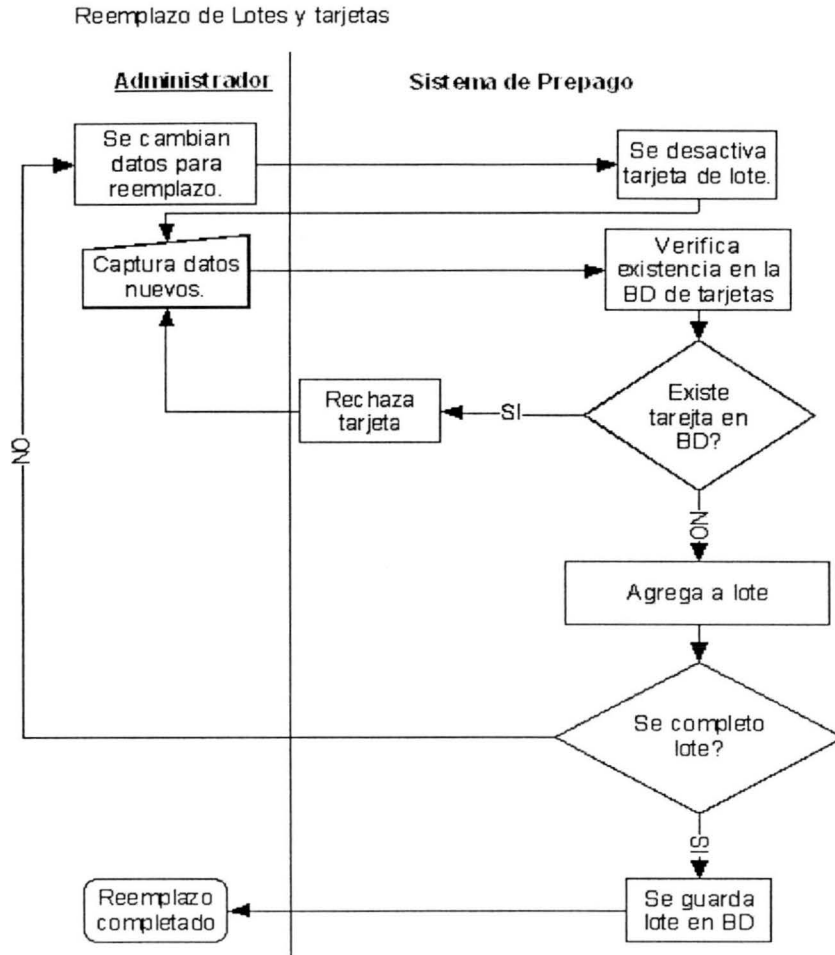


Figura 4.15 Escenario para Reemplazo de Lotes

Caso 2: Reemplazo de Lotes

Descripción	Muestra como las tarjetas que pertenecen a un lote específico se pueden asignar a otro lote
Identificador	L.2
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Administrador, Sistema Internet Prepagado
Pre-condiciones	Tener dos lotes existentes
Acciones	Introducir datos Validar datos Reemplazar tarjetas de Lote Terminar
Post-condiciones	Lote Reemplazado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.33 Caso para Reemplazo de Lotes.

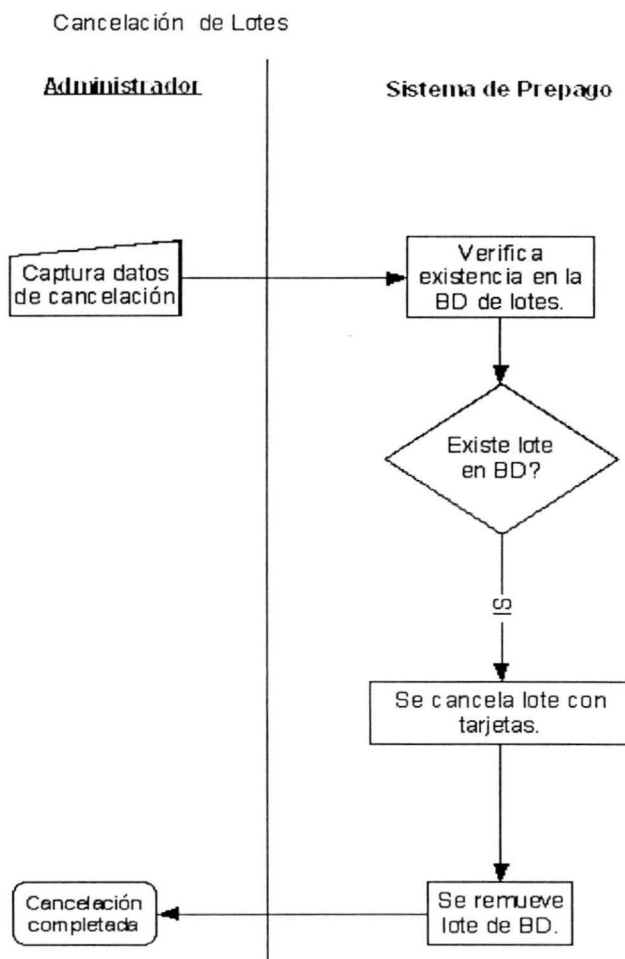


Figura 4.16 Escenario para Cancelación de Lotes

Caso 3: Reemplazo de Lotes

Descripción	Muestra como cancelar un lote de la Base de datos de tarjetas
Identificador	L.3
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Administrador, Sistema Internet Prepagado
Pre-condiciones	Tener un lote existente
Acciones	Introducir datos Validar datos Cancelar Lote Terminar
Post-condiciones	Lote Cancelado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.34 Caso para Cancelación de Lotes.

4.3.2 Tercer Ciclo MSF, Sistema de perfiles

4.3.2.1 Beneficios

A continuación se muestra el análisis de beneficios para tercer ciclo de fases MSF que representa el sistema de perfiles:

Detalle los costos tangibles del Proyecto:

	Costo en unidades de medición apropiadas	Explicación	Traducido a Dinero	Relación con Indicadores
Tiempo:	360 Horas	Se toma en cuenta personal con título profesional, no a nivel practicante.	\$17,000	Tiempo en el que se desarrollarán los módulos del sistema.
Personal Marcatel	2 personas	Basado en el tiempo dedicado de las personas encargadas del proyecto en Marcatel(20 horas al mes en promedio)	\$10,000	Se deberán tomar en cuenta los puntos clave para cumplir con el desarrollo del sistema
Materiales Oficina	2 Paquetes de hojas de máquina y 2 cartuchos de tinta para la impresora	Tomando en cuenta el material utilizado para el análisis , diseño y desarrollo del sistema	\$200	Los recursos son asignados por la empresa para el uso de los desarrolladores
Hardware	1 Servidor,	En los servidores se alojarán el sistemas y las bases de datos ocupados por el mismo.	\$4,000	En los servidores se concentrará la información de la información de los perfiles
Software	Windows 2000 Server, SQL Server 2000, Visual Studio .NET, API de Banco, ACS Radius, API Radius, Microsoft Office 2000 , Microsoft Visio 2000, Erwin	El desarrollo será basado en .Net (ASP.NET y VB.NET) con base de datos en SQL 2000. La documentación y diagramas serán realizados con Office, Edwin y Visio 2000. Todo el software antes mencionado se encuentra disponible y con licencias dentro de la empresa	\$0	Con el uso de ASP.NET y VB.NET se permitirá tener accesos vía Web y el desarrollo de objetos y servicios utilizados por el sistema
Servicios	Agua, Luz e instalaciones	Tomando en cuenta el porcentaje de servicios ocupados por el sistema y el personal del proyecto	\$300	Los servidores sen encontrarán en el edificio de Marcatel Internacional

Costo Total \$31,500

Tabla 4.35 Costos tangibles de proyecto para perfiles

Detalle de los costos de Operación (Operación semestral)

	Descripción	Cantidad
Servicios	Cantidad derivada de los servicios ocupados para que el sistema opere(Luz)	\$2,000
Operadores	Sueldo de las personas encargadas de operar el sistema de perfiles	\$10,000
Mantenimiento	Costo de mantenimiento del sistema	\$5,000

Tabla 4.36 Costos de operación para perfiles

Explique la implicación de que el proyecto no se lleve a cabo:

	Explicación
Tiempo:	No haya problemas durante el desarrollo de perfiles que detengan el desarrollo.
Personas:	Se tendría que destinar 1 personas para la instalación y operación del generación de perfiles.

Tabla 4.37 Implicaciones para que proyecto no se lleve a cabo para sistema de perfiles.

Detalle los beneficios tangibles del proyecto:

	Beneficio en unidades de medición	Explicación del Cálculo	Traducido a Dinero
Ventas:	200 clientes mas por semestre	la generación de perfiles da otra opción más al cliente que utiliza Internet Prepagado.	\$130,000
Materiales:	0	No existe ningún beneficio ya que el sistema sigue utilizando los mismos materiales	\$0
Redes:	0	No existe ningún beneficio en Redes ya que se utilizará la actual.	\$0

Beneficio Total: \$130,000

Tabla 4.38 Beneficios de sistema de perfiles

Retorno sobre la Inversión ROI del proyecto:

Período (6 meses)	Costos del período	Beneficios del período	Beneficio Neto
1 semestre	\$31,500	\$0	-\$31,500
2 semestre	\$17,000	\$130,000	\$113,000
3 semestre	\$17,000	\$130,000	\$113,000
4 semestre	\$17,000	\$130,000	\$113,000
5 semestre	\$17,000	\$130,000	\$113,000
6 semestre	\$17,000	\$130,000	\$113,000
Total	\$116,500	\$650,000	\$533,500

Tabla 4.39 Retorno de Inversión para sistema de perfiles.

4.3.2.2 Análisis de Usuarios

Para la tercera vuelta del ciclo *MSF* que involucran los perfiles se tomaron en cuenta el administrador y cliente. (Ver inciso 4.1.5)

4.3.2.3 Requerimientos

Los requerimientos para la tercera vuelta del ciclo *MSF* son las mismas. Las actividades excluidas del proyecto actual de todos los requerimientos que se encuentran en el inciso 4.1.6.

4.3.2.4 Equipo de trabajo, distribución de competencias y responsabilidades

Las actividades para el tercer ciclo del *MSF* de uso de perfiles son las siguientes:

No. Actividad	Actividad
1	Junta para definir funciones del proyecto
2	Junta para definir el producto
3	Diseño de la Arquitectura del Sistema
4	Diseño del Modelo de Base de datos(Entidad relación)
5	Elaboración de las Pruebas de Laboratorio
6	Elaboración del Plan de trabajo
7	Elaboración de la Matriz de riegos
8	Elaboración del Plan de contingencia
9	Aprobación de la Planificación y Diseño de Arquitectura
10	Construcción de tabla perfiles
11	Creación de correos electrónicos y páginas <i>web</i>
12	Activación de perfiles
13	Cancelar perfiles
14	Ligar perfil a tarjeta prepagada Internet
15	Selección de entorno de pruebas piloto
16	Realización de pruebas sobre la maqueta
17	Captura de incidencias durante la prueba
18	Aprobación de la implementación de Arquitectura Final del sistema.
19	Elaboración del Plan de despliegue
20	Despliegue
21	Captura de Incidencia 2ª versión
22	Elaboración de Documento de mejoras
23	Entrega de documentos establecidos
24	Entrega del proyecto

Tabla 4.40 Actividades a realizar para tercer vuelta MSF

A continuación se define la distribución de actividades para el tercer ciclo:

No. Actividad	Actividad	Asignación
1	Junta para definir funciones del proyecto	Gerardo Canepa Fabián Orta
2	Junta para definir el producto	Gerardo Canepa Fabián Orta
3	Diseño de la Arquitectura del Sistema	Gerardo Canepa Fabián Orta
4	Diseño del Modelo de Base de datos(Entidad relación)	Gerardo Canepa Fabián Orta
5	Elaboración de las Pruebas de Laboratorio	Gerardo Canepa Fabián Orta
6	Elaboración del Plan de trabajo	Gerardo Canepa Fabián Orta
7	Elaboración de la Matriz de riegos	Gerardo Canepa Fabián Orta
8	Elaboración del Plan de contingencia	Gerardo Canepa Fabián Orta
9	Aprobación de la Planificación y Diseño de Arquitectura	Gerardo Canepa Fabián Orta
10	Construcción de tabla perfiles	Gerardo Canepa
11	Activación de perfiles	Gerardo Canepa Fabián Orta Ing. Erich Von Hauske
12	Cancelar perfiles	Fabián Orta
13	Ligar perfil a tarjeta prepagada Internet	Gerardo Canepa Fabián Orta

Tabla 4.41 Distribución de actividades para tercera vuelta.

4.3.2.5 Arquitectura de Sistema

La arquitectura de sistema quedó definida de la siguiente forma para la tercera vuelta del ciclo *MSF* para perfiles de usuarios:

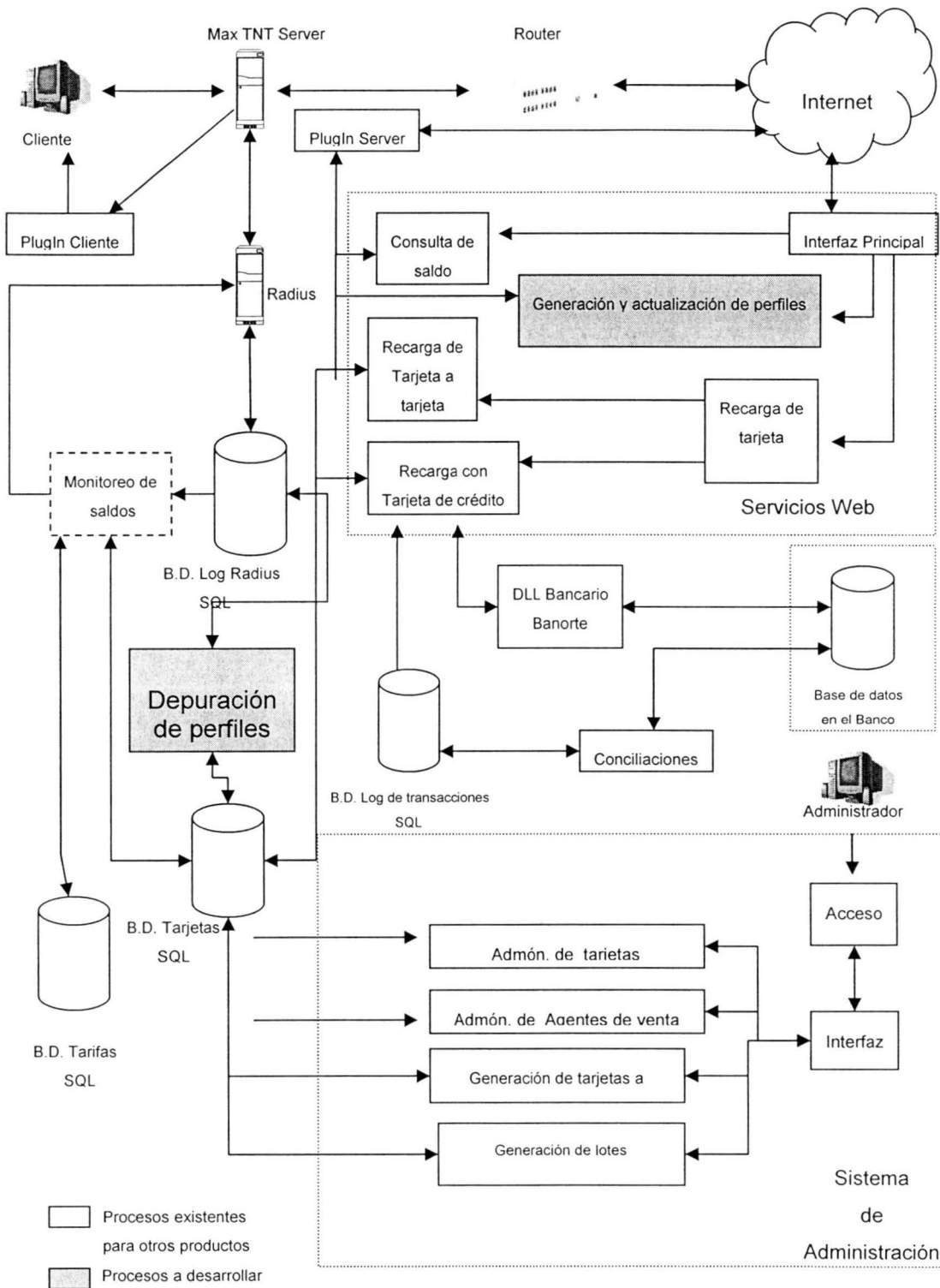


FIGURA 4.17 Arquitectura definida para tercera vuelta del ciclo MSF.

Solo se describen los procesos a desarrollar en el tercer ciclo del *MSF*.

Generación y actualización de perfiles:

Permite al usuario generar su propio perfil para asignar tarjetas prepagadas y utilizarlas. También permite modificar la información de un perfil existente.

Depuración de perfiles:

Proceso que depura las cuentas que no han sido utilizadas en más de 90 días.

4.3.2.6 Plan de Laboratorio

A continuación se presentan los escenarios para la tercera vuelta del ciclo *MSF* de generación de perfiles:

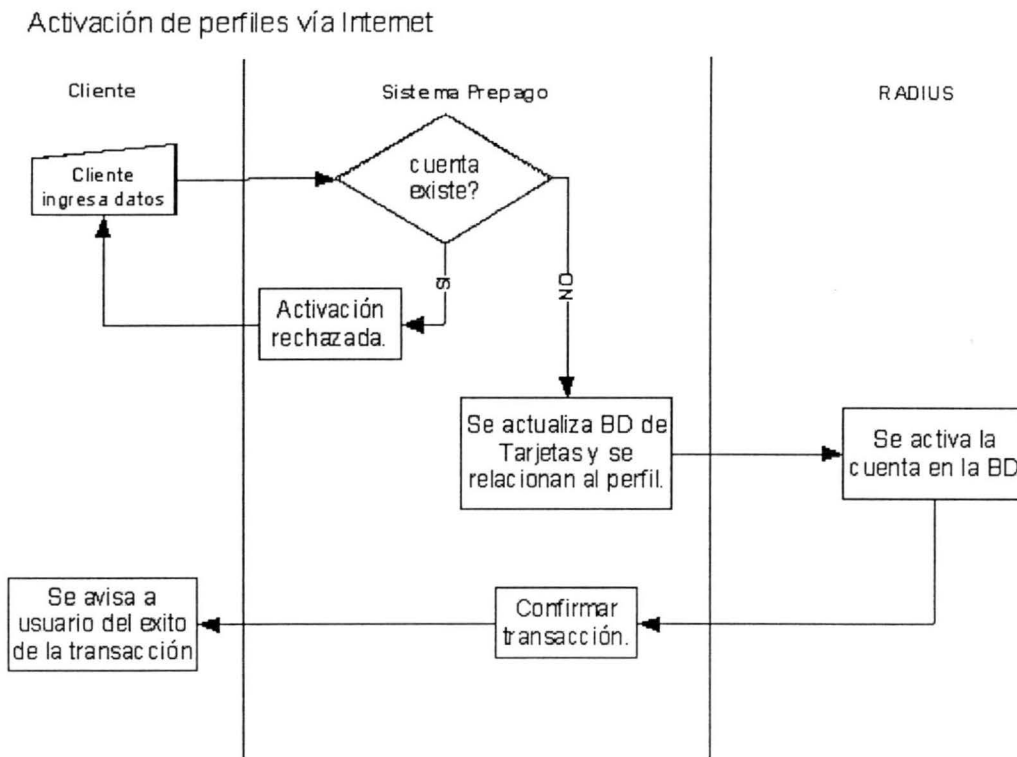


Figura 4.18 Escenario para Activación de Perfiles

Caso 1: Activación de Perfiles

Descripción	Muestra como dar de alta un perfil
Identificador	P.1
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Sistema Prepago, Radius
Pre-condiciones	Ninguno
Acciones	Introducir datos Validar información de cuenta Activar Perfil y almacenar en BD Aviso de éxito o rechazo
Post-condiciones	Perfil creado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.42 Caso para Activación de Perfiles.

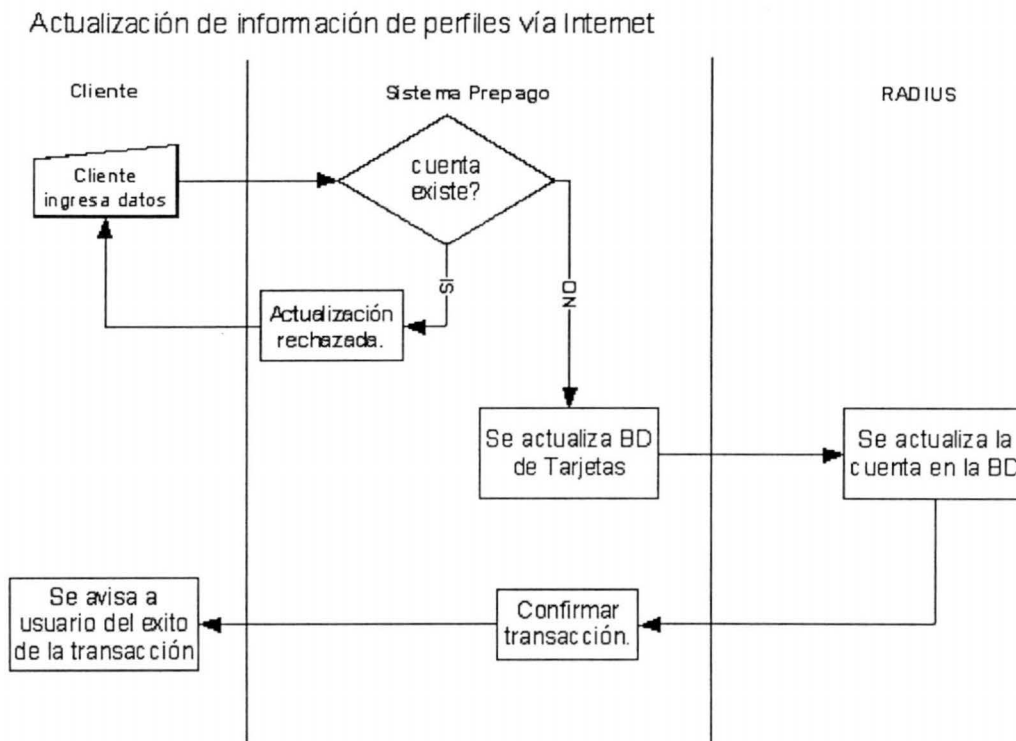


Figura 4.19 Escenario para Actualización de Perfiles

Caso 2: Actualización de Perfil

Descripción	Muestra como actualizar la información de una cuenta ya creada
Identificador	P.2
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Sistema Prepago, Radius
Pre-condiciones	Tener un perfil existente
Acciones	Introducir datos Validar información de cuenta Actualizar perfil y almacenar en BD Aviso de éxito o rechazo
Post-condiciones	Perfil actualizado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.43 Caso para Activación de Perfiles.

Depuración de perfiles por inactividad

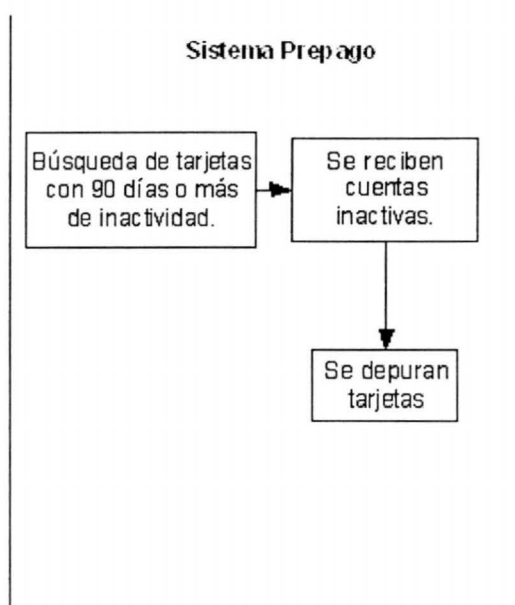


Figura 4.20 Escenario para Depuración de Perfiles

Caso 3: Depuración de Perfil

Descripción	Muestra la depuración de perfiles por ser falta de uso.
Identificador	P.3
Autor	Gerardo Canepa
Fecha	18/Sep/2003
Revisión	01/Oct/2003
Actores	Cliente, Sistema Prepago, Radius
Pre-condiciones	Tener un perfil existente
Acciones	Buscar cuentas que cumplan con la inactividad Depurar cuentas
Post-condiciones	Perfil Depurado
Incluye	Ninguno
Extensión	Ninguno
Generalización	Ninguno

Tabla 4.44 Caso para Depuración de Perfiles.

4.4 Conclusiones del Capítulo

En este capítulo se cubrió la fase de análisis del proyecto. Se mostró la visión, que es la misma definida en el capítulo 2 y versiones anteriores del proyecto que a través de juntas dieron como resultado la arquitectura de sistema final. (Ver figura 4.4) Se utilizaron escenarios para mostrar lo que se espera de la tarjeta prepagada Internet y los riesgos involucrado en el proyecto.

El documento de alcance y estrategia sirvió para poder definir claramente el proyecto y reutilizar procesos ya existentes para la tarjeta.

5 DISEÑO

5.1 Bases de Datos

A continuación se muestran y explican las bases de datos desarrollados para el sistema de prepago Internet. Se comienza con el Modelo Entidad Relación. El diagrama incluye todas las entidades involucradas en el proyecto. Se muestra el diseño físico de la base de datos de tarjetas, *log* de transacciones, tarifas y base de datos administrativa de tarjetas. Finalmente se muestra el diseño de las pantallas para servicio a cliente y monitoreo de saldos de cliente.

5.1.1 Modelo Entidad Relación

El siguiente diagrama muestra el modelo entidad relación que utiliza el sistema de prepago Internet:

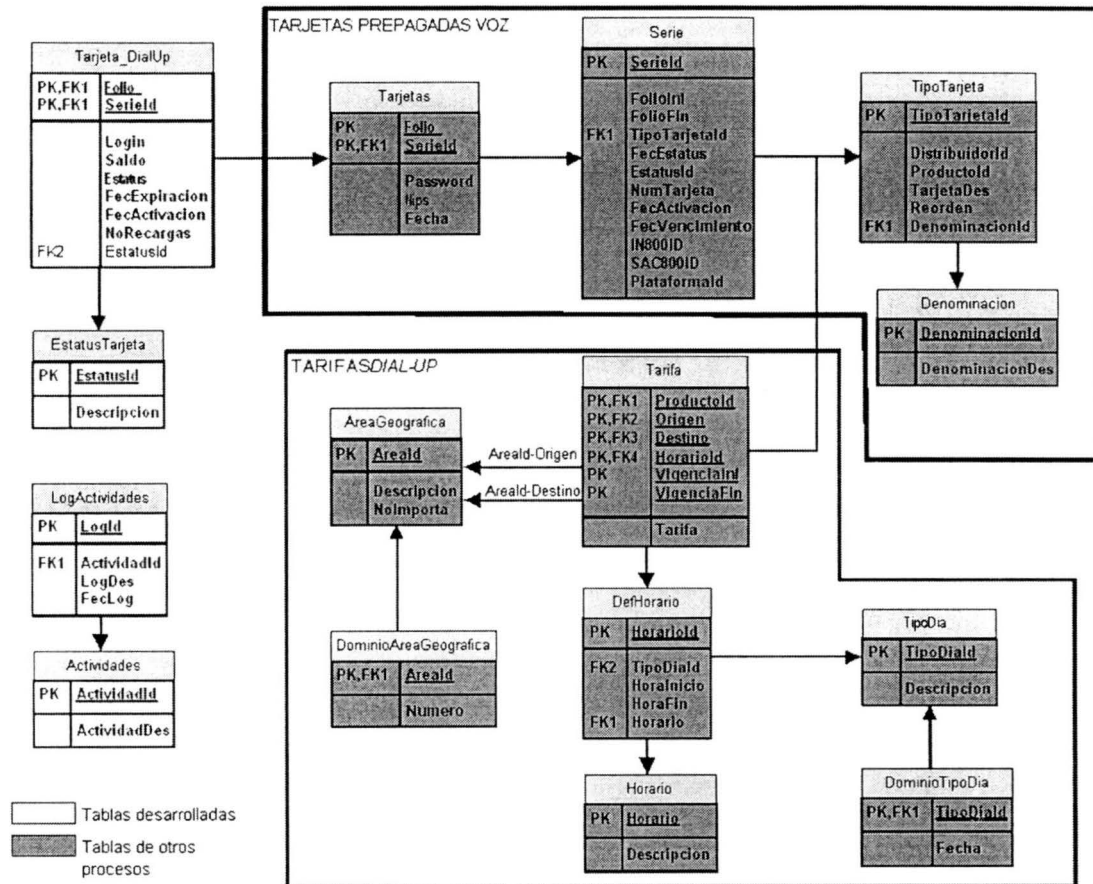


Figura 5.1 Modelo Entidad-Relación Prepago Internet

Las siguientes tablas se muestran los atributos de las entidades, que fueron definidas e ilustradas en la figura 5.1:

Tarjeta DialUp

<u>Folio</u>	Identificador que nos dice a que folio pertenece la tarjeta
<u>Seriield</u>	Identificador que nos dice a que serie pertenece la tarjeta
Login	Cuenta de la tarjeta para conectarse a TNT o recargas vía Web
Saldo	Cantidad que representa el dinero abonado en la tarjeta
Estatus	Identificador que nos dice en que estatus se encuentra la tarjeta
FecExpiración	Fecha en la cual la tarjeta dejará de ser válida
FecActivación	Fecha en la cual el usuario utilizó por primera vez la tarjeta
NoRecargas	No de recargas que se le ha hecho a la tarjeta

Tabla 5.1 Descripción de atributos de tabla Tarjeta_DialUp

Estatus

<u>EstatusID</u>	Identificador del estatus
Descripción	Describe el estatus en el cual se encuentra la tarjeta

Tabla 5.2 Descripción de atributos de tabla Estatus

Tarjetas

<u>Folio</u>	Identificador del folio
<u>Seriield</u>	Identificador que nos dice a que serie pertenece la tarjeta
Password	Clave que sirve para validar al usuario al tratar de conectarse o hacer uso de los servicios Web
Nips	Clave para validar tarjetas de Voz
Fecha	Fecha en que se activó internamente la tarjeta

Tabla 5.3 Descripción de atributos de tabla Tarjetas

Serie

<u>SeriId</u>	Identificador de la serie
FolioIni	Folio en el cual inicia la serie
FolioFin	Folio en el cual termina la serie
TipoTarjetaId	Identificador que nos dice a que tipo de tarjeta pertenece la serie
NumTarjeta	Número de tarjeta utilizado por personal administrativo Marcatel para tarjetas de voz.
FecActivacion	Fecha que indica cuando fue activada la serie
FecVencimiento	Fecha que nos dice cual es el último día para que un cliente active su serie
IN800ID	Número identificador 800 para tarjetas de voz
SAC800ID	Numero 800 de Servicio de Atención a Clientes para tarjetas de voz
EstatusId	Identificador que nos dice en que estatus se encuentra la serie
PlataformaId	Plataforma telefónica que utiliza la tarjeta de voz.

Tabla 5.4 Descripción de atributos de tabla Serie

TipoTarjeta

<u>TipoTarjetaId</u>	Identificador del tipo de Tarjeta
DistribuidorId	Identificador que nos dice a que distribuidor pertenece el tipo de tarjeta
ProductId	Identificador que nos dice a que tipo de producto Interno pertenece el tipo de tarjeta
TarjetaDes	Descripción de el tipo de tarjeta
Reorden	Utilizado para definir las series de tarjeta de voz.
DenominacionId	Identificador que nos dice a que denominación pertenece el tipo de tarjeta

Tabla 5.5 Descripción de atributos de tabla TipoTarjeta

Denominación

<u>DenominacionId</u>	Identificador de la denominación
DenominacionDes	Descripción de la denominación

Tabla 5.6 Descripción de atributos de tabla Denominación

Tarifa

<u>Productoid</u>	Identificador que nos dice a que producto pertenece la tarifa
Origen	Identificador que nos dice a que origen de llamada pertenece la tarifa
Destino	Identificador que nos dice a que destino de llamada pertenece la tarifa
Horarioid	Identificador que nos dice a que horario pertenece la tarifa
Vigencialni	Nos dice la fecha en que empieza a ser vigente la tarifa
VigenciaFin	Nos dice la fecha en la cual expira la tarifa
Tarifa	Cantidad que indica el costo por minuto del uso de DialUp prepago

Tabla 5.7 Descripción de atributos de tabla Tarifa

AreaGeografica

<u>Areald</u>	Identificador del área de llamada
Descripción	Descripción del área de llamada
NoImporta	Campo para definir si uso de tarifa es relevante.

Tabla 5.8 Descripción de atributos de tabla AreaGeografica

DominioAreaGeografica

<u>Areald</u>	Identificador que nos dice a que área pertenece el dominio
Numero	Numero telefónico con el cual se identifica el área de la llamada

Tabla 5.9 Descripción de atributos de tabla Dominio AreaGeografica

DefHorario

<u>Horariold</u>	Identificador de la definición del horario
TipoDiald	Identificador que nos dice a que tipo de día pertenece la definición del horario
Horainicio	Hora inicial en la cual se empieza a tarificar en esta definición del horario
HoraFin	Hora final en la cual se termina a tarificar en esta definición del horario
Horario	Identificador que nos dice a que horario pertenece la definición del horario

Tabla 5.10 Descripción de atributos de tabla DefHorario

Horario

<u>Horario</u>	Identificador del horario
Descripción	Descripción del horario

Tabla 5.11 Descripción de atributos de tabla Horario

TipoDia

<u>TipoDia</u>	Identificador del tipo de día. Estos días se dividen en festivos o especiales para Marcatel.
Descripción	Descripción del tipo de día

Tabla 5.12 Descripción de atributos de tabla TipoDia

DominioTipoDia

<u>TipoDia</u>	Identificador que nos dice a que tipo de día pertenece el dominio
Fecha	Día al que pertenece el dominio

Tabla 5.13 Descripción de atributos de tabla DominiotipoDia

LogActividades

<u>Login</u>	Identificador del Log de actividades
ActividadId	Identificador que nos dice a que actividad pertenece el Log de actividades
LogDes	Descripción del Log de actividades
FecLog	Fecha en que se realizó el Log de actividades

Tabla 5.14 Descripción de atributos de tabla LogActividades

Actividades

<u>ActividadId</u>	Identificador de la actividad
ActividadDes	Descripción de la actividad

Tabla 5.15 Descripción de atributos de tabla Actividades

5.1.2 Construcción de Tablas de Tarjetas

A continuación se muestra la construcción de tablas hecha para tarjetas, se muestra el nombre del atributo y el tipo de dato utilizado. La construcción de tablas está documentada con las tablas que fueron mostradas en el modelo E-R en la figura 5.1. En la construcción del sistema los nombres de las tablas no fueron modificados debido a su relación con otros procesos críticos y productos de Marcatel:

Tabla TipoTarjeta

Nombre	Tipo de Dato
*TipoTarjetaID	Tinyint
DistribuidorID	Tinyint
DenominacionID	Tinyint
PlanID	Int
TarjetaDes	Varchar(50)
Reorden	Smallint

Tabla 5.16 Tabla tarjeta

Tabla Serie

Nombre	Tipo de Dato
*SerieID	Integer
FoliolIni	Integer
Foliofin	Integer
NumTarjeta	Integer
FecActivacion	DateTime
FecVenimiento	DateTime
TarjetaID	Tinyint
PlanID	Integer
IN800ID	Tinyint
SAC800ID	Tinyint
PlataformaID	Tinyint

Tabla 5.17 Tabla Serie

Tabla Tarjetas

Nombre	Tipo de Dato
*SerieID	Integer
*Folio	Integer
TarjetaNum	Varchar(15)
Nips	Varchar(10)
Fecha	DateTime

Tabla 5.18 Tabla NIPS

Tabla Tarjeta_DialUp

<u>Nombre</u>	<u>Tipo de Dato</u>
*SerialID	Integer
*Folio	Integer
Login	Varchar(10)
Saldo	Money
Estatus	Smallint
Fecexpiracion	DateTime
Fecactivacion	DateTime
No_recargas	Integer
Productold	Integer

Tabla 5.19 Tabla NIPS_DialUp

Tabla Denominacion

<u>Nombre</u>	<u>Tipo de Dato</u>
*DenominacionId	TinyInt
DenominacionDes	Money

Tabla 5.20 Tabla Denominacion

5.1.3 Construcción de Tablas para Log de Transacciones

A continuación se muestra la construcción de tablas hecha para registrar transacciones bancarias y actividades del usuario:

Tabla Actividades

<u>Nombre</u>	<u>Tipo de Dato</u>
*ActividadID	SmallInt
ActividadDes	Varchar(50)

Tabla 5.21 Tabla Denominacion

Actividad TC

<u>Nombre</u>	<u>Tipo de Dato</u>
TransaccionID	Bigint
LogID	Bigint
Login	Varchar(10)
Cantidad	Money
Ini_fectransaccion	DateTime
Fin_fectransaccion	DateTime
TipoTarjeta	Smallint
Banco	SmallInt
Referencia	bigint
Respuesta	Varchar(100)
Nombre	Varchar(50)
TarjetaCredito	Varchar(8)

Tabla 5.22 Tabla ActividadTC (actividades con tarjeta de crédito)

LogActividades

<u>Nombre</u>	<u>Tipo de Dato</u>
LogID	Bigint
ActividadID	Smallint
FolioOrigen	Varchar(10)
FolioDestino	Varchar(50)
Monto	Money
LogDes	Varchar(100)
FecLog	Datetime

Tabla 5.23 Tabla LogActividades

5.1.4 Integración de Tablas de Tarifas

A continuación se muestra lo utilizado para determinar las tarifas necesarias por minuto:

AreaGeográfica

<u>Nombre</u>	<u>Tipo de Dato</u>
AreaID	SmallInt
Descripcion	Varchar(50)
NoImporta	Bit

Tabla 5.24 Tabla AreaGeografica

DefHorario

<u>Nombre</u>	<u>Tipo de Dato</u>
Id	Smallint
TipoDialD	Smallint
Horainicio	DateTime
HoraFin	DateTime
Horario	TinyInt

Tabla 5.25 Tabla DefHorario

DominioAreaGeografica

<u>Nombre</u>	<u>Tipo de Dato</u>
Areald	Smallint
Numero	Varchar(20)

Tabla 5.26 Tabla DominioAreaGeografica

DominioTipoDia

<u>Nombre</u>	<u>Tipo de Dato</u>
TipoDialD	Smallint
Fecha	DateTime

Tabla 5.27 Tabla DominioTipoDia

Horario

<u>Nombre</u>	<u>Tipo de Dato</u>
Horario	Smallint
Descripcion	DateTime

Tabla 5.28 Tabla Horario

Tarifa

<u>Nombre</u>	<u>Tipo de Dato</u>
ProductID	Integer
Origen	SmallInt
Destino	SmallInt
Id	SmallInt
Vigencialni	DateTime
VigeniaFin	DateTime
Tarifa	Money

Tabla 5.29 Tabla Tarifa

TipoDialD

<u>Nombre</u>	<u>Tipo de Dato</u>
TipoDialD	Smallint
Descripcion	Varchar(30)

Tabla 5.30 Tabla TipoDialD

5.1.4 Integración de Tablas Radius y Conexión de Usuarios

A continuación se muestra lo utilizado para determinar que usuarios se encuentran en línea y cuales fueron desconectados:

RadiusAccounting

<u>Nombre</u>	<u>Tipo de Dato</u>
SessionID	Bigint
LoggedAt	DateTime
Group_Name	Varchar(64)
User_Name	Varchar(64)
Acct_Status_Type	Integer
Calling_station_ID	Varchar(32)

Tabla 5.31 Tabla RaiusAccounting

UsersConnected

<u>Nombre</u>	<u>Tipo de Dato</u>
*Id_user	Integer
User_Name	Varchar(64)
SessionID	Varchar(64)
Calling_Station	Varchar(32)
Called_Station	Varchar(32)
LoggedAt	DateTime
Tarifa	Money

Tabla 5.32 Tabla LogActividades

5.2 Diseño de Pantallas para Servicio a Cliente

El diseño de las pantallas para el servicio a clientes se basó en un formato prediseñado utilizado para la página principal de Marcatel. A continuación se muestra un ejemplo del diseño utilizado:

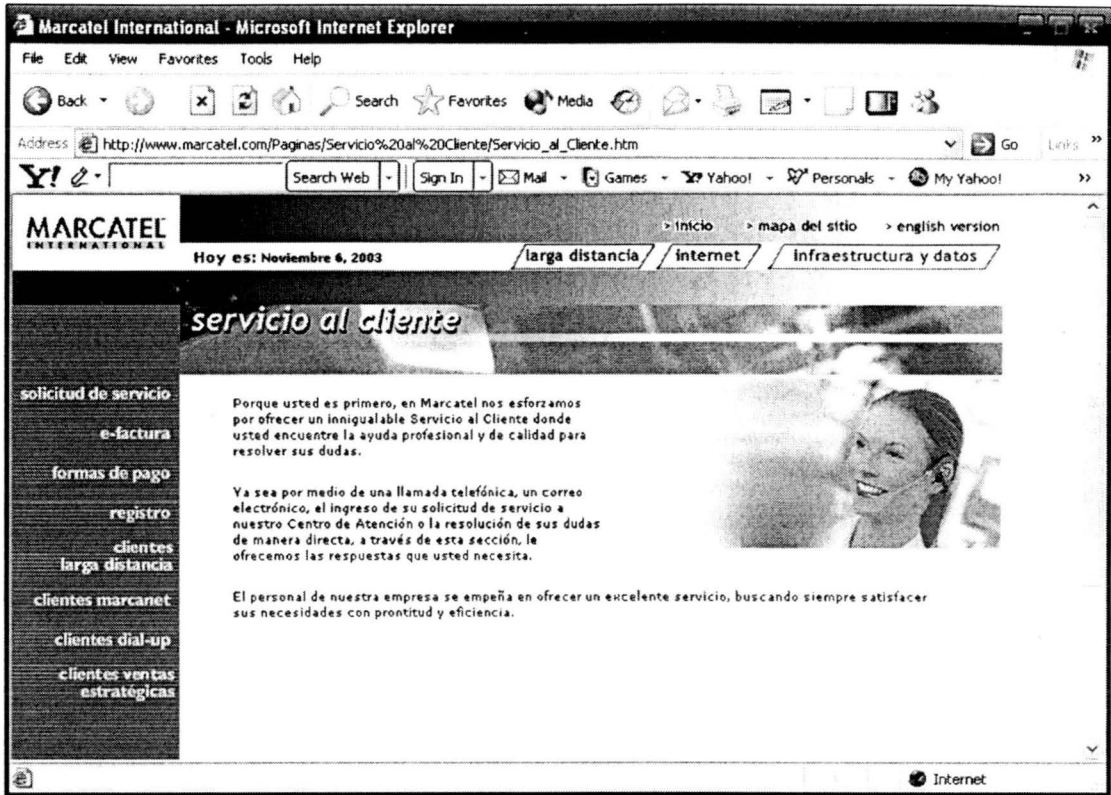


Figura 5.2 Diseño utilizado en páginas principales de Internet para servicio a cliente

Se modificaron las opciones que se encuentran la parte izquierda de la figura 5.5 para el uso de opciones de prepago internet. El logotipo de servicio a cliente y fotografía se conservaron y se usó el mismo tipo de letra. Se utilizó el font Tahoma color negro con un fondo blanco para mostrar el contenido. El menú en la parte izquierda utiliza imágenes con texto blanco con fondo de rayas gris oscuras y combinación de negro. Las opciones de menú tienen un texto alterno color amarillo al sobrepasar el puntero del mouse en cada una. Para las ligas de texto se utiliza negro sin subrayar. El resultado final al usar este diseño es el siguiente:

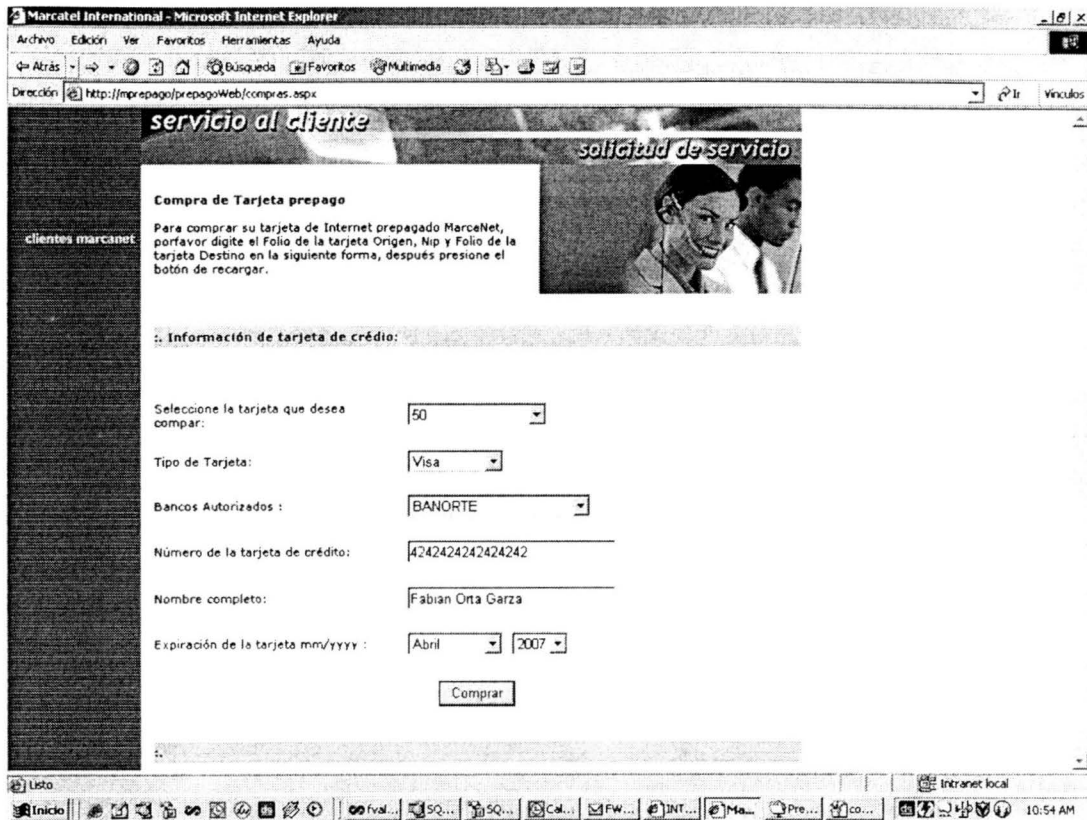


Figura 5.3 Resultado final de aplicación de diseño para compra de tarjetas.

5.3 Diseño de Pantalla para Monitoreo de Saldos

A continuación se muestra el diseño final de monitoreo de saldos:

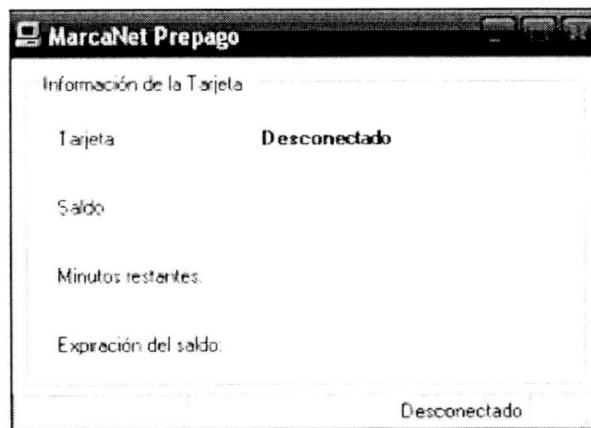


Figura 5.4 Pantalla de monitoreo de saldos

Se utilizó un frame con título “Información de Tarjeta” que contiene lo siguiente:

- Tarjeta: La tarjeta que se está utilizando en ese momento.
- Saldo: El saldo restante de la tarjeta.
- Minutos restantes: Los minutos que faltan según la tarifa utilizada.
- Expiración de saldo: Fecha en la que expira el saldo de la tarjeta.

La pantalla al ser minimizada, se esconde y se muestra un icono de actividad en la bandeja de sistema, esto se muestra en la siguiente figura:

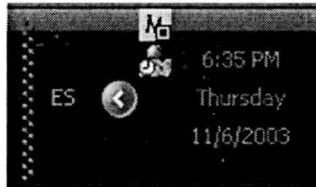


Figura 5.5 Icono de monitoreo de saldo

Este icono muestra dos estados, conectado y desconectado. En la siguiente tabla podemos ver una explicación de cada estado:



Icono de estado	Descripción
	Monitoreo de saldos. Esta conectado con el servidor.
	Monitoreo de saldos. Se encuentra desconectado del servidor.

Tabla 5.33 Icono de estados para monitoreo de saldos

5.4 Diseño de pantallas de reportes

A continuación se muestra la pantalla que se utilizó como estándar para todos los reportes establecidos:

= Reporte de Usuarios Conectados =		
Cuenta	Hora	Tarifa
0328000001	06/06/2003 01:11:16 p.m.	0.7
0328000003	06/06/2003 05:37:19 p.m.	1.15
0328000012	06/06/2003 08:19:47 p.m.	0.5

Marcatel International 2003

Figura 5.6 Diseño de pantalla para usuarios conectados

En las esquinas superiores se colocaron los logotipos de Marcatel y Universidad de Monterrey respectivamente. La pantalla que muestra los reportes consiste en dos partes. La primera representa los títulos superior e inferior. El tipo de letra utilizado es *verdana* tamaño dos. El color utilizado es un verde oscuro, su valor en hexadecimal es B0BB9D. La segunda parte representa la información a mostrar que se encuentra en la parte central. El tipo de letra utilizado es *arial* tamaño cuatro. El fondo es un verde claro, su valor en hexadecimal es FFFFF0. A continuación se muestran los demás diseños para los reportes establecidos:

Reporte de Actividad Diaria con Tarjeta de Crédito
Del 20 Octubre del 2003

Actividad	Número de Tarjeta	Hora	Cantidad
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:20:43 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:23:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	45692301	29/08/2003 11:23:59 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:24:31 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:25:22 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	34560345	29/08/2003 11:26:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:40:01 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	23450000	03/09/2003 11:58:34 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	17/09/2003 10:41:44 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	39485234	17/09/2003 10:41:58 a.m.	0
Approved	65349078	17/09/2003 10:48:02 a.m.	160.9000
Approved	32094345	17/09/2003 10:55:52 a.m.	100.0000
Approved	65349023	17/09/2003 11:02:21 a.m.	250.0000
Approved	65201392	17/09/2003 11:12:42 a.m.	50.0000
Approved	23455079	17/09/2003 11:19:06 a.m.	124.9800
Approved	23455079	17/09/2003 11:22:28 a.m.	400.000
Approved	65961179	17/09/2003 11:27:13 a.m.	340.0000
Approved	45693000	17/09/2003 11:36:21 a.m.	250.0000

Figura 5.7 Diseño de pantalla para reportes actividad con tarjeta de crédito

Reporte de Actividad Diaria
Del 20 Octubre del 2003

Cuenta	Hora	Tarifa
acero tek	06/11/2003 04:07:00 p.m.	0.65
acero tek	06/11/2003 04:07:00 p.m.	0.65
acero tek	06/11/2003 04:14:13 p.m.	0.65
acero tek	06/11/2003 04:14:13 p.m.	0.65
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
acero tek	06/11/2003 04:14:34 p.m.	0.65
acero tek	06/11/2003 04:14:34 p.m.	0.65
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
acero tek	06/11/2003 04:15:03 p.m.	0.65
acero tek	06/11/2003 04:15:03 p.m.	0.65

Figura 5.8 Diseño de pantalla para reporte de cobro de tarifas diaria.

Esta pantalla utiliza los mismos colores establecidos y mostrados en la figura 5.6, la diferencia del diseño entre cada uno de los reportes establecidos se encuentra en el título superior. Este indica la información a mostrar en el reporte.

5.5 Conclusiones del capitulo

Este capítulo cubrió el desarrollo de las pantallas que proveen el servicio a los clientes. Este diseño es una pequeña parte de la tercera fase del desarrollo *MSF* que continuó con la construcción de módulos presentados.

6 CONSTRUCCIÓN

El siguiente capitulo muestra la construcción del sistema de prepago de tarjetas de Internet. Se comienza explicando los estándares de programación utilizados. A continuación se da una breve explicación de cómo se construyeron los módulos para el cliente, sistema y administrativos. Después, se explican la gestión de incidencias hechas a cada módulo y sus resultados.

6.1 Estándares de Programación

La siguiente tabla muestra los estándares de programación utilizados durante la construcción:

Estándar	Descripción	Ejemplo
Clase	<ul style="list-style-type: none"> La letra inicial del nombre debe de ser mayúscula Si el nombre abarca más de 1 palabra, todas las demás palabras deben iniciar con mayúscula 	EjemploDeClase
Método	<ul style="list-style-type: none"> La letra inicial del nombre debe de ser minúscula Si el nombre abarca más de 1 palabra, todas las demás palabras deben iniciar con mayúscula El nombre debe reflejar el funcionamiento del método 	actualizarEjemplo
Función	<ul style="list-style-type: none"> La letra inicial del nombre debe de ser minúscula Si el nombre abarca más de una palabra, todas las demás palabras deben iniciar con mayúscula El nombre debe reflejar la acción de la función 	esEjemploValido
Variables	<ul style="list-style-type: none"> La letra inicial del nombre debe de ser minúscula Si el nombre abarca más de 1 palabra, todas las demás palabras deben iniciar con mayúscula El nombre debe reflejar el uso que se le va a dar a la variable 	Ejemplo de una variable utilizada para un contador : contadorEjemplo
Objeto	<ul style="list-style-type: none"> El nombre debe de iniciar con la palabra "Obj" indicando que es un objeto externo, seguido del nombre del objeto Si el nombre abarca más de 1 palabra, todas las demás palabras deben iniciar con mayúscula El nombre debe reflejar el uso que se le va a dar a la variable 	ObjEjemplo

Estándar	Descripción	Ejemplo
Stored Procedures	<ul style="list-style-type: none"> • El nombre del Stored Procedure debe iniciar con la palabra "SP". Seguido del carácter "_" para separar el tipo de stored procedure • El tipo de store procedure se identifica utilizando una palabra que defina el tipo de acción que realiza: <ul style="list-style-type: none"> Sel - Seleccionar Del – Borrar Ins – Insertar Upd – Actualizar • Siguen una o más palabras que identifiquen que información de la base de datos se va a utilizar, cada una de estas palabras deben de iniciar con mayúscula 	SP_SelEjemploDatos
String de conexión	<ul style="list-style-type: none"> • El nombre debe de iniciar con la palabra "conStr" • Debe seguir el nombre de la base de datos o tabla a utilizar, cada palabra debe de iniciar con mayúscula 	conStrEjemploRadius
Comando de SQL	<ul style="list-style-type: none"> • El nombre debe de iniciar con la palabra "cmd" • Debe seguir el nombre de la base de datos o tabla a utilizar, cada palabra debe de iniciar con mayúscula 	cmdEjemploRadius

Tabla 6.1 Estándares de programación

6.2 Construcción del Sistema

A continuación se muestra el desarrollo de los módulos para tarjetas prepagadas de Internet

6.2.1 Módulos del Cliente

Para los siguientes módulos, se creó una librería en VB.NET para desarrollar un objeto llamado tarjeta que maneja todas las tareas que debe hacer la tarjeta prepagada. Después se creó otra solución en Visual Studio .NET con ASP.NET para construir los servicios para el cliente.

A continuación se presenta una explicación de cómo se construyo cada servicio para cumplir los objetivos particulares del cliente. Los servicios a cubrir son los siguientes:

- Objeto Tarjeta
- Recarga de saldo con tarjeta de crédito
- Compra de Tarjeta
- Traspaso de saldo de tarjeta a tarjeta
- Monitor de saldos para cliente.

6.2.1.1 Objeto Tarjeta

El objeto tarjeta representa una clase con los métodos principales que permiten el funcionamiento adecuado de los servicios cliente. Lo métodos públicos utilizados son los siguientes:

Método	Parámetros	Descripción
recargarTarjeta	folioOrigen tipo String folioDestino tipo String	Método de traspaso de saldo de una tarjeta origen a una tarjeta destino.
recargarTarjeta	nombre tipo String folio tipo String fecExp tipo String tipoTarejta tipo Int16 Monto tipo Decimal Banco tipo int16 TarjetaCredito tipo String respuesta tipo String	Método de recarga de tarjeta prepagada por medio de una tarjeta de crédito.
validarTarjeta	folio tipo String	Método para validar si la tarjeta existe en la base de datos.
validarTarjeta	Folio tipo swing Nip tipo string	Método para validar si la contraseña de una tarjeta es válida
consultarTarjeta	folio tipo String saldo tipo Decimal fecExpiracion tipo DateTime	Método para consultar tarjetas prepago que devuelve el saldo y la fecha de expiración por medio de las variables saldo y fecExpiracion respectivamente
obtenerSaldo	folio tipo String	Método para obtener el saldo de una tarjeta.
comprarTarjeta	nombre tipo String fecExp tipo String tipoTarjeta tipo String tipoMonto tipo String banco tipo Int16 tarjetaCredito tipo String respuesta tipo String denominación tipo Decimal login tipo String pass tipo String	Método utilizado para comprar una tarejta por Internet. Este método llamará al objeto bancario y manda procesar la transacción.

Tabla 6.2 Métodos utilizados por el objeto tarjeta

6.2.2 Módulos Sistema y Cliente

A continuación se muestran la construcción hecha para los módulos del sistema.

6.2.1.2 Servicio para manejo de conexiones y tarifas

El proceso de tarificación está compuesto de dos métodos principalmente, la inicialización de usuarios a tarificar y la tarificación de usuarios.

Inicialización de usuarios a tarificar

La inicialización se ejecuta con el método **inicializarLista**, el cual recibe como parámetro **fechaProceso** que sirve para identificar la hora en que inicio la ejecución del método.

La función principal del método es obtener los usuarios del Servidor TNT que se encuentran conectados en ese preciso momento por medio de un objeto llamado **Plataforma_SNMPTNT**. Después obtener los usuarios, éstos son depurados mediante un método llamado **depuraListaSesiones**, el cual separa los usuarios son de Internet prepago de los usuarios de *Dial-Up* normal.

Después de haber sido depurada la lista de usuarios, los usuarios de Internet prepago son insertados en la Lista de usuarios a tarificar (la cual contiene los usuarios a utilizar por el método de Tarificación posteriormente), como lo muestra el siguiente código:

```
Dim connStr As SqlConnection
    connStr = conectarBD()
    connStr.Open()
'La lista de usuarios a tarificar es limpiada totalmente para su inicialización
    Dim cmdRadius As SqlCommand = New SqlCommand("Sp_DelUsuariosIniciales",
connStr)
    cmdRadius.CommandType = CommandType.StoredProcedure
    cmdRadius.ExecuteNonQuery()
    cmdRadius.CommandText = "Sp_InsUsuariosIniciales"

'Sesiones contiene la lista de usuarios de Internet prepago obtenida por la
Plataforma_SNMPTNT
    For tamArreglo = 0 To sesiones.GetLength(1) - 1
        If sesiones(1, tamArreglo) <> "" Then
            cmdRadius.Parameters.Add(New SqlParameter("@sessionID",
SqlDbType.VarChar)).Value = sesiones(0, tamArreglo)
            cmdRadius.Parameters.Add(New SqlParameter("@userName",
SqlDbType.VarChar)).Value = sesiones(1, tamArreglo)
```

```

        cmdRadius.Parameters.Add(New
SqlCommandType.VarChar).Value = sesiones(2, tamArreglo)
        cmdRadius.Parameters.Add(New
SqlCommandType.VarChar).Value = sesiones(3, tamArreglo)
        cmdRadius.ExecuteNonQuery()
        cmdRadius.Parameters.Clear()
    End If
Next
connStr.Close()

```

Tarificación de usuarios

La tarificación se realiza por medio del método **tarificaUsuarios** el cual se divide en 6 pasos principales, los cuales son ejecutados por los siguientes métodos:

- a) insertarUsuariosConectados
- b) actualizarTarifas
- c) tarificar
- d) cobrarSalDOS
- e) removerUsuariosdesconectados
- f) validarSalDOS

Método insertarUsuariosConectados

Obtiene los usuarios de Internet prepago conectados en el último minuto a tarificar, insertándolos en la lista de usuarios a tarificar (que es utilizada por el método de cobrarSalDOS para descontar el minuto consumido) como lo muestra el siguiente código:

```

'El store procedure Inserta la información de los usuarios a la lista de usuarios a tarificar
Dim cmdPrepago As SqlCommand = New
SqlCommand("Sp_InsUsuariosConectadosRadius", connStr)
cmdPrepago.CommandType = CommandType.StoredProcedure
'sqlReader contiene los registros obtenidos de la Base de Datos de Radius
'al hacer la selección de los usuarios conectados en el último minuto
While sqlReader.Read
    cmdPrepago.Parameters.Add(New
SqlCommandType.VarChar).Value = sqlReader.GetValue(0)
    cmdPrepago.Parameters.Add(New
SqlCommandType.VarChar).Value = sqlReader.GetValue(1)
    cmdPrepago.Parameters.Add(New
SqlCommandType.VarChar).Value = sqlReader.GetValue(3)
    cmdPrepago.Parameters.Add(New
SqlCommandType.VarChar).Value = sqlReader.GetValue(4)
    cmdPrepago.Parameters.Add(New
SqlCommandType.VarChar).Value = sqlReader.GetValue(2)
    cmdPrepago.ExecuteNonQuery()
    cmdPrepago.Parameters.Clear()
End While

```

Método actualizarTarifas

Obtiene la tarifa correspondiente a cada uno de los usuarios de la lista de usuarios a tarificar, actualizando su campo de tarifa como lo muestra el siguiente Store procedure:

```

CREATE Procedure Sp_UpdTarifas
    @areaDestino varchar(10),
    @login varchar(10),
    @sessionID varchar(64)
AS
DECLARE
    @productID int,
    @tarifa money,
    @horaTarifa varchar(50),
    @minTarifa varchar(50),
    @horafinal datetime
/* Obtiene el tipo de producto de la tarjeta*/
set @ProductID=(select T.PlanID from nips_DialUp ND,Nips N,Serie S,Tarjeta T
    where ND.login=@login and ND.Folio=N.Folio and ND.SerieID=n.SerieID
    AND N.SerieID=S.SerieID AND S.TarjetaID=T.TarjetaID)
set @horaTarifa=convert(varchar,datepart(hh,getdate()))
set @minTarifa=convert(varchar,datepart(mi,getdate()))
set @horafinal= convert(datetime,'1/1/1900')
set @horafinal=dateadd(hh,convert(smallint,@horaTarifa),@horafinal)
set @horafinal=dateadd(mi,convert(smallint,@minTarifa),@horafinal)
set @tarifa=(Select ISNULL(min(TF.Tarifa),0.5) from DominioTipoDia D,TipoDia
T,DefHorario H,Tarifa TF,AreaGeografica AG,DominioAreaGeografica DAG where
datepart(day,D.fecha)=datepart(day,getdate()) and
datepart(mm,D.fecha)=datepart(mm,getdate()) and
datepart(yy,D.fecha)=datepart(yy,getdate())
    and T.TipoDialD=D.TipoDialD and H.TipoDialD=T.TipoDialD and H.id=TF.Id
and H.Horainicio<@horafinal and H.Horafin>@horafinal AND ProductID=@ProductID and
TF.Destino=AG.Areald and AG.Areald=DAG.Areald and DAG.numero=@areaDestino
    )

BEGIN TRAN myTran

    UPDATE UsersConnected
    SET Tarifa=@tarifa
    WHERE User_name=@login and sessionID=@sessionID
    IF @@ERROR<>0
    BEGIN
        RAISERROR 2000100 'Error al tratar de actualizar tarifa'
        ROLLBACK TRAN myTran
        RETURN
    END
COMMIT TRAN

GO

```

Método tarifcar

La función principal de tarifcar es Insertar la información de los usuarios a tarifcar a un Log de transacciones diarias, el cual sirve para llevar un registro de los consumos realizados.

Método cobrarSaldos

Éste método descuenta la tarifa por minuto del saldo de cada tarjeta de Internet prepagado que se encuentra en la lista de usuarios a tarifcar, tal y como lo muestra el siguiente store procedure:

```
CREATE Procedure Sp_UpdCobroTarifa
    @login varchar(10),
    @sessionID varchar(64)
AS
DECLARE
    @tarifa money,
    @saldo money
    set @tarifa=(Select max(tarifa) from usersconnected where sessionid=@sessionid
and user_name=@login
    )
    set @saldo=(Select saldo-convert(money,@tarifa) from Nips_dialUp WHERE
login=@login)

    IF @saldo<=0
    Set @saldo=0
    BEGIN TRAN myTran

        UPDATE Nips_DialUp
        SET saldo=@saldo
        WHERE login=@login
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al tratar de cobrar tarifa'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
GO
```

Método removerUsuariosdesconectados

Obtiene los usuarios que se desconectaron del servidor Radius en el último minuto y los remueve de la lista de usuarios a tarifcar para no ser tomados en cuenta en la siguiente tarifcación.

Método validaSalDOS

Todos los usuarios en la lista de usuarios a tarificar son validados de manera que los usuarios conectados que no cuenten con suficiente saldo para el consumo de otro minuto, sean removidos de la lista de usuarios a tarificar y desconectados del servidor TNT por medio del objeto *plataforma_SNMPTNT*, como lo muestra el siguiente código:

```

connStrVerif = conectarBD()
connStrVerif.Open()
Dim cmdVerif As SqlCommand = New SqlCommand("Sp_verificaSalDOS",
connStrVerif)
'sqlreader contiene los registros de la lista de usuarios a tarificar
Do While sqlReader.Read()
    cmdVerif.Parameters.Clear()
    cmdVerif.CommandType = CommandType.StoredProcedure
    cmdVerif.CommandText = "Sp_verificaSalDOS"
    cmdVerif.Parameters.Add(New SqlParameter("@login",
SqlDbType.VarChar)).Value = Trim(sqlReader.GetString(0))
    saldoValido = cmdVerif.ExecuteScalar
    If Not saldoValido Then
        'Desconexión Telnet del Usuario ----- aqui -----
        obj_SNMP.desconectar_Usuario("", Trim(sqlReader.GetString(1)))
        'Desconexión por medio de Telnet del Usuario ----- aqui -----
        cmdVerif.CommandText = "Sp_DelUsuarioSaldo"
        cmdVerif.Parameters.Add(New SqlParameter("@SessionId",
SqlDbType.VarChar)).Value = Trim(sqlReader.GetValue(1))
        cmdVerif.ExecuteNonQuery()
        control.AppendText(CStr(numActividad) & "- " & "Desconexion : La cuenta " &
Trim(sqlReader.GetString(0)) & " ha sido desconectada debido a falta de saldo." &
Microsoft.VisualBasic.ControlChars.CrLf)
        numActividad = numActividad + 1
    End If
Loop
connStrUsuarios.Close()
connStrVerif.Close()

```

6.2.1.3 Componente de Monitoreo de SalDOS

El *plugin* de monitoreo de salDOS consta de dos partes principales: el cliente de monitoreo y el servidor de notificación.

El servidor de notificación

Consiste en una aplicación Windows, la cual por medio de una conexión TCP por el puerto 2003 recibe peticiones de los clientes activos, recibiendo la IP de los clientes, obteniendo de la Base de datos de Radius la cuenta de la tarjeta de Internet prepago que se encuentra relacionada con

dicha IP. Una vez obtenida la información de la tarjeta de Internet prepago, el servidor la envía al cliente en forma de paquetes de datos, los cuales son interpretados por el cliente y desplegados al usuario.

```
Private Function obtenerUsuario(ByVal Ip As String) As String
    Dim connStrRadius, connStrTarifa As SqlConnection
    Dim informacion As String
    Dim sqlReaderRadius, sqlReaderTarifa As SqlDataReader

    Try
        connStrRadius = ConectarBDRadius()
        connStrRadius.Open()
        Dim cmdRadius As SqlCommand = New SqlCommand("Sp_SelUsuarioRadius",
connStrRadius)
        cmdRadius.Parameters.Add(New SqlParameter("@IP", SqlDbType.VarChar)).Value
= Ip
        cmdRadius.CommandType = CommandType.StoredProcedure
        sqlReaderRadius = cmdRadius.ExecuteReader
        If sqlReaderRadius.Read() Then
            informacion = sqlReaderRadius.GetValue(0) & "," & sqlReaderRadius.GetValue(1)
        Else
            Return "0,0"
        End If
        sqlReaderRadius = Nothing
        'connStrRadius.Close()
        Return informacion
    Catch ex As Exception
        Return "0,0"
        Dim errorCs As New ErrorCS()
        errorCs.enviarError("Error al iniciar el servidor: " & ex.ToString())
    End Try
End Function
```

El cliente de monitoreo

El cliente consiste de una aplicación que es obtenida opcionalmente por los usuarios de tarjetas de Internet prepago, ésta aplicación busca la comunicación con el servidor de notificación cada periodo de 1 minuto, conectándose por medio del puerto 2003.

Una vez conectado el cliente, éste envía la IP al servidor de notificación para que sea procesada, esperando recibir como resultado, la información de la tarjeta relacionada con dicha IP.

```
Public Sub obtenerInfo(ByVal tarjeta As System.Windows.Forms.Label, ByVal saldo As
System.Windows.Forms.Label, ByVal minutos As System.Windows.Forms.Label, ByVal
expiracion As System.Windows.Forms.Label, ByVal mylcon As NotifyIcon, ByVal status As
StatusBar, ByRef flgRecarga As Boolean, ByRef nombreUsuario As String, ByRef sesion As
String)
    Dim flgLeerDatos As Boolean = False
```

```

Dim Timeout As Int16 = 0
Try
    status.Panels.Item(1).Text = "Conectando..."
'La ip no es mostrada por cuestión de seguridad
    tcpCliente = New ClienteTcp("0.0.0.0", puerto)
    If tcpCliente.IsConnected Then
        ntsCliente = tcpCliente.GetStream()
        myIcon.Icon = New Icon("m.ico")
        status.Panels.Item(1).Text = "Conectado"
        If ntsCliente.CanWrite Then
            If nombreUsuario = "" Or nombreUsuario = "0" Then
                Dim dirCliente As System.Net.IPAddress
                Dim ipCliente As String
                dirCliente = New
System.Net.IPAddress(System.Net.Dns.GetHostByName(System.Net.Dns.GetHostName()).
AddressList(0).Address)
                ipCliente = dirCliente.ToString
                'Envio Inicial de IP
                Dim envio As String = "INI " & ipCliente
                Dim paqueteEnvio() As Byte =
System.Text.Encoding.ASCII.GetBytes(envio.ToCharArray())
                ntsCliente.Write(paqueteEnvio, 0, paqueteEnvio.Length)
                ntsCliente.Flush()
            Else
                Dim envio As String = "MIN " & nombreUsuario & "," & sesion
                Dim paqueteEnvio() As Byte =
System.Text.Encoding.ASCII.GetBytes(envio.ToCharArray())
                ntsCliente.Write(paqueteEnvio, 0, paqueteEnvio.Length)
                ntsCliente.Flush()
            End If
        End If
        Do While (Not flgLeerDatos And ntsCliente.CanRead) And Timeout < 5
            If ntsCliente.DataAvailable Then
                Dim recepcion(128) As Byte
                Dim i As Int32 = ntsCliente.Read(recepcion, 0, 128)
                Dim ree As String = System.Text.Encoding.ASCII.GetString(recepcion)
                Dim unwanted() As Char = {" ", " ", " ", " "}
                aceptarComando(ree, ntsCliente, tarjeta, saldo, minutos, expiracion, myIcon,
flgRecarga)
                Dim datos() As String
                datos = Split(Mid(ree, 5, Len(ree) - 3), ",")
                nombreUsuario = datos(0)
                sesion = datos(1)
                flgLeerDatos = True
            End If
            Thread.Sleep(1000)
            Timeout = Timeout + 1
        Loop
        tcpCliente.Close()
        ntsCliente.Close()
    End If
Catch e As Exception
    myIcon.Icon = New Icon("mOff.ico")
    mostrarInfo(False, "", tarjeta, saldo, minutos, expiracion, myIcon, flgRecarga)
    status.Panels.Item(1).Text = "Desconectado"
    nombreUsuario = ""

```

```

    sesion = ""
    If Not (tcpcCliente Is Nothing) Then tcpcCliente.Close()
End Try
End Sub

```

6.2.1.4 Plataforma SNMP, Telnet y MaxTNT

El objeto de la plataforma TNT contiene dos métodos principales que son: la obtención de usuarios conectados a TNT mediante una conexión SNMP y la desconexión de los usuarios del TNT mediante una conexión Telnet.

Método obtener Usuarios:

Éste método se conecta al servidor TNT mediante una conexión SNMP, por medio de la cual se obtienen las sesiones de los usuarios que se encuentran conectados en ese momento.

```

Public Sub obtener_Usuarios(ByVal IP As String) Implements
Plataforma_SNMP_TNT.obtener_Usuarios
    Dim obj_SNMP As New SNMP_NET.SNMP()
    Dim key As String

    Try
        Dim tablahs As Hashtable = obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.12.3.1.4", True)
        Dim tablahsOrigen As Hashtable = obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.11.16.1.17", True)
        Dim tablahsDestino As Hashtable = obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.11.16.1.16", True)

        obtenerSesiones(tablahs, tablahsOrigen, tablahsDestino)
    Catch ex As Exception
        Dim errorMsg As New ErrorCs()
        errorMsg.SendError("Error al obtener usuarios via SNMP de la IP-" & IP & " : " &
ex.ToString)
    End Try
End Sub

```

Método desconectar Usuarios

Éste método se conecta al servidor TNT mediante una sesión de Telnet (puerto 23), recibiendo el parámetro de número de sesión para después realizar la desconexión del usuario.

*En el siguiente ejemplo se asume que existe una sesión Telnet establecida con el servidor TNT. (mediante el método conexión_TNT)

```
Public Sub desconectar_Usuario(ByVal IP As String, ByVal SessionID As String) Implements
Plataforma_SNMPTNT.desconectar_Usuario
    Dim flgLeerDatos As Boolean = False
    Dim timeout As Int16 = 0
    Try
        tcpCliente = New ClienteTelnet(IP, puertoTelnet)
        If tcpCliente.IsConnected Then
            ntsTelnet = tcpCliente.GetStream()
            'Conexión TNT, establece la sesión telnet, validando el usuario y contraseña de
            conexión 'a l servidor TNT
            conexion_TNT()
            If ntsTelnet.CanWrite Then
                Dim datosEnvio As String = "stuser -k " & SessionID
                Dim paqueteEnvio() As Byte =
System.Text.Encoding.ASCII.GetBytes(datosEnvio.ToCharArray())
                ntsTelnet.Write(paqueteEnvio, 0, paqueteEnvio.Length)
                ntsTelnet.Flush()
            End If
            Do While (Not flgLeerDatos And ntsTelnet.CanRead) And timeout < 5
                If ntsTelnet.DataAvailable Then
                    Dim datosRecepcion(128) As Byte
                    Dim i As Int32 = ntsTelnet.Read(datosRecepcion, 0, 128)
                    Dim paqueteRecepcion As String =
System.Text.Encoding.ASCII.GetString(datosRecepcion)
                    Dim datosNosolicitados() As Char = {" ", " ", " ", " "}

                    flgLeerDatos = True
                End If
                Thread.Sleep(1000)
                timeout = timeout + 1
            Loop
            tcpCliente.Close()
            ntsTelnet.Close()
        End If
        Catch ex As Exception
            If Not (tcpCliente Is Nothing) Then tcpCliente.Close()
            Dim errorMsg As New ErrorCs()
            errorMsg.SendError("Error al desconectar usuarios via TELNET de la IP-" & IP & " : "
& ex.ToString)
        End Try
    End Sub
```

6.2.1.5 Módulo Bancario

El objeto **Tarjeta** contiene todos los métodos y funciones utilizados para realizar transacciones con el banco. Este objeto es utilizado para las páginas diseñadas en ASP.NET que proveen servicios al cliente como recarga de saldo, compra de tarjeta, consulta de saldo y traspaso de saldo de tarjeta.

A continuación se nombrarán algunas de las funciones y métodos principales de este objeto, así como una descripción de su funcionalidad:

Método recargarTarjeta

Recibe como parámetros la información de la tarjeta de crédito y la tarjeta de Internet prepago, validando la primera de éstas y recargando la tarjeta de Internet prepago con la cantidad correspondiente en caso que la transacción sea aceptada por el banco.

```
Public Function recargarTarjeta(ByVal nombre As String, ByVal folio As String, ByVal fecExp
As String, ByVal tipoTarjeta As Int16, ByVal monto As Decimal, ByVal banco As Int16, ByVal
tarjetaCredito As String, ByRef respuesta As String) As Boolean
Dim consecutivo As Int32
    'Objeto bancario
    Dim APIPayWorks As New ConectorPayWorks()

    Try
        'numero consecutivo para el log de transacciones de tarjeta de crédito
        consecutivo = insertarConsecutivo(folio, monto, tipoTarjeta, banco, nombre,
tarjetaCredito)
        APIPayWorks = ConectorPayWorks.EfectuarCargo(tarjetaCredito, fecExp,
consecutivo, monto, "00000", "177.77.3.66:80")
        ' Si la transacción es valida el estatus de pago es diferente de 0
        If Not APIPayWorks.EstatusPago = 0 Then
            GuardarPago(consecutivo, APIPayWorks.CodigoAutorizacionProcesador,
APIPayWorks.MensajeRetornoProcesador, APIPayWorks.TiempoInicioTransaccion,
APIPayWorks.TiempoFinTransaccion)
            If APIPayWorks.CodigoAutorizacionProcesador > 0 Then
                Dim connStrRecarga As SqlConnection

                Try
                    connStrRecarga = ConectarBD()
                    connStrRecarga.Open()
                    'Actualiza el saldo de la tarjeta de prepago
                    Dim cmdRecarga As SqlCommand = New SqlCommand("Sp_UpdSaldo",
connStrRecarga)
                    cmdRecarga.Parameters.Add(New SqlParameter("@folio",
SqlDbType.VarChar)).Value = folio
                    cmdRecarga.Parameters.Add(New SqlParameter("@saldo",
SqlDbType.Money)).Value = monto
                    cmdRecarga.CommandType = CommandType.StoredProcedure
```

```

        cmdRecarga.ExecuteNonQuery()
        connStrRecarga.Close()
        guardarLog(3, folio, "", monto, "Recarga de tarjeta " & folio & " con tarjeta de
crédito " & tarjetaCredito & " por " & monto)
        Return True
    Catch ex As Exception
        mandarMail("Error al recargar el saldo de Tarjeta con TC : " & ex.ToString)
        Throw New Exception(ex.ToString)
        Return False
    End Try
End If
Else
    GuardarPago(consecutivo, -1, APIPayWorks.ToString, Now(), Now())
    If APIPayWorks.TextoError = "" Then
        respuesta = "Por el momento el servicio no se encuentra disponible."
    Else
        respuesta = "Tarjeta inválida/expirada, favor de verificar su tarjeta."
    End If
    Return False
End If
Catch ex As Exception
    GuardarPago(consecutivo, -1, "Error: tarjeta de crédito inválida - " & ex.ToString,
Now(), Now())
    mandarMail("Error al comprar tarjeta: " & ex.ToString)
    respuesta = "La tarjeta de crédito no se pudo procesar, favor de verificar los datos e
intentar de nuevo."
    Return False
End Try
End Function

```

Método validarTarjeta

Recibe como parámetro el folio y opcionalmente el password de la tarjeta de prepago para validar su existencia en la base de datos.

```

Public Function validarTarjeta(ByVal folio As String) As Boolean
    Dim connStrFolio As SqlConnection
    Dim flgTarjetaValida As Boolean

    Try
        connStrFolio = ConectarBD()
        connStrFolio.Open()
        Dim cmdFolio As SqlCommand = New SqlCommand("Sp_SelFolio", connStrFolio)
        cmdFolio.Parameters.Add(New SqlParameter("@folio", SqlDbType.VarChar)).Value =
folio
        cmdFolio.CommandType = CommandType.StoredProcedure
        flgTarjetaValida = cmdFolio.ExecuteScalar()
        connStrFolio.Close()
        Return flgTarjetaValida

    Catch ex As Exception
        mandarMail("Error en Validacion de Folio: " & ex.ToString)
    End Try
End Function

```

```

Return False
End Try
End Function

```

Método consultarTarjeta

Recibe como parámetro el Folio de la tarjeta de Internet prepago para después realizar una consulta a la base de datos de tarjetas prepago y obtener la información de la tarjeta.

```

Public Sub consultarTarjeta(ByVal folio As String, ByRef saldo As Decimal, ByRef
fecExpiracion As DateTime)
    Dim connStrTarjeta As SqlConnection
    Dim sqlReaderTarjeta As SqlDataReader
    Try
        connStrTarjeta = ConectarBD()
        connStrTarjeta.Open()
        Dim cmdTarjeta As SqlCommand = New SqlCommand("Sp_SelInfoTarjeta",
connStrTarjeta)
        cmdTarjeta.Parameters.Add(New SqlParameter("@folio",
SqlDbType.VarChar)).Value = folio
        cmdTarjeta.CommandType = CommandType.StoredProcedure
        sqlReaderTarjeta = cmdTarjeta.ExecuteReader()
        If sqlReaderTarjeta.Read() Then
            saldo = sqlReaderTarjeta.GetValue(0)
            fecExpiracion = sqlReaderTarjeta.GetValue(1)
        Else
            saldo = 0
            fecExpiracion = "1/1/1900"
        End If
        guardarLog(5, folio, "", 0, "Consulta de tarjeta " & folio)
        sqlReaderTarjeta = Nothing
        connStrTarjeta.Close()
    Catch ex As Exception
        mandarMail("Error al Consultar tarjeta: " & ex.ToString)
    End Try
End Sub

```

Método obtenerSaldo

Recibe como parámetro el folio de la tarjeta de Internet prepago, consultando posteriormente la base de datos de Tarjetas prepago y obteniendo el saldo correspondiente a dicho folio.

```

Public Function obtenerSaldo(ByVal folio As String) As Decimal
    Dim connStrSaldo As SqlConnection
    Dim saldoTarjeta As String

    Try
        connStrSaldo = ConectarBD()

```

```

    connStrSaldo.Open()
    Dim cmdTarjeta As SqlCommand = New SqlCommand("Sp_SelSaldo", connStrSaldo)
    cmdTarjeta.Parameters.Add(New SqlParameter("@folio",
SqlDbType.VarChar)).Value = folio
    cmdTarjeta.CommandType = CommandType.StoredProcedure
    saldoTarjeta = cmdTarjeta.ExecuteScalar()
    connStrSaldo.Close()
    Return saldoTarjeta
Catch ex As Exception
    mandarMail("Error al Consultar tarjeta - datos DB: " & ex.ToString)
    Return 0
End Try
End Function

```

Método comprarTarjeta

Recibe como parámetros la información de la tarjeta de crédito, validando estos datos con el banco, y obteniendo la información de una tarjeta de Internet prepago disponible en dado de que la transacción de compra sea aceptada por el banco.

```

Public Function comprarTarjeta(ByVal nombre As String, ByVal fecExp As String, ByVal
tipoTarjeta As Int16, ByVal tipoMonto As Int16, ByVal banco As Int16, ByVal tarjetaCredito
As String, ByRef respuesta As String, ByVal denominacion As Decimal, ByRef login As
String, ByRef pass As String) As Boolean
    Dim consecutivo As Int32
    Dim APIPayWorks As New ConectorPayWorks()
    Dim monto As Decimal

    'Buscar si existe el tipo de tarjeta y devuelve el monto si hay disponible
    If Not (revisarInventario(denominacion)) Then
        respuesta = "No hay tarejtas disponibles de $" & CStr(denominacion)
        mandarMail("Tarjetas para venta en internet con denominacion: " & denominacion & "
estan agotadas")
        Return False
    End If

    Try
        consecutivo = insertarConsecutivo("0000000000", 0.0, tipoTarjeta, banco, nombre,
tarjetaCredito)
        APIPayWorks = ConectorPayWorks.EfectuarCargo(tarjetaCredito, fecExp,
consecutivo, denominacion, "00000", "177.77.3.66:80")
        If Not APIPayWorks.EstatusPago = 0 Then
            guardarPago(consecutivo, APIPayWorks.CodigoAutorizacionProcesador,
APIPayWorks.MensajeRetornoProcesador, APIPayWorks.TiempoInicioTransaccion,
APIPayWorks.TiempoFinTransaccion)
            If APIPayWorks.CodigoAutorizacionProcesador > 0 Then
                Dim myConnection As SqlConnection

                Try
                    'Se obtiene la tarjeta disponible y se modifica el estatus

```

```

    obtenerTarjeta(denominacion, login, pass)
    actualizarEstatusTarj(login)
    guardarLog(4, login, "", denominacion, "Compra de Tarjeta " & login & " por "
& denominacion)

    Return True
Catch e As Exception
    mandarMail("Error al comprar tarjeta con Tarjeta con TC : " & e.ToString)
    Return False
End Try
End If
Else
    guardarPago(consecutivo, -1, APIPayWorks.ToString, Now(), Now())
    If APIPayWorks.TextoError = "" Then
        respuesta = "Por el momento el servicio no se encuentra disponible."
    Else
        respuesta = "Tarjeta inválida/expirada, favor de verificar su tarjeta."
    End If
    Return False
End If
Catch ex As Exception
    guardarPago(consecutivo, -1, "Error: tarjeta de crédito inválida - " & ex.ToString,
Now(), Now())
    mandarMail("Error al comprar tarjeta: " & ex.ToString)
    respuesta = "La tarjeta de crédito no se pudo procesar, favor de verificar los datos e
intentar de nuevo."
    Return False
End Try
End Function

```

6.2.3 Módulos Administrativos

A continuación los servicios administrativos utilizados para las tarjetas prepagadas. Los procesos que se explican a continuación fueron integrados al sistema de prepago Internet. Estos módulos se utilizan para otros productos de Marcatel.

6.2.3.1 Tarifas para *Dial-up*

El sistema de tarifas para cuentas de *Dial-up* solo cuenta con su base de datos. Los responsables de las tarifas modifican directamente la base de datos por medio de *stored procedures* en SQL Server 2000.

6.2.3.2 Conciliaciones Bancarias

El proceso de conciliaciones bancarias manda llamar a través del objeto bancario un método que devuelve un archivo de texto de todas las transacciones bancarias hechas hasta la última vez revisados. Esta información, que esta en un archivo de texto separado por comas, se compara con las transacciones registradas en la base de datos de Prepago en la tabla ActividadesTC. La comparación es hecha contra el banco y viceversa, revisando que los pagos hechos con tarjeta de crédito estén registrados en las dos bases de datos.

6.2.3.3 Generación de Archivo de Pagos

El archivo de pagos en un archivo de texto que genera un proceso de VB.NET. Una vez que el archivo de texto esta creado con la información de la tarjeta de crédito (solo primeros y últimos 4 dígitos de la tarjeta), el tipo de banco y tarjeta el archivo es transferido al proceso encargado de capturar esta información y copiarla al sistema de facturación de Marcatel.

6.2.3.4 Adaptación a bases de datos de tarjetas voz

La información de tarjetas de voz ya se encuentra disponible en una base de datos. Esta información es procesada por nuestro sistema para generar los nombres de usuario y contraseñas necesarias utilizados el siguiente formato:

El número de serie debe ser 4 dígitos y si la cifra es menor a 4, entonces se rellena con ceros del lado izquierdo, ejemplo:

4 -- 0004

Esto aplica también para el número de folio que se compone de 6 dígitos, ejemplo:

23 – 000025

Al juntar estas cifras la cuenta de usuario se compone de la unión del número de serie y el número de folio, ejemplo:

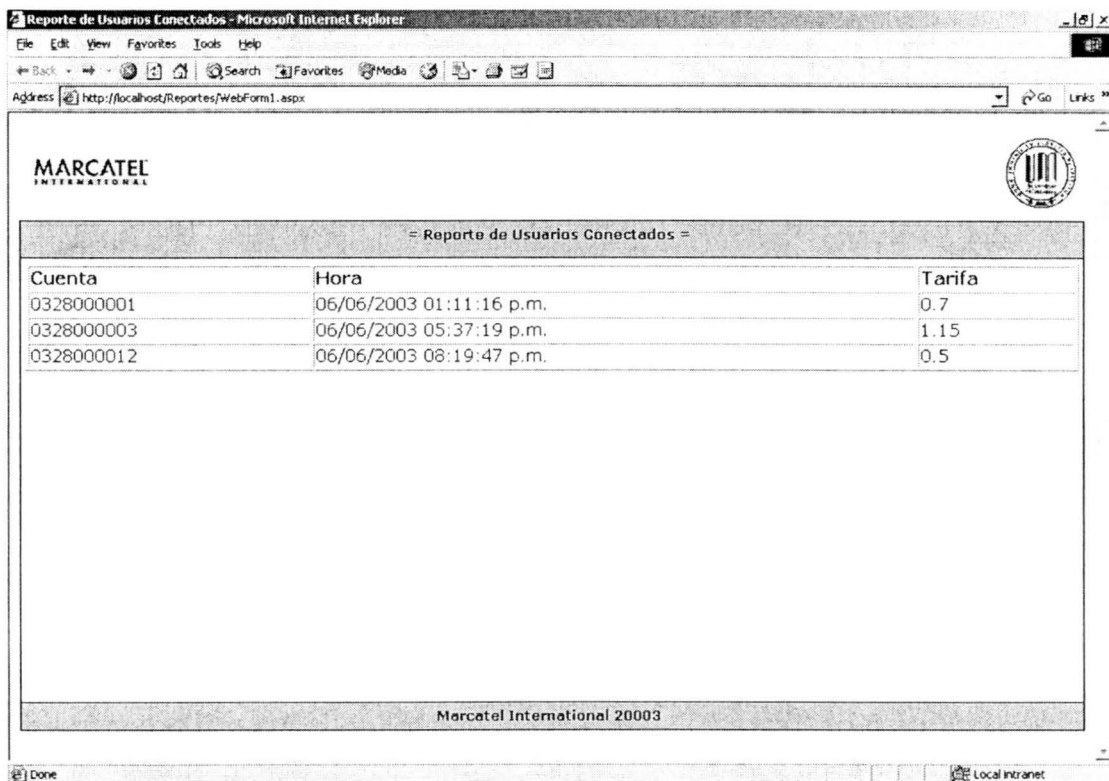
Nombre de usuario: 0004000025

La contraseña que utiliza la tarjeta se encuentra en el campo TarjetaNum de la tabla Serie de la base de datos de tarjetas.

6.2.3.5 Reportes establecidos

Se establecieron los siguientes reportes para prepago Internet. Los reportes mencionados utilizarán el diseño explicado en el punto 5.4:

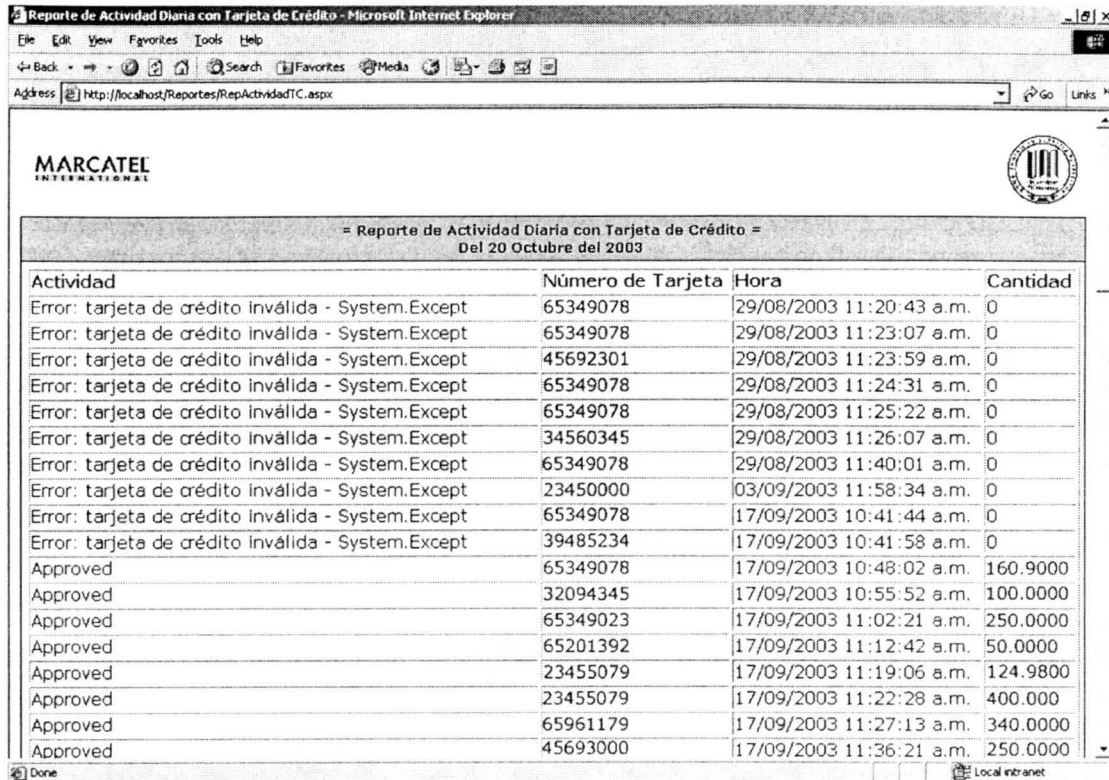
- Usuarios conectados: No tiene ninguna entrada, el administrador podrá utilizar esta página en ASP.NET para revisar que usuarios de Internet prepago se encuentran en ese momento. Los resultados a mostrar son: cuenta, hora a la que ingresó y Tarifa que se le esta cobrando. A continuación se muestra un ejemplo del reporte para usuarios conectados:



Cuenta	Hora	Tarifa
0328000001	06/06/2003 01:11:16 p.m.	0.7
0328000003	06/06/2003 05:37:19 p.m.	1.15
0328000012	06/06/2003 08:19:47 p.m.	0.5

Figura 6.1 Resultado de reporte para usuarios conectados al servicio prepago.

- Actividades con Tarjeta de Crédito: Como entrada recibe la fecha inicial y la fecha final. Los resultados a mostrar son: Numero de tarjeta (solo primeros y últimos cuatro dígitos de la tarjeta), la fecha de la transacción, la cantidad en pesos y si la actividad fue aprobada (*Approved*) o rechazada (*indicando la razón del rechazo*). A continuación se muestra un ejemplo del resultado del reporte para actividades con tarjeta de crédito:



= Reporte de Actividad Diaria con Tarjeta de Crédito =
Del 20 Octubre del 2003

Actividad	Número de Tarjeta	Hora	Cantidad
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:20:43 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:23:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	45692301	29/08/2003 11:23:59 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:24:31 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:25:22 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	34560345	29/08/2003 11:26:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:40:01 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	23450000	03/09/2003 11:58:34 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	17/09/2003 10:41:44 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	39485234	17/09/2003 10:41:58 a.m.	0
Approved	65349078	17/09/2003 10:48:02 a.m.	160.9000
Approved	32094345	17/09/2003 10:55:52 a.m.	100.0000
Approved	65349023	17/09/2003 11:02:21 a.m.	250.0000
Approved	65201392	17/09/2003 11:12:42 a.m.	50.0000
Approved	23455079	17/09/2003 11:19:06 a.m.	124.9800
Approved	23455079	17/09/2003 11:22:28 a.m.	400.0000
Approved	65961179	17/09/2003 11:27:13 a.m.	340.0000
Approved	45693000	17/09/2003 11:36:21 a.m.	250.0000

Figura 6.2 Resultado de reporte para actividades con tarjeta de crédito en Internet.

- **Cobro de Tarifas:** A través de un rango de fecha inicial y final, se mostrará el cobro de tarifas por minuto del servicio Internet prepago. Los resultados a mostrar son: la cuenta a la que se le cobró, la fecha y hora y la tarifa en pesos. A continuación se muestra un ejemplo del reporte para cobro de tarifas diaria:

= Reporte de Actividad Diaria =
Del 20 Octubre del 2003

Cuenta	Hora	Tarifa
acerotek	06/11/2003 04:07:00 p.m.	0.65
acerotek	06/11/2003 04:07:00 p.m.	0.65
acerotek	06/11/2003 04:14:13 p.m.	0.65
acerotek	06/11/2003 04:14:13 p.m.	0.65
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
acerotek	06/11/2003 04:14:34 p.m.	0.65
acerotek	06/11/2003 04:14:34 p.m.	0.65
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
acerotek	06/11/2003 04:15:03 p.m.	0.65
acerotek	06/11/2003 04:15:03 p.m.	0.65

Figura 6.3 Resultado de reporte de cobro diario de tarifas a cuentas de prepago.

6.3 Gestión de Incidencias

A continuación se muestra la sección de gestión de incidencias, donde se muestran los resultados de las pruebas de desarrollo e integración completa del sistema bajo un ambiente de pruebas. Los resultados se encuentran en tablas donde se especificó el proceso y funcionalidad a probar y su resultado. La mayoría de las pruebas de validación fueron exitosas al primer intento gracias a una herramienta que valida el contenido de los campos mediante una expresión regular y otra que forzó el campo a tener un valor. Por medio de estas expresiones, no se tuvo ningún problema al validar los datos de cada campo.

6.3.4 Módulo de Consulta de Saldo

A continuación se presentan las pruebas elaboradas para el módulo de consulta de saldo:

Nombre del proceso	Consulta de saldo
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero(válido)
Resultado esperado	Número de folio aceptado y despliegue de la información de la tarjeta de prepago
Resultado obtenido	Folio aceptado y despliegue de información correcta
Comentarios	

Tabla 6.3 Prueba de consulta de saldo para validación de folio 1

Nombre del proceso	Consulta de saldo
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero(no válido)
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.4 Prueba de consulta de saldo para validación de folio 2

Nombre del proceso	Consulta de saldo
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	String, decimal y caracteres especiales
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.5 Prueba de consulta de saldo para validación de folio 3

Nombre del proceso	Consulta de saldo
Funcionalidad a probar	Validación de campo de password en captura
Tipo de prueba	Captura de datos
Tipo de Dato(s)	Numérico, string, caracteres especiales
Resultado esperado	Despliegue del password con máscara de caracteres
Resultado obtenido	Pasword aparece con caracteres "*"
Comentarios	

Tabla 6.6 Prueba de consulta de saldo para validación de folio 4

Nombre del proceso	Consulta de saldo
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Vacío
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.7 Prueba de consulta de saldo para validación de folio 5

6.3.5 Módulo de Traspaso de Saldo de Tarjeta a Tarjeta

A continuación se muestran los resultados obtenidos de las pruebas del módulo traspaso de saldo:

Nombre del proceso	Traspaso de saldo
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Vacío
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.8 Prueba de traspaso de saldo para folios de tarjeta 1

Nombre del proceso	Traspaso de saldo
Funcionalidad a probar	Validación de folios de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numéricos enteros (válidos)
Resultado esperado	Número de folios aceptados y despliegue de la información de la tarjeta de prepago con el traspaso realizado
Resultado obtenido	Folios aceptados y despliegue de información correcta con el traspaso exitoso
Comentarios	

Tabla 6.9 Prueba de traspaso de saldo para folios de tarjeta 2

Nombre del proceso	Traspaso de saldo
Funcionalidad a probar	Validación de folios de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numéricos enteros (no válidos). Números negativos.
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.10 Prueba de traspaso de saldo para folios de tarjeta 3

Nombre del proceso	Traspaso de saldo
Funcionalidad a probar	Validación de folios de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	String, decimal y caracteres especiales
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.11 Prueba de traspaso de saldo para folios de tarjeta 4

Nombre del proceso	Traspaso de saldo
Funcionalidad a probar	Validación de campo de password en captura
Tipo de prueba	Captura de datos
Tipo de Dato(s)	Numérico, string, caracteres especiales
Resultado esperado	Despliegue del password con máscara de caracteres
Resultado obtenido	Aparece el carácter “*”
Comentarios	

Tabla 6.12 Prueba de traspaso de saldo para folios de tarjeta 5

6.3.6 Módulo de Recarga de Tarjeta con Tarjeta de Crédito

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero (válido)
Resultado esperado	Número de folio aceptado
Resultado obtenido	Folio aceptado
Comentarios	

Tabla 6.13 Prueba de recarga para folios de tarjeta 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero (no válido)
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.14 Prueba de recarga para folios de tarjeta 2

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	String, decimal y caracteres especiales
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.15 Prueba de recarga para folios de tarjeta 3

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Vacío
Resultado esperado	Despliegue de mensaje de error - "número de folio no válido"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.16 Prueba de recarga para folios de tarjeta 4

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de cantidad a recargar
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico decimal mayor a 100.00
Resultado esperado	Cantidad aceptada
Resultado obtenido	Se acepto la cantidad de 100.00
Comentarios	

Tabla 6.17 Prueba para validación de cantidad a recargar 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de cantidad a recargar
Tipo de prueba	Validación
Tipo de Dato(s)	String, decimal y caracteres especiales
Resultado esperado	Despliegue de mensaje de error - "cantidad no válido"
Resultado obtenido	Error - "cantidad no válido"
Comentarios	

Tabla 6.18 Prueba para validación de cantidad a recargar 2

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de cantidad a recargar
Tipo de prueba	Validación
Tipo de Dato(s)	Vacío
Resultado esperado	Despliegue de mensaje de error - "cantidad no válida"
Resultado obtenido	Error - "cantidad no válida"
Comentarios	

Tabla 6.19 Prueba para validación de cantidad a recargar 3

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de Tipo de Tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Selección del combo de tipo válido
Resultado esperado	Tipo aceptada
Resultado obtenido	Se acepto el tipo de tarejta
Comentarios	

Tabla 6.20 Prueba para validación de tipo de tarjeta de crédito 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de Tipo de Tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	No seleccionar
Resultado esperado	Despliegue de mensaje de error - "Seleccione un tipo de tarjeta"
Resultado obtenido	Error - "Seleccione un tipo de tarjeta"
Comentarios	

Tabla 6.21 Prueba para validación de tipo de tarjeta de crédito 2

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación del banco
Tipo de prueba	Validación
Tipo de Dato(s)	Selección del combo de banco
Resultado esperado	Banco Aceptado
Resultado obtenido	Se acepto el Banco capturado
Comentarios	

Tabla 6.22 Prueba para validación de tipo de banco 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación del banco
Tipo de prueba	Validación
Tipo de Dato(s)	No seleccionar
Resultado esperado	Despliegue de mensaje de error - "Seleccione un banco"
Resultado obtenido	Error - "Seleccione un banco"
Comentarios	

Tabla 6.23 Prueba para validación de tipo de banco 2

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación del número de tarjeta de crédito
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico (16 dígitos) válido
Resultado esperado	Tarjeta de crédito aceptada
Resultado obtenido	Se aceptó la tarjeta de crédito
Comentarios	La aceptación de la transacción depende del saldo de la tarjeta

Tabla 6.24 Prueba para número de tarjeta de crédito 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación del número de tarjeta de crédito
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico inválido (menor y mayor a 16 dígitos)
Resultado esperado	Despliegue de mensaje de error - "número de tarjeta no válida"
Resultado obtenido	Error - "número de tarjeta no válido"
Comentarios	

Tabla 6.25 Prueba para número de tarjeta de crédito 2

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación del número de tarjeta de crédito
Tipo de prueba	Validación
Tipo de Dato(s)	String, decimal y caracteres especiales
Resultado esperado	Despliegue de mensaje de error - "número de tarjeta no válida"
Resultado obtenido	Error - "número de tarjeta no válido"
Comentarios	

Tabla 6.26 Prueba para número de tarjeta de crédito 3

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de folio de tarjeta
Tipo de prueba	Validación
Tipo de Dato(s)	Vacío
Resultado esperado	Despliegue de mensaje de error - "número de folio no válida"
Resultado obtenido	Error - "número de folio no válido"
Comentarios	

Tabla 6.27 Prueba para validación de folio de tarjeta

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de cantidad a recargar
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico decimal mayor a 100.00
Resultado esperado	Cantidad aceptada
Resultado obtenido	Se aceptó la cantidad de 100.00
Comentarios	

Tabla 6.28 Prueba para validación de cantidad a recargar 1

Nombre del proceso	Recarga de tarjeta
Funcionalidad a probar	Validación de cantidad a recargar
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico decimal menor a 100.00 o negativo
Resultado esperado	Despliegue de mensaje de error - "cantidad no válido"
Resultado obtenido	Error - "cantidad no válido"
Comentarios	

Tabla 6.29 Prueba para cantidad a recargar 2

6.3.7 Módulo de Compra de Tarjeta con Tarjeta de Crédito

Para el módulo de compra de tarjetas por Internet, se utilizaron las mismas pruebas de las tablas 6.20 a 6.30 para transacciones bancarias. Las siguientes tablas muestran las pruebas para el módulo de compra de tarjeta independientemente de las funcionalidades bancarias:

Nombre del proceso	Compra de tarjeta
Funcionalidad a probar	Validación de inventario
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero que se encuentra en combo. 500 pesos.
Resultado esperado	Despliegue de mensaje de error - "tarjeta no disponible por el momento"
Resultado obtenido	Cero registros encontrados
Comentarios	Error en query de búsqueda de tarjetas activadas. Al corregirse la prueba tuvo éxito

Tabla 6.30 Prueba para validación de inventario 1

Nombre del proceso	Compra de tarjeta
Funcionalidad a probar	Validación de inventario
Tipo de prueba	Validación
Tipo de Dato(s)	Numérico entero que se encuentra en combo. 100 pesos.
Resultado esperado	Compra satisfactoria
Resultado obtenido	Pantalla regresa sin tener actividad.
Comentarios	Se tuvieron problemas al realizar la transacción debido a un problema con el objeto bancario. El formato de fecha utilizado por Banorte fue cambiado dos veces. Se obtuvo el nuevo objeto bancario y la compra fue satisfactoria.

Tabla 6.31 Prueba para validación de inventario 2

Nombre del proceso	Compra de tarjeta
Funcionalidad a probar	Envío de correo electrónico con información de tarjeta adquirida
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Formato de correo electrónico
Resultado esperado	Envío de e-mail a cuenta deseada
Resultado obtenido	Envío el e-mail pero no había datos, página marcaba error de validación pero continuaba con el proceso.
Comentarios	<p>Se encontró un error en las variables locales que contienen los datos de la compra. Las variables perdían su valor después de hacer el submit.</p> <p>Se corrigió el error convirtiendo las variables de tipo local a sesión.</p> <p>El error de validación se corrigió desactivando todos los procesos de validación de la página de compra, que se utilizaban para validar los datos de la tarjeta.</p>

Tabla 6.32 Prueba para envío de correo electrónico

6.3.8 Módulo de Monitoreo de Saldos

Para el siguiente módulo se probó la pura funcionalidad, la información que recibe es fija y si por algún motivo hay una inconsistencia en esta el modulo sigue funcionando. En la siguiente tabla se muestran las pruebas:

Nombre del proceso	Monitoreo de saldos
Funcionalidad a probar	Muestra de saldo correcto para tarjeta
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Envía dirección <i>IP</i> y recibe string de información por protocolo <i>TCP/IP</i>
Resultado esperado	Muestra de saldo correcto
Resultado obtenido	Saldo mostrado fue el correcto
Comentarios	

Tabla 6.33 Prueba de monitoreo de saldos para muestra de saldo correcto

Nombre del proceso	Monitoreo de saldos
Funcionalidad a probar	Prueba de clientes múltiples.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Envía dirección <i>IP</i> y recibe string de información por protocolo <i>TCP/IP</i>
Resultado esperado	Muestra de saldo correcto en los dos clientes.
Resultado obtenido	Muestra de saldo correcto en un cliente y el otro no hace nada.
Comentarios	Se encontraron problemas cuando se hacían peticiones múltiples con varios clientes. El problema se corrigió durmiendo el proceso del servidor 1 segundo para no perder la petición de otro cliente.

Tabla 6.34 Prueba de monitoreo de saldos para clientes múltiples

Nombre del proceso	Monitoreo de saldos
Funcionalidad a probar	Prueba de desconexión.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	No aplica
Resultado esperado	Mostrar estatus "Desconectado" al remover cable de red. Esto simula una desconexión telefónica.
Resultado obtenido	Mostró estatus "Desconectado"
Comentarios	Se volvió a conectar el cable y el módulo se restauró automáticamente mostrando información de la tarjeta.

Tabla 6.35 Prueba de monitoreo de saldos para desconexión por falla

6.3.9 Monitoreo de Conexiones y Saldos

La estructura de tablas que se reutilizó para obtener la tarifa correcta a cobrar por minuto se probó

Nombre del proceso	Monitoreo de conexiones y saldos
Funcionalidad a probar	Selección de tarifa correcta al detectar conexión de usuario.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	<ul style="list-style-type: none"> - Numero al que se llamo de tipo varchar para determinar Área Geográfica. - Hora y fecha de tipo date/time para determinar horario.
Resultado esperado	Tabla de base de datos <i>UsersConnected</i> muestra tarifa correcta para usuario.
Resultado obtenido	<i>UsersConnected</i> mostró tarifa correcta para usuario.
Comentarios	Algunas veces la tarifa no se registraba en la tabla debido a falta de información en las tablas de tarifas para determinar el horario y lugar a donde esta comunicandose. No había registro para la fecha y hora actual. Al llenarlos devolvió la tarfia a utilizar.

Tabla 6.36 Prueba de monitoreo de conexiones y saldos para selcción de tarifa correcta

6.3.10 Generación de Cuentas Válidas para Tarjetas Prepago

A continuación se muestra el resultado obtenido para generar cuentas válidas de tarjetas prepago para internet según las bases de datos utilizadas para tarjetas prepago de voz:

Nombre del proceso	Generación de cuentas válidas de tarjetas prepago
Funcionalidad a probar	Generación correcta de cuentas prepago
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	<ul style="list-style-type: none"> - Numero de serie y folio tipo varchar. - TarjetaNum de tipo varchar para password.
Resultado esperado	<p>Generación de cuentas prepago en la tabla Nips_dialup con el siguiente formato:</p> <ul style="list-style-type: none"> - Numero de Serie: 38 - Numero de Folio : 1 - TarjetaNum: 1111111111 <p>Información que debe generar:</p> <p>Login: 0038000001 Password: 1111111111</p>
Resultado obtenido	<p>Login: 0038000001 Password: 1111111111</p>
Comentarios	Se hizo la misma prueba para 1000 cuentas y los resultados fueron satisfactorios.

Tabla 6.37 Prueba de generación de cuentas válidas

6.3.11 Pruebas sobre ambiente Cisco ACS Radius y Max TNT Server

Estos resultados representan pruebas individuales para obtener la información necesaria de los equipos responsables para conexiones de Internet. Las siguientes tablas muestran los resultados obtenidos de las pruebas:

Nombre del proceso	Ambiente Cisco ACS Radius y Max TNT Sever
Funcionalidad a probar	Obtener usuarios conectados del Max TNT Server por SNMP.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Entrada: String de datos alfanuméricos.
Resultado esperado	Lista de usuarios conectados.
Resultado obtenido	Se regreso una lista de usuario conectados.
Comentarios	El objeto es utilizado para otros procesos, no hubo mucho que validar.

Tabla 6.38 Prueba de ambiente MaxTNT

Nombre del proceso	Ambiente Cisco ACS <i>Radius</i> y <i>MaxTNT</i>
Funcionalidad a probar	Obtener registros de usuarios conectados y desconectados de <i>Cisco ACS Radius</i>
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Entrada: Query de SQL que busca los usuarios conectados y desconectados en la última hora.
Resultado esperado	Lista de usuarios conectados y desconectados en la última hora.
Resultado obtenido	Muestra los usuarios conectados y desconectados.
Comentarios	La búsqueda se probó en el ambiente real de RADIUS, para escritura, modificación o eliminación de datos se usará una tabla de prueba con la misma información.

Tabla 6.39 Prueba de ambiente Radius

6.3.12 Generación de archivo de pagos para Facturación

A continuación se muestra las pruebas hechas para determinar si los pagos por tarjeta de crédito se registraron en el sistema de facturación.

Nombre del proceso	Generación de archivo de pagos para Facturación
Funcionalidad a probar	Generación de archivo de texto con últimos pagos hechos.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	Numero de tarjeta de tipo varchar Nombre de banco de tipo varchar Cantidad pagada de tipo decimal Numero de cuenta para identificar compras en Internet de tipo varchar
Resultado esperado	Archivo de texto con la información necesaria para registrar en facturación.
Resultado obtenido	Se generó archivo de texto.
Comentarios	Con este archivo de texto se agregaron los datos al sistema de facturación con un proceso fuera de los objetivos de nuestro sistema. Este mismo es usado para el sistema de Pago de factura en línea

Tabla 6.40 Prueba de generación de archivos a facturación

6.4 Pruebas piloto (maqueta)

Esta sección muestra los resultados de las pruebas sobre todos los módulos implementados en un ambiente de prueba. Los resultados se muestran de la misma forma que la sección anterior. Nos enfocamos en pruebas que tengan dependencia con otros módulos, ya que es necesario probar el funcionamiento del sistema completo y asegurar que se cumplan los objetivos. Los resultados son los siguientes:

Nombre del proceso	Monitoreo de Saldos por cliente
Funcionalidad a probar	Deducción de saldo por aplicación de tarifa.
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	No Aplica
Resultado esperado	Deducción de saldo cada minuto.
Resultado obtenido	Se deduce el saldo
Comentarios	

Tabla 6.41 Prueba piloto de monitoreo de saldos por cliente para deducir tarifa

Nombre del proceso	Monitoreo de conexiones y saldos
Funcionalidad a probar	Registro de conexiones activas al iniciar el servicio de monitoreo de conexiones y saldos
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	No Aplica
Resultado esperado	Obtener la lista de usuarios activos en MaxTNT y almacenar en la tabla de UsersConnected
Resultado obtenido	Tabla de Usersconnected contiene los usuarios conectados al iniciar el servicio.
Comentarios	

Tabla 6.42 Prueba piloto de monitoreo de conexiones y saldos para registrar conexiones

Nombre del proceso	Monitoreo de conexiones y saldos
Funcionalidad a probar	Desconexión forzada por falta de saldo
Tipo de prueba	Funcionalidad
Tipo de Dato(s)	identificador de sesión de tipo varchar
Resultado esperado	Desconectar al usuario por medio de su identificador de sesión en el MaxTNT server.
Resultado obtenido	Usuario desconectado del servicio
Comentarios	La realización de la desconexión varia dependiendo de la carga del servidor. No excede 60 segundos para la mauquina de prueba.

Tabla 6.43 Prueba piloto de monitoreo de conexiones y saldos para desconexión forzada por falta de saldo

6.5 Plan de Despliegue (Liberación)

La liberación del sistema se pronosticó para Enero del 2004 y dependerá de la resolución de la situación legal de derechos reservados que tiene Todito.com para explotar el producto hasta el 2007. La liberación del servicio se planea hacer por región geográfica. , MarcaTel cuenta con 5 centrales ubicadas en Monterrey, Guadalajara, Ciudad de México, Puebla y Torreón.

El servicio se planea liberar en Monterrey inicialmente para observar la demanda de este producto y corregir fallas o retroalimentación que se reciba de los clientes del servicio. Si el sistema presenta una fuerte demanda se comienza a extender el producto a las demás centrales cubriendo sus respectivas áreas.

6.6 Captura de 2da Incidencia (Versiones Futuras)

El proyecto necesita atención en las áreas administrativas de reportes. Se tuvieron muchos problemas para hacer pruebas completas con la generación de lotes de tarjetas, catálogos de proveedores y tarjetas, generación de archivo de impresión, y las consultas y reportes a aprovechar del sistema de tarjetas de voz.

Esto se debió a los cambios que decidieron los responsables en cuanto al desarrollo. En un principio estos procesos estaban planeados para desarrollo y terminaron como parte de una integración al sistema de tarjetas prepagadas de voz. No se pudo mostrar con claridad el funcionamiento de los módulos o procesos que se adaptaron a prepago Internet.

La versión futura para este producto es la fusión de la tarjeta prepago de voz con la tarjeta prepago Internet. Así mismo, se puede mejorar el uso de proceso mencionados anteriormente y se forma un producto nuevo que se puede utilizar para llamadas telefónicas y uso de Internet.

El objeto bancario requiere del número de la tarjeta de crédito y la fecha de expiración para hacer una transacción. Se recomienda en otra versión cambiar el objeto para que utilice el número de tarjeta, fecha de expiración y código verificador de la tarjeta.

6.7 Resumen de herramientas y su aplicación al proyecto.

A continuación se muestran las herramientas utilizadas en el proyecto y su aplicación:

Herramienta	Aplicación en el proyecto
Microsoft .NET	Es el ambiente que maneja MarcaTel para sus aplicaciones y que cuentan todas las bases de datos incluyendo la de Radius.
ASP.NET	Es donde serán elaboradas las aplicaciones para el cliente.
Microsoft Visual Studio .NET y Framework	Las aplicaciones a nivel administrador recaen en estas herramientas.
MaxTNT y Cisco ACS Radius	Equipo y herramientas de tecnologías que representan la entrada principal a los servicios de Internet y que permite mediante Radius su autorización y administración.

Tabla 6.44 Resumen de herramientas a utilizar.

6.8 Conclusiones del capítulo

En este capítulo se mostró la construcción del sistema utilizando las herramientas de .NET. Estas han sido de gran ayuda para desarrollar este sistema, la clave se encuentra en la facilidad de validaciones y herramientas especializadas para publicación de servicios en Internet a través del componente de expresiones regulares. La gestión de incidencias ha servido para asegurar una buena funcionalidad de los módulos individuales. Las pruebas piloto, o sobre el sistema completo, permiten asegurar la interoperabilidad de los diferentes módulos creados. El uso de Radius permitió autorizar a los usuarios que utilizan Internet. Max TNT se utilizó como puerta de enlace entre la línea telefónica y el servicio de Internet de Marcatel. También fue útil para la desconexión de tarjetas prepagadas con saldo agotado.

CONCLUSIONES

Al establecer la funcionalidad final para las tarjetas prepagadas utilizando la metodología Microsoft Solution Framework, todos los involucrados obtienen una idea clara de lo que debe incluir prepago Internet. La planeación permitió un claro punto de vista para planear detalladamente el producto de Internet en conjunto con Radius y MaxTNT. La fase de pruebas individuales permitió la eliminación de errores simples de desarrollo para los módulos cliente. La fase de pruebas integrales permitió eliminar errores más complejos entre módulos donde una prueba simple, como validación de número de tarjeta de crédito no podía detectarse con facilidad. La última fase permitió planear la liberación del producto y comenzar la planeación de mejora de los módulos administrativos. Ésta metodología se recomienda para desarrollo de procesos donde se requiere retroalimentación y cooperación de todos los involucrados. También permite a los involucrados conocer claramente los objetivos, manteniendo la integridad entre departamentos con respecto a un proyecto de desarrollo.

La plataforma .NET ha sido valiosa para la inter-operación con Radius y las herramientas bancarias proporcionadas por Banorte. La única desventaja visible de la plataforma de desarrollo es el alto número de recursos que requiere para funcionar. Fue sumamente lenta la compilación de los módulos de prepago al hacer cambios o elaborar pruebas. Permitted

organizar el código elaborado de forma automática, facilitando la detección de errores o secciones donde se necesitaba hacer cambios. El uso de programación orientada a objetos en esta plataforma, permitió una gran flexibilidad para cambios en el código. Los problemas encontrados con el módulo bancario son un buen ejemplo. Los equipos MaxTNT y ACS Radius tuvieron un papel importante durante el proyecto, dando el servicio de Internet a clientes. Sin estas plataformas, el producto de tarjetas prepagadas no hubiera podido realizarse en un tiempo tan corto.

El proyecto para la liberación de tarjetas prepagadas Internet ha servido para adquirir experiencia en el desarrollo de sistemas de productos nuevos. Este ambiente de desarrollo permitió no solo negociar y establecer los requerimientos, sino también investigar el mercado por productos parecidos. Los beneficios tangibles fueron una estimación en las ventas y no una comparación entre procesos internos fijos de la compañía. Es un ambiente diferente al rediseño o desarrollo de procesos internos para una compañía y es muy útil para entender las necesidades del cliente para un producto.

Referencias Bibliográficas

[Balboa, 2003] Balboa, Juan Carlos, Entrevistas y reuniones para definición de necesidades, Monterrey, Nuevo León, los días 25, 26 de febrero y 4, 11 y 12 de Marzo del 2003.

[Behforooz, 1996] Delgado Albert, Fredrick J. Hudson, Software Engineering Fundamentals, Oxford University Press, Estados Unidos, 2001

[Cisco ACS Radius, 2003] Cisco ACS Radius, Cisco IOS Technologies: RADIUS Support in Cisco IOS Software en http://www.cisco.com/en/US/tech/tk583/tk547/technologies_white_paper09186a00800a3c92.shtml

[Delgado 2001] Delgado Albert, Microsoft SQL Server 2000: Edición Especial, Prentice Hall, España, 2001

[Hill, 2001] Hill, Joshua, Un Análisis del Protocolo de Autorización de RADIUS, en <http://www.untruth.org/~josh/security/radius/radius-auth.html>

[Leal, 2003] Leal, Ruth, Definición de sistema administrativo para fusión con tarjetas prepagadas de voz, Monterrey, Nuevo León, el día 12 de Agosto del 2003.

[Lucent, 2003] Lucent, Descripción de Lucent MAXTnt Universal Gateway, en <http://www.lucent.com/products/solution/0,,CTID+2017-STID+10443-SOID+589-LOCL+1,00.html>

[Marcatel, 2003] MarcaTel International, Descripción de la compañía en <http://www.marcatel.com.mx/compania.htm>

[Microsoft, 2003] Microsoft TechNet, Metodología Microsoft Solution Framework, en <http://www.microsoft.com/spain/technet/fases/msf.asp>

[MSF, 2002] Microsoft Solutions Framework White Paper: MSF Process Model v 3.1, Microsoft, Estados Unidos, Junio 2002

[Patton, 2001] Patton Robert A., Designing SQL Server 2000 Databases for .net Enterprise Servers, Syngress Press, Estados Unidos, 2001.

[Pereyra, 1999] Rene Angel, Pereyra Gonzalez, Sistema administrador de la base de datos de clientes morosos y de apoyo a la toma de decisiones, San Pedro Garza Garcia, NL, 1999.

[Pereyra, 2003] Pereyra, Rene, Definición de tarifas para tarjetas prepagadas, Monterrey, Nuevo León, el 12 de Agosto del 2003.

[Pertuosos, 2002] Pertuosos Evangelos, Mastering Visual Basic .NET, SYBEX Inc., Estados Unidos, 2002.

[Platt, 2001] Platt, David S., Introducing Microsoft .NET, Microsoft Press, Estados Unidos, 2001.

[Terra, 2003] Terra Prepago, Descripción de la tarjeta de prepago en <http://www.terra.com.mx/terraprepago/>

[Todito.com, 2003] Todito.com, Descripción de la tarjeta de prepago en <http://www.toditocard.com/cgi-bin/tcardB/toditocard/ahorra.html>.

[Ventas, 2003] Ventas, Junta con el departamento de ventas para establecer, Monterrey, Nuevo León, el día 4 de Agosto del 2003.

[Von Hauske, 2003] Von Hauske, Erich, Entrevistas y reuniones definición de necesidades, Monterrey, Nuevo León, los días 4,11 y 12 de Marzo del 2003.

APÉNDICE

Referencias bibliográficas de Internet

[Hill,2001]

An Analysis of the RADIUS Authentication Protocol

by Joshua Hill, [InfoGard Laboratories](#)

Copyright © 2001, Joshua Hill

You may distribute unaltered copies of this document without restriction

A current version of this article is maintained at:

<http://www.untruth.org/~josh/security/radius/>

Please send comments to josh-radius@untruth.org

Last Modified Sat Nov 24 23:58:41 PST 2001

See the Document Change History section for a history of modification

1 Introduction

RADIUS is a widely used protocol in network environments. It is commonly used for embedded network devices such as routers, modem servers, switches, etc. It is used for several reasons:

- The embedded systems generally cannot deal with a large number of users with distinct authentication information. This requires more storage than many embedded systems possess.
- RADIUS facilitates centralized user administration, which is important for several of these applications. Many ISPs have tens of thousands, hundreds of thousands, or even millions of users. Users are added and deleted continuously throughout the day, and user authentication information changes constantly. Centralized administration of users in this setting is an operational requirement.
- RADIUS consistently provides some level of protection against a sniffing, active attacker. Other remote authentication protocols provide either intermittent protection, inadequate protection or non-existent protection. RADIUS's primary competition for remote authentication is TACACS+ and LDAP. LDAP natively provides no protection against sniffing or active attackers. TACACS+ is subtly flawed, as discussed by Solar Designer in his advisory.
- RADIUS support is nearly omni-present. Other remote authentication protocols do not have consistent support from hardware vendors, whereas RADIUS is uniformly supported. Because the platforms on which RADIUS is implemented on are often embedded systems, there are limited opportunities to support additional protocols. Any changes to the RADIUS protocol would have to be at least minimally compatible with pre-existing (unmodified) RADIUS clients and servers.

RADIUS is currently the de-facto standard for remote authentication. It is very prevalent in both new and legacy systems.

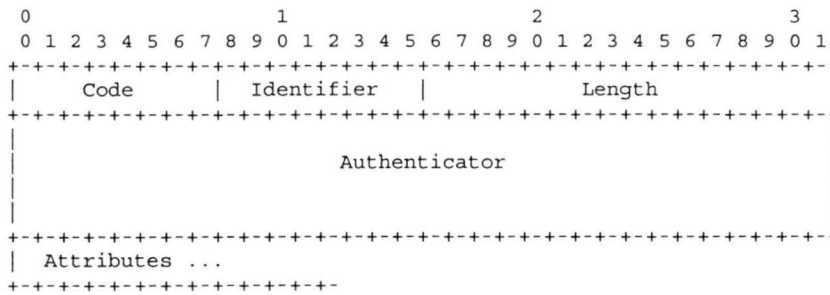
1.1 Applicability

This analysis deals with some of the characteristics of the base RADIUS protocol and of the User-Password attribute. Depending on the mode of authentication used, the described User-Password weaknesses may or may not compromise the security of the underlying authentication scheme. A complete compromise of the User-Password attribute would result in the complete compromise of the normal Username/Password or PAP authentication schemes, because both of these systems include otherwise unprotected authentication information in the User-Password attribute. On the other hand when a Challenge/Response system is in use, a complete compromise of the User-Password attribute would only expose the underlying Challenge/Response information to additional attack, which may or may not lead to a complete compromise of the authentication system, depending on the strength of the underlying authentication system.

This analysis does not cover the RADIUS protocol's accounting functionality (which is, incidentally, also flawed, but normally doesn't transport information that must be kept confidential).

2 Protocol Summary

A summary of the RADIUS packet is below (from the RFC):



The code establishes the type of RADIUS packet. The codes are:

Value	Description
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (experimental)
13	Status-Client (experimental)
255	Reserved

The identifier is a one octet value that allows the RADIUS client to match a RADIUS response with the correct outstanding request.

The attributes section is where an arbitrary number of attribute fields are stored. The only pertinent attributes for this discussion are the User-Name and User-Password attributes.

This description will concentrate on the most common type of RADIUS exchange: An Access-Request involving a username and user password, followed by either an Access-Accept, Access-Reject or a failure. I will refer to the two participants in this protocol as the client and the server. The client is the entity that has authentication information that it wishes to validate. The server is the entity that has access to a database of authentication information that it can use to validate the client's authentication request.

2.1 Initial Client Processing

The client creates an Access-Request RADIUS packet, including at least the User-Name and User-Password attributes.

The Access-Request packet's identifier field is generated by the client. The generation process for the identifier field is not specified by the RADIUS protocol specification, but it is usually implemented as a simple counter that is incremented for each request.

The Access-Request packet contains a 16 octet Request Authenticator in the authenticator field. This Request authenticator is a randomly chosen 16 octet string.

This packet is completely unprotected, except for the User-Password attribute, which is protected as follows:

The client and server share a secret. That shared secret followed by the Request Authenticator is put through an MD5 hash to create a 16 octet value which is XORed with the password entered by the user. If the user password is greater than 16 octets, additional MD5 calculations are performed, using the previous ciphertext instead of the Request Authenticator.

More formally:

Call the shared secret S and the pseudo-random 128-bit Request Authenticator RA . The password is broken into 16-octet blocks p_1, p_2, \dots, p_n , with the last block padded at the end with '0's to a 16-octet boundary. The ciphertext blocks are c_1, c_2, \dots, c_n .

$$c_1 = p_1 \text{ XOR MD5}(S + RA)$$

$$c_2 = p_2 \text{ XOR MD5}(S + c_1)$$

.

.

$$c_n = p_n \text{ XOR MD5}(S + c_{n-1})$$

The User-Password attribute contains $c_1+c_2+\dots+c_n$, Where $+$ denotes concatenation.

2.2 Server Processing

The server receives the RADIUS Access-Request packet and verifies that the server possesses a shared secret for the client. If the server does not possess a shared secret for the client, the request is silently dropped.

Because the server also possesses the shared secret, it can go through a slightly modified version of the client's protection process on the User-Password attribute and obtain the unprotected password. It then uses its authentication database to validate the username and password. If the password is valid, the server creates an Access-Accept packet to send back to the client. If the password is invalid, the server creates an Access-Reject packet to send back to the client.

Both the Access-Accept packet and the Access-Reject packet use the same identifier value from the client's Access-Request packet, and put a Response Authenticator in the Authenticator field. The Response Authenticator is the MD5 hash of the response packet with the associated request packet's Request Authenticator in the Authenticator field, concatenated with the shared secret.

That is, $\text{ResponseAuth} = \text{MD5}(\text{Code}+\text{ID}+\text{Length}+\text{RequestAuth}+\text{Attributes}+\text{Secret})$ where $+$ denotes concatenation.

2.3 Client Post Processing

When the client receives a response packet, it attempts to match it with an outstanding request using the identifier field. If the client does not have an outstanding request using the same identifier, the response is silently discarded. The client then verifies the Response Authenticator by performing the same Response Authenticator calculation the server performed, and then comparing the result with the Authenticator field. If the Response Authenticator does not match, the packet is silently discarded.

If the client received a verified Access-Accept packet, the username and password are considered to be correct, and the user is authenticated. If the client received a verified Access-Reject message, the username and password are considered to be incorrect, and the user is not authenticated.

3 RADIUS Issues

The RADIUS protocol has a set of vulnerabilities that are either caused by the protocol or caused by poor client implementation and exacerbated by the protocol. The vulnerabilities that follow arose during a somewhat shallow exploration of the protocol; this is not expected to be a complete list of vulnerabilities of the RADIUS protocol, these are merely the vulnerabilities that presented themselves to the reviewer.

3.1 Response Authenticator Based Shared Secret Attack

The Response Authenticator is essentially an ad hoc MD5 based keyed hash. This primitive facilitates an attack on the shared secret. If an attacker observes a valid Access-Request packet and the associated Access-Accept or Access-Reject packet, they can launch an off-line exhaustive attack on the shared secret. The attacker can pre-compute the MD5 state for $(\text{Code}+\text{ID}+\text{Length}+\text{RequestAuth}+\text{Attributes})$ and then resume the hash once for each shared secret guess. The ability to pre-compute the leading sections of this keyed hash primitive reduces the computational requirements for a successful attack.

3.2 User-Password Attribute Cipher Design Comments

The User-Password protection scheme is a stream-cipher, where an MD5 hash is used as an ad hoc pseudorandom number generator (PRNG). The first 16 octets of the stream cipher display the same properties as a synchronous stream cipher. After the first 16 octets, the stream cipher state integrates the previous ciphertext, and becomes more accurately described as a self-synchronizing stream cipher.

The security of the cipher rests on the strength of MD5 for this type of use and the selection of the shared secret. It is unclear what the requirements for this cipher are, so it is unclear if the MD5 function is appropriate for this use. MD5 is not designed to be a stream cipher primitive, it is designed to be a cryptographic hash. This sort of misuse of cryptographic primitives often leads to subtly flawed systems.

3.3 User-Password Attribute Based Shared Secret Attack

Because of the selection of a stream cipher for protection of the User-Password attribute, an attacker can gain information about the Shared Secret if they can observe network traffic and attempt an authentication. The attacker attempts to authenticate to the client with a known password. The attacker then captures the resulting Access-Request packet and XORs the protected portion of the User-Password attribute with the password they provided to the client. This results in the value of the MD5(Shared Secret + Request Authenticator) operation. The Request Authenticator is known (it is in the client's Access-Request packet), so the attacker can launch an off-line exhaustive attack on the shared secret. Note, though, that the attacker cannot pre-compute the MD5 state of the hash for the Request Authenticator, because the Request Authenticator is hashed second.

3.4 User-Password Based Password Attack

The use of a stream cipher to protect the User-Password attribute results in a vulnerability that allows an attacker to circumvent any authentication rate limits imposed by the client. The attacker first attempts to authenticate to the client using a valid username and a known (and likely incorrect) user password. The attacker then captures the resulting Access-Request packet and determines the result of the MD5(Shared Secret + Request Authenticator) operation (in the same way as in the previous attack). The attacker can then replay modified Access-Request packets, using the same Request Authenticator and MD5(Shared Secret + Request Authenticator) value, changing the password (and the associated User-Password attribute) for each replay. If the server does not impose user based rate limits, this will allow the attacker to efficiently perform an exhaustive search for the correct user password.

Note that the attacker can only use this method to attack passwords that are 16 characters or less, as the User-Password protection mechanism uses a chaining method that includes previous ciphertext in the state after the first 16 octets of output.

Any sort of strong data authentication in the Access-Request packet would make this attack impossible.

3.5 Request Authenticator Based Attacks

The security of RADIUS depends on the generation of the Request Authenticator field. The Request Authenticator must be both unique and non-predictable in order for the RADIUS implementation to be secure. The RADIUS protocol specification does not emphasize the importance of the Request Authenticator generation, so there are a large number of implementations that use poor PRNGs to generate the Request Authenticator. If the client uses a PRNG that repeats values (or has a short cycle), the protocol ceases to provide the intended level of protection.

The last two of these attacks require the attacker to cause the client to produce a particular identifier value. This is generally not particularly difficult, as identifiers were never meant as a security feature. The actual method of identifier generation is not specified by the protocol specification, but the most common method of generating the identifier is to increment a one octet counter for each request, and include the counter value as the identifier. Because the identifier generation is normally deterministic, it often doesn't increase the work factor very much at all. An attacker can insert a series of extra requests to the client, forcing the desired identifier to reoccur much more rapidly than it would normally. Even if the identifier were not generated in a readily attackable way, it would still only increase the work factor by 256 times.

3.5.1 Passive User-Password Compromise Through Repeated Request Authenticators

If the attacker can sniff the traffic between the RADIUS client and the RADIUS server, they can passively produce a dictionary of Request Authenticators, and the associated (protected) User-Password attributes. If the attacker observes a repeated Request Authenticator, they can remove any influence of the Shared Secret from the first 16 octets of the passwords by XORing the first 16 octets of the protected passwords together. This yields the first 16 octets of the two (now unprotected) user passwords XORed together.

The impact of this attack varies according to how good the user passwords are. If the users all chose random passwords of the same length, the attacker can gain nothing because no information about either password can be extracted. Unfortunately, this is a somewhat unlikely occurrence. In reality, users choose passwords of varying lengths (generally less than 16 characters) and of varying quality.

The easiest problem for the attacker to exploit is the case where the two passwords are of different lengths. Ideally for the attacker, the passwords are both less than 16 characters long and are significantly different lengths. In this situation, one of the passwords has more padding than the other, so the non-overlapping characters of the longer password are XORed with '0' (the

characters do not change). This results in the non-overlapping characters of the longer password being exposed to the attacker with no analysis.

More complex attacks are available if the attacker makes the assumption that the users chose low-entropy passwords. In this situation, the attacker can perform an intelligent dictionary attack guided by statistical analysis of the overlapping region. This dictionary attack can be further refined by noting the length of the two passwords and the trailing portion of the longer password, and then only trying passwords with this length and ending.

Even passwords longer than 16 characters are at risk from this attack, because the attacker still gains information about the first 16 characters of the password. This provides a firm basis for later attack, if nothing else.

3.5.2 Active User-Password Compromise through Repeated Request Authenticators

The attacker can attempt to authenticate many times using known passwords and intercept the generated Access-Request packets, extracting the Request Authenticator and User-Password attributes. The Attacker can then XOR the known password with the User-Password attribute and be left with the MD5(Shared Secret + Request Authenticator) value. The attacker generates a dictionary of Request Authenticator values and associated MD5(Shared Secret + Request Authenticator) values.

When the attacker sees a valid Access-Request packet that has a Request Authenticator value that is in the attacker's dictionary, the attacker can recover the first 16 octets from the protected region of the User-Password field by looking up the associated MD5(Shared Secret + Request Authenticator) value from the dictionary and XORing it with the intercepted protected portion of the User-Password attribute.

3.5.3 Replay of Server Responses through Repeated Request Authenticators

The attacker can build a dictionary of Request Authenticators, identifiers and associated server responses. When the attacker then sees a request that uses a Request Authenticator (and associated identifier) that is in the dictionary, the attacker can masquerade as the server and replay the previously observed server response.

Further, if the attacker can attempt to authenticate, causing the client to produce an Access-Request packet with the same Request Authenticator and identifier as a previously observed successful authentication, the attacker can replay the valid looking Access-Accept server response and successfully authenticate to the client without knowing a valid password.

3.5.4 DOS Arising from the Prediction of the Request Authenticator

If the attacker can predict future values of the Request Authenticator, the attacker can pose as the client and create a dictionary of future Request Authenticator values (with either the expected identifier, or with every possible identifier) and associated (presumably Access-Reject) server responses. The attacker can then masquerade as the server and respond to the client's (possibly valid) requests with valid looking Access-Reject packets, creating a denial of service.

3.6 Shared Secret Hygiene

The RADIUS standard specifically permits use of the same Shared Secret by many clients. This is a very bad idea, as it provides attackers with more data to work from and allows any flawed client to compromise several machines. All RADIUS clients that possess the same shared secret can be viewed as a single RADIUS client for the purpose of all these attacks, because no RADIUS protection is applied to the client or server address.

Most client and server implementations only allow shared secrets to be input as ASCII strings. There are only 94 different ASCII characters that can be entered from a standard US style keyboard (out of the 256 possible). Many implementations also restrict the total length of the shared secret to 16 characters or less. Both of these restrictions artificially reduce the size of the keyspace that an attacker must search in order to guess the shared secret.

4 Conclusions

4.1 Summary Findings

The RADIUS protocol has several interesting issues that arise from its design. The design and policy characteristics that seem to be principally responsible for the security problems are as follows:

- The User-Password protection technique is flawed in many ways. It should not use a stream cipher, and it should not use MD5 as a cipher primitive. (*note 3.2; attacks 3.3, 3.4, 3.5.1, 3.5.2*)

- The Response Authenticator is a good idea, but it is poorly implemented. (*attack 3.1*)
- The Access-Request packet is not authenticated at all. (*attack 3.4*)
- Many client implementations do not create Request Authenticators that are sufficiently random. (*all attacks in 3.5*)
- Many administrators choose RADIUS shared secrets with insufficient information entropy. Many client and host implementations artificially limit the shared secret key space. (*note 3.6*)

4.2 Suggested Protocol Additions

Selection of a well understood symmetric block cipher to protect the user password would be good practice. A new User-Password like attribute that uses an alternate encryption scheme should be created. I suggest TDES (as specified in ANSI X9.52) used in CBC mode. If this new attribute is used, the User-Password attribute should not be.

Ideally the block cipher would be keyed independently from the shared secret, but this may prove unworkable for compatibility reasons. Another option would be to key the cipher from some derived value of the shared secret and the request authenticator. For instance the cipher could be keyed from the output of an HMAC of the Request Authenticator (where the HMAC is keyed by the shared secret) or by seeding a cryptographic PRNG with the shared secret and the request authenticator.

Instead of using an ad hoc keyed hash primitive in the Response Authenticator, an accepted Message Authentication Code (MAC) should be used. An HMAC would be an ideal choice for this primitive. In addition, the Access-Request packet would benefit from authentication.

Though MD5 is a cryptographic hash that could be used in the HMAC primitive, it has several significant attacks against it. The RADIUS protocol would benefit from using SHA-1 instead of MD5 for HMACs.

In order to protect the Access-Request, Access-Accept and Access-Deny packets, a new attribute should be created that contains a SHA-1-HMAC of the entire RADIUS packet (with the SHA-1-HMAC attribute data set to 0). If this attribute is present, the receiving client or server should compute the HMAC for the entire RADIUS packet (with the HMAC set to zeros) and verify that the result is the same as the stored HMAC. If the result is not the same, the packet should be discarded.

When the server generates a RADIUS Access-Accept or Access-Reject packet with a SHA-1-HMAC, it should set the Response Authenticator to the associated Request Authenticator. If a client receives a RADIUS Access-Accept or Access-Reject packet that has the SHA-1-HMAC attribute, it should not test for the validity of the Response Authenticator.

When a client generates a RADIUS Access-Request packet, it should include the SHA-1-HMAC attribute. When the server receives a RADIUS Access-Request packet, it should verify the SHA-1-HMAC attribute.

There is just such an attribute defined as a RADIUS Extension in RFC 2869, called the Message-Authenticator. This attribute contains the output from an MD5 based HMAC, keyed with the shared secret, of the entire RADIUS packet. This attribute adequately protects RADIUS packets that include this attribute. Unfortunately, this attribute is not required to be consistently used (in fact, it is only required to be used when the new EAP-Message attribute is used). RFC 2869 does suggest that this attribute be used in cases where the User-Password attribute is not included in the RADIUS Access-Request packet; unfortunately, the vulnerability seen in section 3.4 requires that the User-Password attribute is in use. Further, RFC 2869 does not suggest that the server and client should have a mode where packets received without the Message-Authenticator are discarded. Without this mode, the attacker can simply strip off the Message-Authenticator attribute from a RADIUS client Access-Request packet, modify the packet and then replay the resulting packet. (It should be noted that the attacker cannot strip off this attribute from a server Access-Accept or Access Reject packet, as that message is separately authenticated by the Response Authenticator).

The Message-Authenticator attribute could provide an effective defense if it were required to be more consistently used. Clients and servers should be able to be placed in a mode where RADIUS packets without the Message-Authenticator attribute are silently discarded.

4.3 Suggested Client Behavior Modifications

Authenticator Behavior

The RADIUS specification should require a strong cryptographic PRNG for generation of the Request-Authenticator, such as the PRNG specified in ANSI X9.17 appendix C or FIPS 186-2, appendix 3.

Shared Secret Behavior

The RADIUS specification should require each RADIUS client use a different Shared Secret. It should also require the shared secret to be a random bit string at least 16 octets long that was generated by a strong cryptographic PRNG.

In order to facilitate entry of this bit string, clients and servers should allow for input of arbitrary binary data. Quite likely, the easiest solution is to allow for the entry of hexadecimal digits.

4.4 General Comments

Both servers and clients should support the base RADIUS protocol and this extended RADIUS protocol. Both the server and the client should allow the administrator to enable the use of these RADIUS extensions on a client-by-client basis. This should be an explicit configuration option, not just an automatic determination made by the server. An automatic determination made by the server could lead to an attack where the attacker attempts to force the client/server interactions into the old RADIUS mode.

If it is not possible to change the RADIUS protocol, the system can still be made much more secure by just following the suggestions in section 4.3, which can all be implemented while still remaining completely compliant with the existing RADIUS protocol.

4.5 Why Modify RADIUS?

So, why attempt to modify RADIUS at all? Why not just go to another (presumably more modern, more secure) protocol? Well, for the most part, the answer is "Because such a protocol doesn't currently exist." In the near future, however, Diameter is likely to be released by the IETF.

Diameter is the planned RADIUS replacement. The great majority of all the protocol work that has gone into Diameter has been directed to removing some of the functional limitations imposed by the RADIUS protocol. Effectively no work has been done as relates to the client/server security of the protocol. (CMS is defined, but this is a security layer for the proxy to proxy interaction, not the client to proxy/server interaction)

So, does this mean that they continue to use even RADIUS's ad hoc system? No, they removed all security functionality from the protocol. They did the protocol designer's equivalent of punting. Section 2.2 of the current Diameter protocol spec says: *"Diameter clients, such as Network Access Servers (NASes) and Foreign Agents MUST support IP Security, and MAY support TLS. Diameter servers MUST support TLS, but the administrator MAY opt to configure IPSec instead of using TLS. Operating the Diameter protocol without any security mechanism is not recommended."*

So, all security aspects of the protocol are handled by IPSec and/or TLS. From a security aspect, this strikes me as a very good idea. Both IPSec and TLS are fully featured (sometimes too fully featured) protocols that many people have reviewed. (That's already much better than RADIUS ever did).

Examining this from a slightly different angle gives me some cause for concern, however. It strikes me that the overhead imposed by a full TLS/IPSec implementation is very significant for many current-day embedded devices. This would seem to indicate that (at least in the near future) manufacturers are going to either continue to use RADIUS or ignore the Diameter standard and perform Diameter without TLS or IPSec.

Because of this, I suspect that it would be advantageous to push for at least minimal RADIUS protocol revision.

5 Document Change History

- (2001-11-13) Modified section 1.1 to remove references to CHAP, as CHAP is not sent using the User-Password attribute. (Thanks to Barney Wolff for pointing this out)
- (2001-11-13) Changed the bibliography reference from RFC 2138 to RFC 2865. The update in RFCs does not change any of the analysis in this document. Also added a reference to RFC 2869.
- (2001-11-13) Modified section 4.2 to include a small discussion about RFC 2869's Message-Authenticator attribute and associated guidance included in the RFC.
- (2001-11-13) Modified the document so that "DIAMETER" is more properly referred to as "Diameter". (Thanks to Barney Wolff for pointing this out)
- (2001-11-14) Reformatted some bibliography references and added some contact information at the beginning of the document.
- (2001-11-24) Modified the reference in section 4.4 that read "section 5.3" to correctly read "section 4.3".

6 Previous Work

There has been some independent previous work with the RADIUS protocol:

Attacks 3.5.3 and 3.5.4 are likely the attacks referred to in the RADIUS RFC.

The known password attack on the shared secret using the Access-Request packet (attack 3.3) appears to have been first observed in September, 1996 by Thomas H. Ptacek.

[Paper #1](#)

The known password attack on the shared secret using the Access-Request packet (attack 3.3), and the shared secret attack on the Access-Reject and Access-Accept packets (attack 3.1) were independently observed in July, 1997 by Reilly

(rich.friedeman@ANIXTER.COM)

[Shared Secret Recovery in RADIUS](#)

7 Acknowledgements

Thanks go to:

- Mark Smith (mark@halibut.com), who provided very useful comments regarding passwords greater than 16 bytes long.
- The Halibutians, for uncovering various grammatical and phrasing issues.
- Barney Wolff, who pointed me toward the updated RADIUS RFCs, allowing me to discuss the implications of the Message-Authenticator. Thanks also for setting me straight on a few DIA... errr... Diameter issues. :-)
- Andrew Pimlott, for pointing out some section numbering inaccuracies.

8 Bibliography

[RFC 2865](#), "Remote Authentication Dial In User Service (RADIUS)", by C. Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

[RFC 2869](#), "RADIUS Extensions", by C. Rigney, W. Willats, P. Calhoun. June 2000.

[The DIAMETER Base Protocol](#), by Pat R. Calhoun, Haseeb Akhtar, Jari Arkko, Erik Guttman, Allan C. Rubens, Glen Zorn. July 2001.

[DIAMETER CMS Security Application](#), by Pat R. Calhoun, Stephen Farrell, William Bulley. July 2001.

[FIPS 186-2](#).

[The Handbook of Applied Cryptography](#), by Alfred J Menezes, Paul C. van Oorschot, Scott A. Vanstone.

Chapter 5, chapter 6 and chapter 9. Most notably:

The MD5 based stream cipher as a synchronous stream cipher (6.1.1, ii)

The use of cryptographic functions in pseudorandom number generation is discussed in section 9.2.6.

The use of a MDC in the creation of a MAC is discussed in 9.5.2.

[An Analysis of the TACACS+ Protocol and its Implementations](#) by Solar Designer. July 2000.

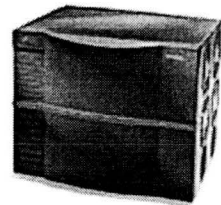
[Lucent, 2003]

MAX TNT® Universal Gateway

Description



The *MAX TNT*® Universal Gateway is a carrier-grade access platform for IP voice and data services. The *MAX TNT* Universal Gateway is a high-density product that features universal port technology to support any service on any port at any time. With uniform capacity, it delivers consistent performance and density for all applications. It can manage up to 720 concurrent calls over a dynamic mix of analog, ISDN and T1/E1 or T3 Frame Relay.



The *MAX TNT* Universal Gateway is one of the most highly-evolved platforms in the industry, combining a scalable and redundant hardware design with multiservice software and comprehensive management capabilities. It features the *True Access*® Operating System (TAOS), which enables high-density dial access applications while equipping you to expand into emerging markets such as IP Telephony. A market-proven operating system, TAOS is already utilized in Lucent's *APX*™ 8100, *APX*™ 8000, *APX*™ 1000, *MAX TNT* and *MAX*™ 6000 access platforms, with more than 35 million ports deployed globally.

For more information on our related products and solutions, please follow the links under Complementary Solutions.

Value description

The *MAX TNT* allows you to:

- Customize your network infrastructure to specific application and bandwidth requirements, lowering your costs
- Gain high-performance hardware and software capabilities in a single solution
- Maximize system capacity while reducing infrastructure and operating costs
- Reduce network equipment and transmission costs by eliminating the need for separate modem banks, terminal servers, routers, and access lines
- Eliminate high operating costs associated with analog modem technology

Features

Bandwidth on Demand

- Dial-up connections are automatically set up and torn down for transparent client-server computing across the WAN
- Dynamic bandwidth allocation aggregates multiple calls for greater bandwidth and lower costs

Connects Remote Offices and Dial-up Users with Backbone Network Services

- Provides options for connecting into a local or remote backbone network
- Transport options: Ethernet (AUI, 10Base-T, 100Base-T), Frame Relay over V.35 serial port (up to 8 Mbps/port), Unchannelized T1/E1 ports with integral CSU

Consolidation

- Eliminates the need for separate modem banks, terminal servers, routers, and access lines
- Consolidates a variety of access lines (including channelized T1/E1, channelized DS3, ISDN PRI, leased T1/E1 or Fractional T1/E1 Frame Relay, and channelized T1/E1 frame relay) over high-speed digital trunks for up to 960 simultaneous data connections

High-Speed Digital Modems

- Provides full access to analog callers that dial in over digital access lines such as channelized T1/E1, channelized DS3 or PRI
- DSP module ensures reliability and eliminates costs associated with analog modem technology

Modular Design

- Optional digital modem, hybrid access, and FrameLine modules to support up to 960 concurrent data connections to a central site, carrier, or an Internet POP
- Expansion modules along with enhanced software capabilities help build high-density, multiservice network infrastructures as well as virtual private networks

Reliability

- NEBS Level-3 compliant
- Redundant, hot-swappable, load-balancing power supplies on each shelf
- Hot-swappable modules with external indicator lamps for fault isolation
- Remote downloadable software upgrades via built-in flash RAM

Routing and Terminal Server Functions

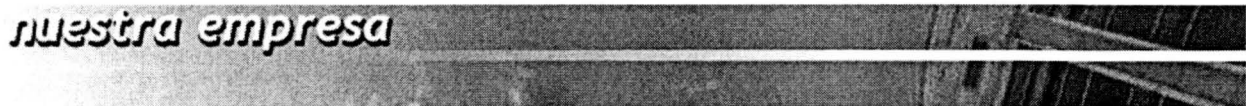
- Designed specifically for switching WAN connections
- Supports the most widely-used routing protocols for remote access
- Robust support for standard protocols including RIP, RIP2, OSPF, TCP/IP and IPX ensures efficient Internet access and connectivity for remote users within corporate intranets

Security

- Support for standard user-authentication systems fits into your current network security architecture
- Networked, service-based authentication facilitates management of large-scale remote access applications from a central site
- Supports PAP, CHAP, and MS-CHAP; Ascend Access Control (extended RADIUS), encrypted token card security, callback (digital connections), calling line ID (CLID), password protected terminal server access, transmit and receive packet filtering

[Marcatel, 2003]

nuestra empresa



Marcatel tiene como principal propósito proporcionar servicios integrales de telecomunicación a las empresas, tales como Larga Distancia Nacional e Internacional, Números 800, Tarjetas Prepagadas, Líneas Privadas Empresariales, Internet Dedicado y Dial-Up, entre otros servicios.

En Marcatel nos esforzamos en estar al día para conocer las necesidades de comunicación de las empresas y saber cuales son las tecnologías que mejor nos permitan ofrecer mejores servicios a nuestros clientes.

Debido a que nuestra red de fibra óptica está situada en sitios estratégicos podemos ofrecer las tarifas más atractivas del mercado.

En Marcatel cumplimos nuestras metas gracias al uso de alta tecnología y a un equipo humano altamente calificado así como orientado a la completa satisfacción del cliente.

Contamos con 16 Centros de Atención al Cliente en diferentes puntos de la República Mexicana, así como un Centro de Atención Telefónica y Servicio de Operadoras las 24 horas de los 365 días del año. De esta manera usted nunca estara solo.

Nuestra Misión

Ofrecer los mejores productos y servicios en telecomunicaciones contribuyendo en el proceso de diversificación tecnológica y globalización de las empresas mexicanas.

Nuestros Valores

- Entusiasmo
- Servicio al cliente
- Calidad
- Mejora continua
- Frugalidad
- Honestidad
- Innovación y creatividad
- Trabajo en equipo

Nuestra Filosofía

- Ofrecer un inigualable servicio al cliente.
- Crear sinergias entre los empleados y los clientes.
- Mantener un espíritu emprendedor.
- Adaptación al cambio ante un mundo en constante evolución

Historia

Con la apertura del mercado de las telecomunicaciones en México, Marcatel fue una de las primeras empresas en ofrecer sus servicios a los usuarios mexicanos, iniciando con el de larga distancia nacional e internacional en 1997.

En su origen, Marcatel fue el resultado de la alianza estratégica entre la empresa mexicana Radio Beep, con las empresas estadounidenses IXC y Westel. A la fecha, los accionistas son Radio Beep y Progress International, también estadounidense.

Con el transcurso de los años, Marcatel ha ido diversificando su oferta de servicios de telecomunicaciones. En adición a la telefonía tradicional de larga distancia, ofrece ahora la transmisión de datos e imagen, internet, líneas privadas, números 800, tarjetas telefónicas de postpago y prepago, entre otros.

Todos estos servicios son provistos gracias a la capacidad de la red e infraestructura propia con la que se cuenta. Construida por la empresa alemana Siemens, empresa líder en tecnología a nivel mundial, la red de Marcatel ha sido constantemente mejorada con la introducción de nuevos avances tecnológicos como el DWDM y otros.

Hoy en día, gracias a su diseño y ubicación estratégicos en la República Mexicana, podemos ofrecer beneficios de comunicación a los usuarios a través de las tarifas más atractivas del mercado.

En Marcatel cumplimos nuestras metas, gracias al uso de dicha tecnología y a un equipo humano altamente calificado y orientado a la completa satisfacción del cliente.

Nuestros Socios

Radio Beep

Radio Beep es una empresa especializada en el servicio de radiolocalización móvil de personas, desde 1973. Radio Beep es la primera empresa de paging en Latinoamérica y busca siempre desarrollar innovaciones tecnológicas en la industria. Actualmente cuenta con una cobertura nacional en 61 ciudades del país y 45 poblaciones aledañas.

Progress

Progress es una empresa incorporada bajo las leyes del estado de Texas en Estados Unidos, con socios tecnológicos e inversionistas diversos. Progress cuenta con una autorización 214, otorgada por la Federal Communications Commission (FCC) para proveer servicios de telecomunicaciones en los Estados Unidos de América.

Infraestructura Marcatel

Con el propósito de lograr una comunicación a costos competitivos en llamadas de larga distancia a Estados Unidos, Canadá y el resto del mundo, se estableció una conexión con la red de varios operadores estadounidenses, en la frontera de Reynosa, Tamaulipas y McAllen, Texas.

Se realizó una inversión de 75 millones de dólares para la construcción de una red de fibra óptica de 2500 Km. en su primera etapa, por lo cual podemos garantizar un servicio sin interrupciones.

Actualmente es posible cursar hasta 30,240 mil llamadas simultáneas con claridad excepcional por nuestra Red de Fibra Óptica.

Actualmente contamos con:

- Tres centrales nacionales de conmutación digital de tecnología alemana SIEMENS en Ciudad de México, Monterrey y Guadalajara y una central internacional
- 45 estaciones de transmisión con interconexión a las centrales de la empresa telefónica local.
- Cable de 48 Fibras Ópticas.
- Sistema de Transmisión SDH (Synchronous Digital Hierarchy).
- Capacidad de Transmisión de 40 Gigabytes en nuestro backbone.
- Anillos de protección de tráfico usando fibra y ancho de banda.
- Capacidad de 1 millón de llamadas por hora.

Para ofrecer el servicio óptimo de internet dedicado contamos con:

- Tecnología Siemens y Marconi en transmisión STM-16.
- Routers Cisco Serie 7500 en México, Monterrey y Guadalajara.
- Salida a Estados Unidos a nivel hasta STM1 (155 Mbps).
- Contamos con una ruta de respaldo de alto ancho de banda en la ciudad de México.

El servicio de Marcanet Dial-up lo proveemos gracias al equipo Lucent Serie TNT el cual está instalado en Guadalajara, Monterrey y México.

Adicionalmente a lo anterior, Nuestro Centro de Operación de la Red labora las 24 horas los 365 días del año para dar supervisión y atención a cualquier necesidad.

Cobertura

Gracias a la extensión de fibra óptica de Marcatel, ofrecemos el servicio de larga distancia a las principales entidades de la República Mexicana. Actualmente son más de 100 ciudades desde las cuales se pueden originar llamadas de larga distancia.

Ver [mapa](#) de Infraestructura Click Aquí



Actualmente el Internet es una poderosa herramienta de mercado que ofrece la posibilidad de darse a conocer, agilizar procesos y brinda diversas maneras para comercializar sus servicios.

Marcatel le ofrece una variedad de tipos de conexión dependiendo de sus necesidades de comunicación y transmisión de información.

Conecte su negocio a Internet y goce de los beneficios de estar disponible y poder ser contactado por futuros clientes o empresas interesadas en realizar alianzas estratégicas.

Marcatel le ofrece una conexión rápida y segura, debido que contamos con nuestra propia red de fibra óptica, y con al tecnología mas avanzada para darle el servicio que se merece.

Lo invitamos a conocer cada una de las opciones que le ofrecemos.

Marchante DSL

¿Le interesa contar con un servicio a internet rápido y con una accesible inversión para su negocio?

Le ofrecemos el revolucionario servicio de internet Marcanet DSL.

La tecnología de internet por DSL (Digital subscriber line) es utilizada para transmitir capacidades de ancho de banda de alta velocidad a casas y negocios pequeños.

En Marcatel ofrecemos el acceso a Internet con alta velocidad con tecnología ADSL (Asymmetric Digital Subscriber Line), es decir, la velocidad de "bajada ó download" es mayor que la velocidad de "subida o upload" hacia la red de Internet.

El servicio Marcanet DSL se integra en 2 partes:

- Acceso. Marcatel ofrece este servicio a nombre del proveedor de última milla.
- Servicio de internet. En este caso ofrecemos el equipamiento necesario para lograr la conectividad de su red a Internet.

Características:

Velocidad

Las velocidades y niveles que se puede ofrecer son:

Download	Upload
256 Kbps	128 Kbps.
512 Kbps	256 Kbps.
2048 Kbps.	512 Kbps.

Asignación de direcciones IP

La asignación de direcciones IP de Marcanet DSL se ofrece en 2 modalidades:

IP Dinámica. En este caso la dirección IP de su negocio siempre esta cambiando cuando accesa la conexión a Internet.

IP Fija. Marcatel asigna una dirección IP única a internet para una navegación más confiable.

Equipamiento

Junto con la entrega del servicio, le asignaremos un equipo ruteador para la configuración de su red LAN.

Protección de consulta en páginas web

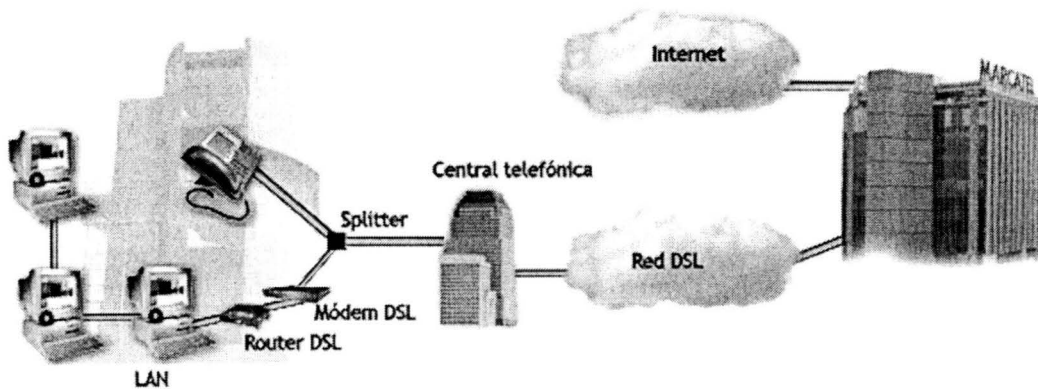
Le ofrecemos la administración de acceso a páginas web a través de tecnología implementada por Marcatel.

Modo de instalación

El servicio de Marcanet DSL se conecta a través de su línea telefónica. Un equipo técnico especializado le realiza la configuración del sitio a su red local.

DIAGRAMA

Se cuenta con una conectividad al proveedor Backbone de Internet a través de red de fibra óptica de Marcatel.

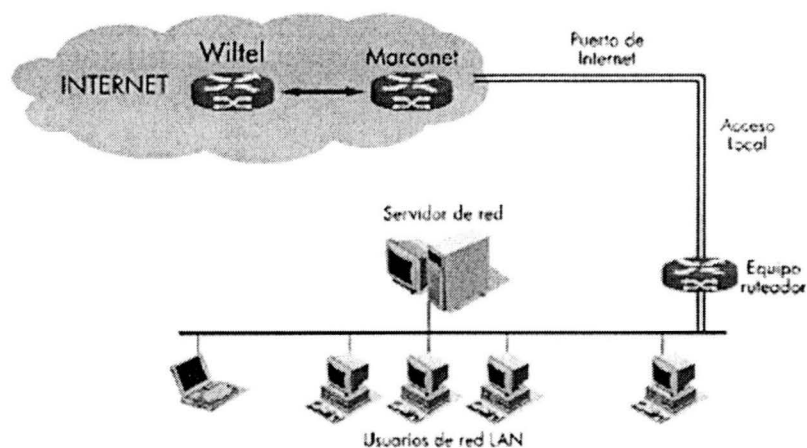


Solicite más información [aquí](#).

Marcanet Dedicado

Marcanet dedicado es el servicio de Internet con el cual su empresa podrá estar permanentemente conectada a este servicio, sin necesidad de contar con un módem y línea telefónica para su utilización. Toda conexión a internet dedicado está constituida por 3 elementos:

- Ruteador. Equipo que administra el tráfico de información entre la red local (LAN) y el enlace a Internet
- Acceso Local. Es el medio por el cual se comunica el equipo terminal del cliente con el equipo de Marcatel
- Puerto de Internet. Permite el enlace de nuestro nodo central al backbone de Internet en Estados Unidos



Servicios de Valor Agregado

Adicional al servicio de acceso a Internet de Marcanet Dedicado le ofrecemos los siguientes servicios de valor

agregado:

- Direcciones IP. Asignación de direcciones IP públicas fijas para su uso con diversas aplicaciones
- E-mail. Acceso a servicios de Correo Electrónico proporcionados directamente por Marcanet
- Web Hosting. Disponibilidad de espacio en nuestro servidor de páginas Web para hospedaje de su sitio

Solicite más información [aquí](#).

Marcanet Dial-up

A través de Marcanet Dial Up, podrá estar en comunicación continua con clientes y proveedores donde quiera que este y bajo el esquema de contratación que usted elija.

Instalación

Con una sencilla instalación, usted puede definir el nombre de usuario y password para que tenga acceso al servicio de internet.

Correo Electrónico

Además disfrute de 2 cuentas de correo electrónico de 10 Mb. cada una y tenga acceso a ellas a través cualquier programa de correo electrónico.

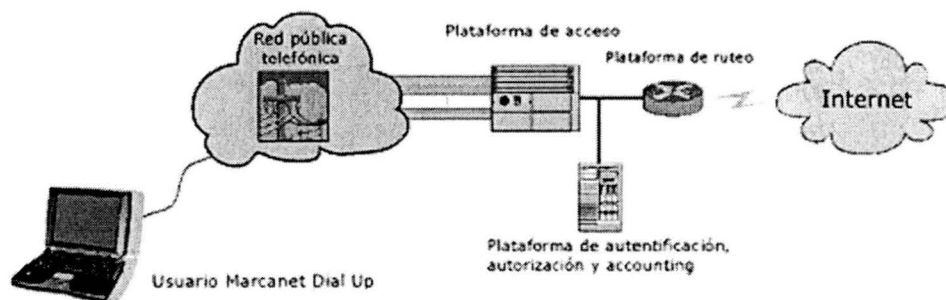
Espacio Virtual

Usted puede tener una página en internet pagando el mismo costo por su servicio dial-up ya que al convertirse en cliente Marcanet Dial Up podrá tener un sitio en internet con un espacio de 5 Mb así como un dominio virtual gratis.

¿Cómo se realiza el acceso a internet?

Se da acceso a internet a través de una computadora que cuente con un módem, del sistema operativo Windows o Macintosh y una línea telefónica que se enlaza a las instalaciones de Marcatel tal como se muestra a continuación:

Esquema de conexión



Marcanet Burstable

Por fin en México el servicio que revoluciona la manera de ofrecer internet de una manera justa a su empresa aunque utilice su servicio de manera variable.

Le ofrecemos Marcanet Burstable, el cual es el servicio de internet dedicado cobrado en base a la demanda de uso que en su empresa le de.

¿Cómo funciona?

- Se cobra el tráfico en Megabytes desde y hacia las instalaciones de su empresa.
- Otra herramienta en línea que es el MRTG
- Se entrega el ancho de banda y se cobra una cuota preestablecida.
- Los servicios de web y mail con los que puede contar
- Las direcciones de IP fijas que se otorgan sin cargo alguno

¿Por qué debería aprovecharlo?

Es un servicio único en el mercado. Somos los primeros en pensar de qué manera podríamos ofrecer el servicio más justo por el pago de su servicio de internet.

Utilizamos un sistema de cobro innovador ya que ningún proveedor cobra servicios por uso de paquetes.

Usted no tendrá de qué preocuparse al tener "topado" el ancho de banda recibido.

Solicite informes [aquí](#) acerca del cobro justo por su servicio Marcanet Burstable.

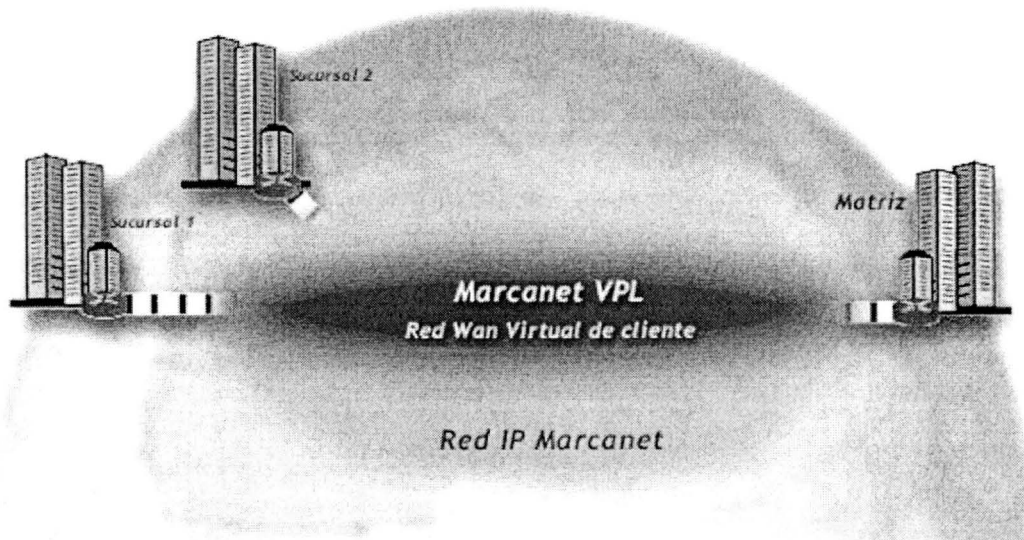
Marcanet VPL

¿En su empresa necesita enviar y recibir datos a otra o más sucursales de manera confiable, segura y económica?

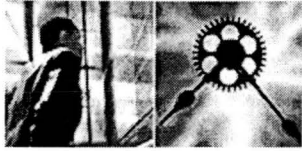
La Red Privada Virtual de Marcanet le permite realizar la transmisión de datos de manera privada a través nuestra red IP.

Elija el ancho de banda que su empresa requiera y olvídense de considerar el volumen de información transmitida.

Construya una red virtual con los nodos de sus sitios remotos tal como le mostramos a continuación un esquema:



Metodología Microsoft Solution Framework



A continuación se describen los pasos que debería cumplir un proyecto según la metodología de desarrollo de proyectos de Microsoft (Microsoft Solution Framework o MSF)

Ejemplo de metodología MSF aplicada

- [Fase 1: Estrategia y alcance](#)
- [Fase 2: Planificación y Prueba de Concepto](#)
- [Fase 3: Estabilización](#)
- [Fase 4: Despliegue](#)

Como ejemplo de una aplicación de metodología MSF a un proyecto, a continuación se describe el contenido de cada una de las fases y, en la medida de lo posible, un detalle de acciones concretas y estimación de carga de trabajo en términos de jornadas, número de personas implicadas y perfil de las mismas. El proyecto ejemplo se trata de una implantación de infraestructuras, en concreto, migración a Windows 2000 de una red de servidores.

Metodología Microsoft Solution Framework: Ejemplo

■ Fase 1 - Estrategia y alcance

En esta fase deberían tener lugar los siguientes trabajos:

- Elaboración y aprobación del Documento de Alcance y Estrategia definitivo: debe ser un documento de consenso con la participación del mayor número de agentes implicados en el proyecto. En este documento quedarán definitivamente reflejadas las funcionalidades y servicios que, ineludiblemente, debe ofrecer la solución a implantar.
- Formación del Equipo de Trabajo y distribución de competencias y responsabilidades: generalmente se definen como áreas principales la de Diseño de Arquitectura, Pruebas de Laboratorio, Documentación, Logística y Coordinación.
- Elaboración del Plan de Trabajo: deben marcarse fechas y contenidos para esta fase y las siguientes. Los mecanismos y protocolos de intercambio de información y coordinación deben quedar suficientemente bien establecidos y consensuados.
- Elaboración de la matriz de Riesgos y Plan de Contingencia: los principales riesgos detectados deben tener un plan de mitigación y actuación y revisarse con periodicidad.

Para un proyecto de migración a Windows 2000 podría estimarse que los trabajos indicados pueden requerir en torno a 20 jornadas de trabajo y la intervención de un Consultor de Microsoft junto con el equipo de trabajo que formen El cliente y el Partner.

■ Fase 2 - Planificación y Prueba de Concepto

Deben alcanzarse los siguientes objetivos e hitos:

- Documento de Planificación y Diseño de Arquitectura: es el documento principal, donde se describen en detalle los aspectos funcionales y operativos de la nueva plataforma. La aprobación de este documento es el hito principal de esta fase, y supone la directriz última de todos los trabajos técnicos, que, a partir de ese momento, deben ser consistentes con esta Guía. Si en el curso de las fases sucesivas fuera necesario revisar estos contenidos, se deberá hacer por acuerdo y conocimiento de todo el equipo de trabajo y se llevará un registro de versiones que permita hacer un seguimiento adecuado de estas revisiones.
- Documento de Plan de Laboratorio - Prueba de Concepto: la descripción del contenido del laboratorio de prueba de concepto, los diversos escenarios a simular, los criterios de validez, el control de incidencias y las métricas de calidad son objetivos a cubrir en este documento. Es un documento dinámico, en el que se recoge la idea y la experiencia práctica al llevarla a cabo en entorno controlado y aislado. La etapa de prueba de laboratorio concluye cuando la maqueta ofrece todos los servicios y funciones descritos en el Documento de Alcance y Estrategia, y su grado de estabilidad y rendimiento es considerado como

"suficiente".

Habitualmente, en las propuestas de preventa no se pueden indicar con precisión parámetros como el esfuerzo técnico, tiempo o coste definitivo que puede suponer esta fase. De otras experiencias anteriores se puede obtener una relación de trabajos sólo a título orientativo, y que debe ser revisado y acordado por todas las partes:

Fase	Jornadas	Personas (mínimo)	Contenido	Documento
Servicios Básicos de Red, diseño de sites y Active Directory	10	El Partner: 2 El cliente: 1	Instalación de la plataforma Windows 2000 Definición de parámetros y puesta en servicio de SBR: Active Directory, DHCP y DDNS	Documento de instalación de Windows 2000 y Servicios Básicos de Red
Seguridad y Accesos	10	El Partner: 2 El cliente: 2	Definición de grupos de usuarios. Nomenclatura Modelo de seguridad de acceso. Pruebas con MSCHAP, Kerberos y Smart Card (si está disponible).	Documento de definición del método de validación y seguridad lógica.
Administración	15	El Partner: 2 El cliente: 1	Definición de las tareas administrativas relevantes: mantenimiento de usuarios y grupos, copia de seguridad, uso de auditoría, recuperación de máquinas, reinstalación de software, etc Definición de perfiles y políticas de usuario y grupo	Documento guía de administración.Documento Plan de Contingencias (Disaster & Recovery Plan)
Servicios y aplicaciones corporativas	10	El Partner: 2 El cliente: 2	Instalación de servicios y aplicaciones corporativas en la nueva plataforma y procedimientos para conseguirlo: Configuración de recursos compartidos (File & Print) Distribución de Software	Documento de migración e implantación de aplicaciones y servicios corporativos (estrategia e implantacion)
Alta Disponibilidad y tolerancia a fallos	10	El Partner: 2 El cliente: 1	Escenarios de operación 24x7 y balanceo de carga con Windows 2000 AS.	Documento de pruebas de Alta Disponibilidad
Conectividad y modalidades de acceso remoto	5	El Partner: 2 El cliente: 1	Configuración de la DMZ, firewalls, proxys y medios físicos de acceso	Documento de configuración de la Infraestructura de Acceso Remoto
Windows Terminal Server	5	El Partner: 2 El cliente: 2	Instalación y configuración de Terminal Server. Configuración del perfil de acceso administrativo y aplicaciones	Documento de Instalación, configuración y escenarios de uso de W2000TS.
Servicios web y provisión de contenidos por la red	5	El Partner: 2 El cliente: 2	Instalación y configuración de IIS5.0 para provisión de servicios on-line de Intranet (formación, documentación, etc)	Documento de implantación y servicios de Intranet
Escalabilidad	5	El Partner: 2 El cliente: 2	Definición de los escenarios de crecimiento de la nueva arquitectura descrita y requerimientos de infraestructura, administración y seguridad	Documento de pruebas de escalabilidad

El cómputo de jornadas para la relación de actividades descritas (que como se observa sólo recogen las relativas a la Planificación y Diseño, y deja aparte las necesarias para elaborar el plan de Migración), ofrecería este resultado:

Jornadas totales: 80 (aprox. 4 meses)
 Jornadas / técnico del PARTNER: 150 jornadas (2 personas)
 Jornadas / técnico del CLIENTE: 110 jornadas (max. 2 personas)

■ Fase 3 - Estabilización

La solución implantada en la maqueta se pasa a un entorno real de explotación, restringido en número de usuarios y en condiciones tales que se pueda llevar un control efectivo de la situación. Los hitos y objetivos fundamentales de esta fase son:

- **Selección del entorno de prueba piloto:** se acordará la composición y ubicación del conjunto de máquinas y usuarios que entrarán en la prueba. Esta selección se recomienda que se haga atendiendo a la mayor variedad posible de casos, de manera que puedan aflorar el máximo de incidentes potenciales en el menor tiempo posible. La dimensión de la muestra tiene también que calcularse, sin perder de vista que la prueba piloto no es el despliegue propiamente, sino una fase de observación en la que es absolutamente crítico establecer unos cauces efectivos de tratamiento de los errores.
- **Gestión de Incidencias:** aunque esta labor se habrá iniciado en la fase anterior, el éxito de la prueba piloto dependerá de que se forme un sistema de recogida de incidentes (helpdesk o similar), de atención al usuario (formación, consultas) y de resolución de problemas y documentación de los mismos (versionado de la plataforma).
- **Revisión de la documentación final de Arquitectura:** el documento de Planificación y Diseño de Arquitectura se puede ver alterado parcialmente como resultado de esta fase. El documento final, aprobado por consenso, supone el principal documento del Proyecto y la culminación de los trabajos de diseño, al menos en sus líneas principales. Este documento se considerará definitivo cuando la solución puesta en marcha se muestre estable y el número de incidencias graves (de intervención o de resolución) sea nulo y la cantidad de las consideradas leves quede por debajo de un límite establecido en las Métricas de Calidad.
- **Elaboración de la documentación de Formación y Operaciones:** con vistas al soporte post proyecto y los programas de formación a usuarios y administradores, en esta fase deben elaborarse las Guías de Usuario, de Administración, las "paso-a-paso", y otros cuyos contenidos deben acordarse previamente.
- **Elaboración del Plan de Despliegue:** se debe consensuar la fecha de finalización de la fase Piloto, y las condiciones de calidad que debe cumplir la solución final para iniciar el despliegue. En el Plan deben identificarse las fases, estrategia de implantación, fechas, tareas a realizar, procedimientos de validación y método de control de incidencias.
- **Elaboración del Plan de Formación:** con anterioridad al despliegue definitivo, debe haberse aprobado el Plan de Formación orientado a usuarios finales y administradores, y debe hacerse compatible con los ritmos acordados en el Plan de Despliegue.

El tiempo necesario para abordar esta fase es variable y depende en parte de factores ajenos a la complejidad de la propia solución, como es la adecuada selección del entorno de prueba y el momento del año en que tenga lugar (evitando que coincida con periodos de vacaciones o puntas de trabajo críticas como Fin de Año). En proyectos de similar envergadura se ha llegado al momento "Error Free Version" en 30 jornadas de trabajo (aproximadamente mes y medio) con una muestra de 50 usuarios.

■ Fase 4 - Despliegue

Se llevarán a cabo en esta fase los planes diseñados en la anterior, principalmente el de despliegue y el de formación. Los principales trabajos e hitos a conseguir son, en este caso, además de los obvios (implantación de la plataforma, puesta en servicio de todas las funciones, formación a los usuarios y administradores), los siguientes:

- Continuación con las labores de recepción de incidencias, clasificación, tratamiento, resolución y distribución de fixes o intervención on-site.
- Registro de mejoras y sugerencias, funcionalidades no cubiertas y novedades a incorporar en sucesivas versiones de la plataforma, incluyendo mejoras aportadas por los fabricantes de software (nuevas versiones o Service Packs, por ejemplo)
- Revisión de las Guías y manuales de usuario, rectificación de errores y obtención de los documentos de formación definitivos.
- Entrega de los documentos definitivos acordados como "deliverables" en la fase de Vision Scope.
- Revisión (si procede) de la matriz de riesgos, las métricas de calidad y establecimiento de los estándares de calidad y SLA definitivos.
- Finalmente, entrega del Proyecto y cierre del mismo, con o sin apertura de nuevo proyecto en base a la información y experiencia obtenidos.

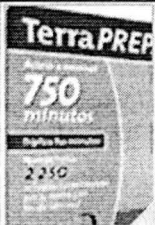
La duración fase de despliegue, puesto que debe planificarse, no puede establecerse a priori. Depende de numerosos

factores externos al propio proyecto (incluyendo factores de oportunidad política o de negocio) que pueden retardar o acelerar la conclusión.

La experiencia demuestra que no hay una relación directa entre número de máquinas y tiempo necesario para el despliegue. Los factores más relevantes en el cálculo suelen ser la dispersión o concentración geográfica, la complejidad del proceso de migración, el grado de automatización alcanzado, la experiencia y nivel de los técnicos que realizan la operación y condicionantes de calendario, a menudo con restricciones no técnicas, sino de otros tipos (las fechas-objetivo suelen marcarse por criterios de oportunidad de negocio).

[Terra, 2003]

Terra Prepago



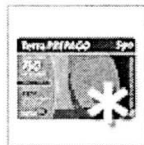
¡Olvídate de pagos fijos mensuales!
 Compra **Terra PREPAGO**, tu tarjeta de Internet
 Puedes comprar tarjetas de **750, 1500 o 4000** minutos de acceso a Internet y tendrás hasta **3 meses** para terminártelos.

COMPRA EN LÍNEA AQUÍ

¡Tú decides cuánto tiempo comprar y cuándo gastártelo!



COMPRA
 COMPRA EN LÍNEA tus minutos de conexión. ...



ACTIVA
 ACTIVA UNA TARJETA que hayas comprado. ...



CONSULTA
 CONSULTA EL SALDO de tu cuenta de Terra Prepago. ...

TIPS

POLÍTICAS ADMINISTRATIVAS

Políticas Administrativas Vigentes del acceso a Internet Terra Prepago.

SOFTWARE DE CONEXIÓN

Software de Conexión a Internet de Terra Prepago.

BENEFICIOS

Conoce los beneficios de Terra Prepago.

DISTRIBUIDORES

¡Compra las tarjetas con los Distribuidores Terra

BENEFICIOS

Con el acceso a Internet Terra Prepago navegas los minutos de conexión que tu quieras y disfrutas de:

3x1 El triple de tus minutos de conexión. Obtén 3x1 en minutos de conexión al navegar en Internet en los horarios de Lunes a Viernes de 9 p.m. a 7 a.m. y Sábados y Domingos todo el día. Por ejemplo, si navegas en este horario 15 minutos, se te descontarán solamente 5.

 **Buzón con 6 MB de espacio** Donde podrás enviar y recibir tus correos electrónicos. Consulta tu mail en web a través del **Terra Mail**

 **Rápida Conexión.** Viajarás a la máxima velocidad que permita tu equipo.



Disco Duro Virtual. Obtén tu espacio (10MB) en la red, donde podrás guardar y compartir archivos de todo tipo.



Soporte en línea. A través de la Ayuda Interactiva o del correo electrónico nuestros ejecutivos te ayudarán a resolver todas tus dudas.



Canales y Servicios para ti en nuestro Portal. Disfruta de más de 17 canales de información además de otros servicios como Chat, Mail en web, InstanTerra (mensajero instantáneo), Empleos, Clasificados, entre otros.

Planes y Tarifas de Terra Prepago

Con el acceso a Internet de **Terra Prepago** tu compras **en línea** o con **Distribuidores Terra** los minutos de conexión que necesites.

Elige la cantidad de minutos que mas te convenga:

Minutos de Conexión en horario normal	750 minutos	1500 minutos	4000 minutos
Minutos de Conexión en horario especial	2,250 minutos	4,500 minutos	12,000 minutos
Precio	\$90	\$170	\$290

[Cisco ACS Radius, 2003]

White Paper

Cisco IOS Technologies: RADIUS Support in Cisco IOS Software

Overview

As networks grow beyond the campus, network security increases in importance and administration complexity. Customers need to protect networks and network resources from unauthorized access by remote users. Cisco Systems uses a strategy known as authentication, authorization, and accounting (AAA) for verifying the identity of, granting access to, and tracking the actions of remote users. In today's networks, the Terminal Access Controller Access Control System plus (TACACS+) and Remote Access Dial-In User Service (RADIUS) protocols are commonly used to provide AAA solutions. Support for RADIUS along with TACACS+ enables Cisco to deliver tremendous flexibility and choice to organizations in AAA functionality.

Cisco Systems introduced support for RADIUS in Cisco IOS Software Release 11.1, and Cisco leads the industry in supporting both RADIUS and TACACS+ in its network access server (NAS) devices.

This white paper provides an overview of the RADIUS protocol and Cisco IOS software support for RADIUS and examines key differences between the RADIUS and TACACS+ protocols. In addition, other sources of information on these topics are identified.

The RADIUS Protocol

The RADIUS protocol was developed by Livingston Enterprises, Inc., as an access server authentication and accounting protocol. Implemented by several vendors of network access servers, RADIUS has gained support among a wide customer base, including Internet service providers (ISPs). The RADIUS authentication protocol is documented separately from the accounting protocol, but the two can be used together for a comprehensive solution.

The RADIUS IETF Standard RFCs

In June 1996, draft 5 of the RADIUS protocol specification was submitted to the Internet Engineering Task Force (IETF). The RADIUS specification (RFC 2058) and RADIUS accounting standard (RFC 2059) are now proposed standard protocols. The text of the IETF proposed standards can be found at the following URLs:

```
ftp://ds.internic.net/rfc/rfc2058.txt
ftp://ds.internic.net/rfc/rfc2059.txt
```

Client/Server Model

The RADIUS protocol is based on a client/server model. A NAS such as a Cisco AS5200 access server operates as a client of RADIUS. The client is responsible for passing user information to a designated RADIUS server and then acting on the response that is returned.

A RADIUS server (or daemon) can provide authentication and accounting services to one or more client NAS devices. RADIUS servers are responsible for receiving user connection requests, authenticating users, and then returning all configuration information necessary for the client to deliver service to the users. A RADIUS access server is generally a dedicated workstation connected to the network.

RADIUS Servers

RADIUS server software is available from several sources. Livingston Enterprises, Inc., and Merit (University of Michigan) offer RADIUS server C source code without use restrictions.

The Merit and Livingston distributions are available from their respective URLs:

```
ftp://RADIUS/releases/RADIUS.2.4.16.tar.{z,gz}
ftp://ftp.livingston.com/pub/livingston/RADIUS/
```

In addition to Merit and Livingston, other vendors have developed or adapted RADIUS server software to support their communications servers, and the software is usually available directly from the vendors. However, in most cases, the RADIUS servers distributed by these companies are not considered supported products, but are provided as a convenience to the user.

CiscoSecure

CiscoSecure software is a UNIX application program that provides AAA for dialup networks. It is the server portion of a distributed security solution using the TACACS+ protocol. Cisco will add RADIUS to CiscoSecure software in the near future. In addition to supporting all IETF attributes, CiscoSecure software will also support Cisco's RADIUS extensions and proprietary RADIUS extensions such as those used by Ascend. Users can also define their own extensions, creating a new RADIUS dictionary. A single database will be used to support both TACACS+ and RADIUS users, and all CiscoSecure V.2.0 features, such as token card support and the hypertext markup language (HTML)-based Web interface, will be supported in RADIUS.

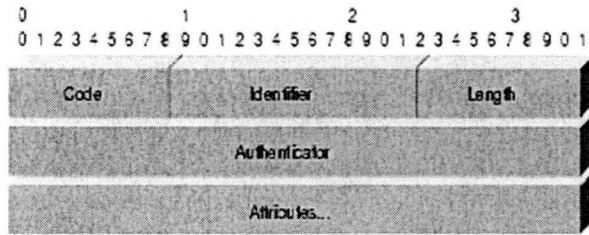
Protocol Operation

Communication between a NAS and a RADIUS server is based on the User Datagram Protocol (UDP). Figure 1 shows the RADIUS packet format.

The authors of the RADIUS protocol selected UDP as the transport protocol for technical reasons. Generally, the RADIUS protocol is considered to be a connectionless service. Issues related to server availability, retransmission, and timeouts are handled by the RADIUS-enabled devices rather than the transmission protocol.

Figure 1 RADIUS Packet Format from RFC 2058

A summary of the RADIUS data format is shown below.
The fields are transmitted from left to right.



Code

The Code field is one octet, and identifies the type of RADIUS packet. When a packet is received with an invalid Code field, it is silently discarded. Radius Codes (decimal) are assigned as follows:

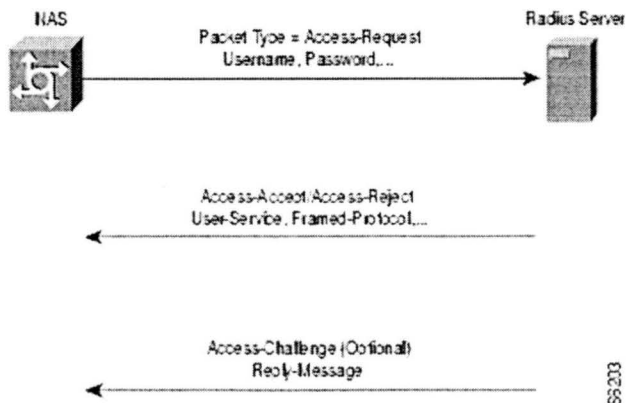
- 1 Access-Request
- 2 Access-Accept
- 3 Access-Reject
- 4 Accounting-Request
- 5 Accounting-Response
- 11 Access-Challenge
- 12 Status-Server (experimental)
- 13 Status-Client (experimental)
- 255 Reserved

Typically, a user login consists of a query (Access-Request) from the NAS to the RADIUS server and a corresponding response (Access-Accept or Access-Reject) from the server. The Access-Request packet contains the username, encrypted password, NAS IP address, and port. The format of the request also provides information on the type of session that the user wants to initiate. For example, if the query is presented in character mode, the inference is "Service-Type = Exec-User," but if the request is presented in Point-to-Point Protocol (PPP) packet mode, the inference is "Service-Type = Framed-User" and "Framed-Type = PPP."

When the RADIUS server receives the Access-Request from the NAS, it searches a database for the username listed. If the username does not exist in the database, either a default profile is loaded or the RADIUS server immediately sends an Access-Reject message. This Access-Reject message can be accompanied by an optional text message, which could indicate the reason for the refusal.

If the username is found and the password is correct, the RADIUS server returns an Access-Accept response, including a list of attribute-value pairs that describe the parameters to be used for this session. Typical parameters include service type (shell or framed), protocol type, IP address to assign the user (static or dynamic), access list to apply, or a static route to install in the NAS routing table. The configuration information in the RADIUS server defines what will be installed on the NAS. Figure 2 illustrates the RADIUS login and authentication process.

Figure 2 RADIUS Login and Authentication Process



Authentication and Authorization Features

Authentication is the most troublesome aspect of remote security because of the difficulty associated with positively identifying a user. To ensure the identity of a remote user, the RADIUS protocol supports several methods of authentication, including Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and token cards. At present, all implementations of RADIUS require that the server of a token card vendor is run in addition to the RADIUS server. When CiscoSecure RADIUS support is released, it will include OEM support for CryptoCard token cards, and an additional server will not be necessary for these tokens.

Enabling RADIUS Authentication

For every type of login that requires authentication, a command line must be entered. This line is the default list, which is used for login via RADIUS unless another list is configured:

```
aaa authentication login default radius local
```

Different login styles can be used, but each requires the creation of a separate method list.

For example:

```
aaa authentication ppp RADIUS tacacs+ local
aaa authentication login line RADIUS
aaa authentication login telnet RADIUS tacacs+ line
aaa authentication login physical RADIUS enable
BRIO
encapsulation ppp
ppp authentication chap ppp
line con 0
login authentication physical
line 1 8
login authentication line
modem InOut
flowcontrol hardware
line vty 0 4
password cisco
login authentication telnet
```

Enabling RADIUS Authorization

RADIUS authorization is enabled with the AAA authorization command. The following command enables RADIUS authentication on exec sessions to the router:

```
aaa authorization exec radius
```

The same steps can be taken for PPP or Serial Line Internet Protocol (SLIP) access. The following command line is also required if RADIUS is to be used to help assign an IP address:

```
aaa authorization network radius
```

The following command authorizes commands to be used at different privilege levels:

```
aaa authorization command <level> radius
```

Of course, the RADIUS access control server must be configured with the appropriate permit or deny criteria. Customization of privilege levels on each NAS may also be appropriate.

Accounting Features

The accounting features of the RADIUS protocol can be used independently of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of sessions, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet Service Provider (ISP) might use RADIUS access control and accounting software to meet special security and billing needs.

Enabling RADIUS Accounting

The **aaa accounting** command with the **radius** keyword are used to turn on RADIUS accounting for each Cisco IOS privilege level and network service. The accounting feature can be used for a variety of different actions that users might want to keep track of, including:

```
aaa accounting exec start-stop radius
aaa accounting network start-stop radius
```

These commands provide the starting and stopping time from a network and an exec session on the NAS.

RADIUS accounting can also be used to keep track of command levels accessed per user with the following:

```
aaa accounting command 5 stop-only radius
aaa accounting command 10 stop-only radius
```

This scenario provides accounting records for users who utilize commands set at level 5 and level 10 to the RADIUS daemon.

RADIUS Support in Cisco IOS Software

Supported RADIUS Attributes

Cisco IOS software supports many of the RADIUS attributes specified in RFC 2058. Table 1 lists the RADIUS attributes supported in Cisco IOS software. These attributes are described in more detail in the "Configuring Network Access Security" section of the Cisco IOS Software Release Notes.

Interoperability

One of the goals of the RADIUS standard is to enable interoperability between the products of different vendors. There are, however, some caveats.

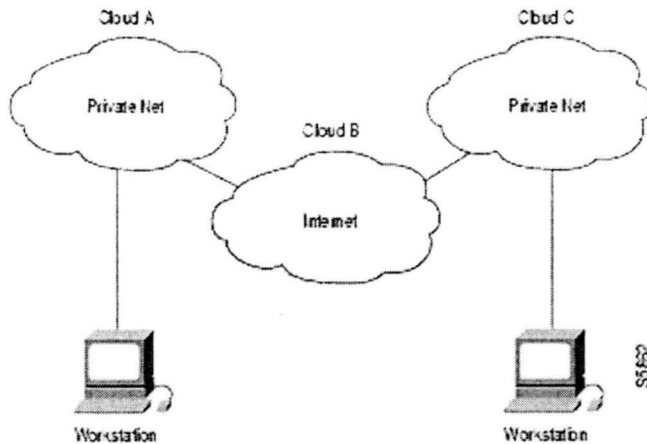
Although several vendors implement RADIUS clients, the products from all vendors are not always interoperable. Approximately 45 attributes are defined in the IETF RADIUS draft standard. Currently, Cisco's NAS products support more than 30 IETF variables through Cisco IOS software. Cisco will continue to expand support for these variables in future software releases. (Generally, as long as customers use products that support the standard RADIUS attributes in their networks, these products will interoperate.) The RADIUS implementation in Cisco IOS software has been extensively tested to ensure interoperability with other RADIUS clients and access control servers that conform to the IETF RADIUS draft standard.

Unfortunately, some vendors, rather than following the RADIUS specification, have implemented proprietary extensions, for example, extended attributes, to the RADIUS protocol. If customers use these proprietary vendor-specific extended attributes, interoperability might be sacrificed.

RADIUS and L2F

Layer 2 Forwarding (L2F) is a generic tunneling technique that encapsulates PPP frames so that devices in Cloud A believe that they have direct PPP connections with devices in Cloud B. (See Figure 3.)

Figure 3 Connecting Private Networks across an Untrusted Link

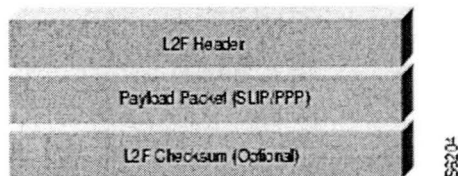


Advantages of encapsulating PPP frames include:

- L2F is insensitive to the protocol carried within the PPP frames and, therefore, can support multiple protocols
- The end systems communicating via the L2F tunnel can employ the same authentication procedures as if they were directly connected and, therefore, can be assured with a high level of confidence that the end systems have been authenticated

The specific details of the L2F protocol can be obtained from the draft RFC. In summary, L2F describes a series of messages that are exchanged between the tunnel endpoints, a format for the exchange of messages, and user data. Figure 4 shows the structure on an L2F packet.

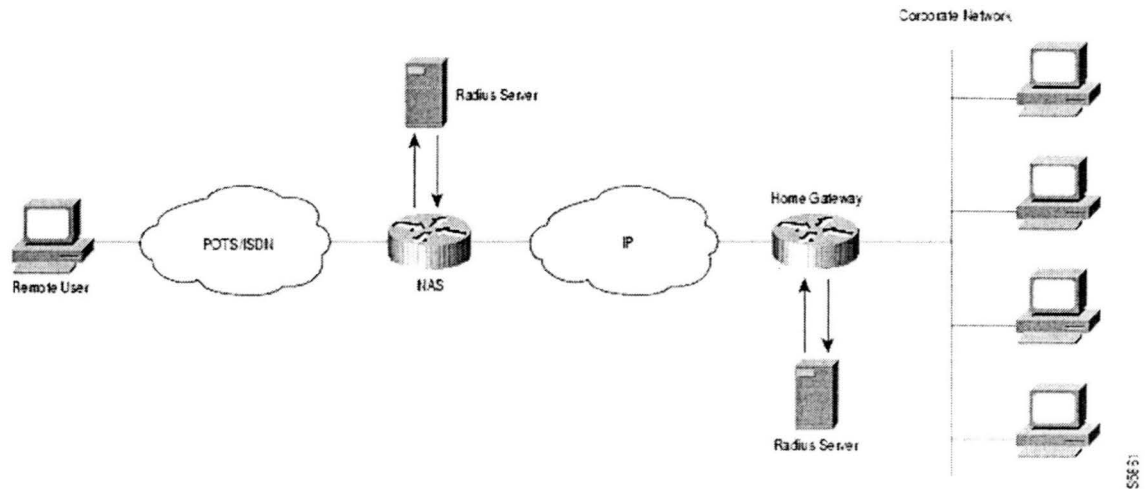
Figure 4 Structure of an L2F Packet



RADIUS and L2F Interaction

In the vernacular used to describe L2F, the remote user (client or router) dials into a NAS. The NAS establishes a tunnel (or uses an existing tunnel) to communicate with the home gateway. For example, the home gateway is the router located at the corporate headquarters. The remote user now effectively has a PPP session with the home gateway as shown in Figure 5.

Figure 5 Client, NAS, Cloud, and Home Gateway



RADIUS and L2F overlap because:

- The home gateway must authenticate the remote user by applying the techniques provided by PPP, namely PAP or CHAP.
- The NAS must identify the appropriate tunnel for the particular remote user.
- The NAS and the home gateway (the two endpoints of the tunnel) must authenticate each other.

As described previously, RADIUS is a client/server protocol that is used to obtain information that in turn can be used to authenticate (and authorize) a user wanting to gain access to network resources. In the L2F scenario described herein, the home gateway can query a RADIUS server (located deeper within the corporate network, for example) to determine whether user = user@bigcompany can be authenticated (via CHAP or PAP) and can be authorized to start a PPP session with the appropriate Network Control Programs (NCPs) (for example, an IP control point [CP] and an IPX CP). This scenario describes the normal use of the RADIUS protocol. In summary, the home gateway captures the remote user's name (user@bigcompany) and, in the case of CHAP, the response (from the remote user) to the CHAP challenge. This information (using an Access-Request) is sent to the RADIUS server. The server responds with an Access-Reject or an Access-Accept, along with a series of attribute/value pairs that allow the NAS to determine which services are authorized for that user.

To identify the appropriate tunnel for the remote user, the NAS queries the RADIUS server with the "name user = bigcompany" query. (In other words, the NAS strips off "user" before passing the username to the RADIUS server.) This RADIUS server is typically located in a telco or ISP network and is not the same as the RADIUS server within the corporate network. The RADIUS server uses the vendor-specific attribute (attribute 26) to convey the authorization to start a tunnel as well as the information describing the endpoint of the tunnel (cisco-avpair="vpdn:ip addresses=a.b.c.d" in RADIUS-speak), the name to be used when authenticating the tunnel (cisco-avpair="vpdn:tunnel-id=nas-name"), and information to authenticate the tunnel (cisco-avpair="vpdn:nas-password =mumble" and cisco-avpair="vpdn:gw-password =grumble"). Attribute 26 is used so that the L2F information can be communicated to the NAS in a syntax that a Cisco router will understand. User = bigcompany is used rather than user = user@bigcompany so that all users associated with bigcompany can establish tunnels with the home gateway without the telco's (or ISP's) RADIUS server needing distinct entries for every user at bigcompany.

To support the NAS and home gateway mutual authentication, the NAS uses the information it received when it queried the RADIUS server with user = bigcompany to authenticate its side of the tunnel. The tunnel is authenticated using the same procedures that are used when using CHAP with PPP. In other words, the NAS must issue a challenge to the home gateway and verify the response that the home gateway sends. This scenario represents normal use of RADIUS and does not pose any particular challenge for RADIUS. In addition, the NAS must respond correctly to the challenge issued by the home gateway. This part poses a severe problem when using RADIUS because the NAS needs access to the clear text version of the password associated with the home gateway. Because RADIUS packets are not encrypted, the RADIUS protocol is not well suited as a means to pass clear text passwords around. However, the use of the Cisco-specific attributes (vpdn:nas-password and vpdn:gw-password) provides a workaround for the NAS.

The home gateway faces the same problem as the NAS when authenticating the tunnel; that is, the home gateway must formulate a challenge and check the validity of the responses, as well as respond to the challenge issued by the NAS. As discussed previously, the first part can be supported by the RADIUS protocol, and the second part is problematical. In current Cisco IOS implementations, the home gateway cannot use RADIUS to authenticate the tunnel. The home gateway must use TACACS+ or

have access to the information locally (that is, manually configured on the home gateway). Once the tunnel has been established, the home gateway can use RADIUS to authenticate the remote user (user@bigcompany).

Support for Token Cards

Each token card vendor has developed its own method for interacting with RADIUS. Typically this scenario has taken the form of specifically modified versions of RADIUS that call the token card server when they encounter authentications with special keywords in the password field. For instance, Security Dynamics and Ascend created a version of the RADIUS server that recognizes a special password, "securid." If the server encounters a user with that password, it passes the authentication information off to the Security Dynamics server, and the response from that server back to the NAS¹. Cisco can now interoperate with this and several other modified RADIUS servers to provide token card support. When CiscoSecure 2.0 RADIUS server support is released, it will interoperate with servers from Security Dynamics and Enigma Logics, as well as provide onboard support for CryptoCard, which Cisco will OEM.

Future RADIUS Enhancements

Cisco continues to enhance support for RADIUS in Cisco IOS software. Some planned enhancements to the RADIUS support in Cisco IOS software include support for additional IETF standard attributes, as well as support for those vendor proprietary attributes that make sense in the context of Cisco NAS implementations. As these features are added to Cisco IOS software, information on their configuration and interoperability will be made available in the appropriate software configuration guides.

Also, as previously noted, Cisco will soon release an enhanced version of the CiscoSecure access control server that will support RADIUS. These two components will work together to offer a comprehensive solution to the problems presented by increasingly important and complex remote access challenges.

Comparisons of RADIUS and TACACS+

RADIUS is just one protocol that provides authentication and accounting functions for dialin services. Another widely deployed protocol is TACACS+, which is the latest evolution of the Cisco TACACS protocol. Although these protocols provide similar functionality, they have several key differences.

Transport Mechanism

The most fundamental difference between TACACS+ and RADIUS is the network transport protocol that each uses. The RADIUS protocol uses UDP to exchange information between the NAS and the access control server, whereas TACACS+ uses Transmission Control Protocol (TCP).

TCP is a connection-oriented transport protocol, whereas UDP offers best-effort delivery. RADIUS utilizes additional, programmable variables to control issues such as retransmit attempts and timeouts in order to compensate for the best-effort transport provided by UDP. TACACS+, using TCP transport, does not need these extra variables, because connection issues are handled transparently by the TCP protocol.

Using TCP provides a separate acknowledgment, via a TCP acknowledgment packet, that a request has been received by the access control server within an approximate network round-trip time. This acknowledgment occurs regardless of the congestion of the access control server.

TCP provides immediate indication of an unavailable server. Long-lived TCP connections will even reveal servers that go down and then come back up. UDP is unable to discriminate between a down server, a slow server, and a nonexistent server.

Using TCP keepalives, server crashes can be detected out of band with actual requests. Connections to multiple servers can be maintained simultaneously, and messages need to be sent only to servers that are known to be up and running. TCP provides a stable foundation that efficiently adapts to growing as well as congested networks.

Confidentiality

RADIUS encrypts only the password in the Access-Request packet from the client to the server. The remainder of the packet is in the clear. Other information such as username, authorized services, and accounting could be captured by a third party, making RADIUS networks potential targets of hackers using session capture and replay attacks. Because of this inherent openness, RADIUS networks must be carefully designed to minimize the opportunities for attack.

TACACS+ encrypts the entire body of the packet but leaves a standard TACACS+ header. Within the header is a field that indicates whether or not the body is encrypted. Normal operation fully encrypts the body of the packet for more secure communications.

Distribution of Functionality

The RADIUS protocol combines the processes of authentication and authorization. The Access-Accept packets sent by the RADIUS server to the client contain all the authorization information, making separation of the authentication and authorization functions difficult. The use of RADIUS is most appropriate when simple, single-step authentication and authorization is required, as with most service provider networks.

TACACS+ uses the AAA architecture, which separates authentication, authorization, and accounting. This setup allows separate authentication solutions that can still use TACACS+ for authorization and accounting. For example, using TACACS+, it is possible to use Kerberos authentication and TACACS+ authorization and accounting. After a network access server authenticates to a Kerberos server, it requests authorization information from a TACACS+ server without having to reauthenticate. The NAS informs the TACACS+ server that it has successfully authenticated on a Kerberos server, and the server then provides authorization information.

During a session, if additional authorization checking is needed, the access server checks with a TACACS+ server to determine if the user is granted permission to use a particular command. This feature provides greater control over the commands that can be executed on the access server while decoupling authorization from the authentication mechanism. TACACS+ is thus more appropriate to use when multiple authentications in a complex network environment are anticipated. This scenario occurs, for example, inside most large corporate networks.

Because network security is exploding with many different competing solutions, the most scalable architecture must separate the AAA functionality and open the door to future options.

Multiprotocol Support

RADIUS has limited support for protocols other than TCP/IP. For example, RADIUS does not natively support the following protocols:

- AppleTalk Remote Access (ARA)
- NetBIOS frame protocol control
- Novell Asynchronous Services Interface (NASI)
- Packet assembler/disassembler (PAD) connection

These protocols are natively supported by TACACS+.

Cisco IOS Software Provides Users the Choice of Which Protocols to Use

Cisco has supported the RADIUS protocol since Cisco IOS Release 11.1 in February 1996. RADIUS support in Cisco IOS software continues to be enhanced with new features and capabilities. Cisco is committed to supporting RADIUS as a standard. Cisco's access servers are unique in that they implement both RADIUS and TACACS+.

Configuring RADIUS to Use TACACS+ Attribute/Value Pairs

The RADIUS protocol specifies a method by which special information not specifically defined as an attribute by the RADIUS specification can be transmitted. The vendor-specific attribute, number 26, can be configured to exchange any desired string of characters. Through the use of this variable, all the TACACS+ attribute/value pairs can be exchanged in a RADIUS environment. The TACACS+ information is passed in the following format:

```
ciscoav = "attribute=value"
```

The TACACS+ IETF Informational Internet Draft

Cisco is committed to open standards and continues to lead the industry in standards development. In October 1996, the IETF published Version 1.75 of the TACACS+ protocol specification as an informational Internet draft. The text of the IETF document can be found at the following URL:

`ftp://ftp.ietf.cnri.reston.va.us/internet-drafts/draft-grant-tacacs-00.txt`.

1 Because the RADIUS server stores its passwords in the clear, it will fail unexpectedly if a user with a clear text password sets it to "secrid." The RADIUS server will attempt to authenticate the user via the token card server, and the authentication will fail.

[Todito.com, 2003]

todito Card te ofrece los siguientes beneficios:

- Ahorras al pagar sólo las horas que usas.
- Excelente conexión.
- Fácil de conseguir

Adquiere la tarjeta que se acomode a tus necesidades de navegación:

15 horas / 900 minutos \$100
30 horas / 1,800 minutos \$150
90 horas / 5,400 minutos \$300

1. ¿Qué es todito Card?

¡Internet fácil y sin compromisos!

todito Card es una tarjeta prepagada para navegar en Internet. Con todito Card sólo pagas el tiempo de navegación que usas. Con **todito Card**, te olvidas de pagos mensuales y contratos que te comprometen por seis meses o un año. No necesitas insertar CD's que pueden desconfigurar tu computadora.

2. ¿Dónde puedo adquirir la tarjeta?

Puedes encontrar **todito Card** en más de 50 ciudades en toda la República.
Por favor consulte la sección "Puntos de Venta".

3. ¿Qué precio tiene?

Selecciona la opción que mejor se acomode a tus necesidades al momento de activar tu tarjeta.

Tabla de equivalencias todito Card

todito Card Precio	Actívala en Paquete de MINUTOS*	Actívala en Paquete de DÍAS**
\$100	900	15
\$150	1,800	25
\$185	2,400	30
\$300	5,400	60

todito Card	Actívala en Paquete de DÍAS**
3 meses por \$104 + IVA mensual	90
6 meses por \$101 + IVA mensual	180

*A consumir en 90 días / **De uso ilimitado

4. ¿Cómo la activo?

En la tarjeta se incluyen instrucciones de activación o bien puedes llamar a nuestro Centro de Atención a Clientes: en la Ciudad de México 5447-8864 y en el interior 01 800-080 0000 donde te guiaremos paso a paso.

5. ¿Cuáles son los requisitos mínimos para conectarme?

Si tu equipo es una **PC** o es compatible:

- Procesador 486 a 66 Mhz o superior.
- Sistema Operativo MS Windows 95 o superior.
- Mínimo de 16 MB en memoria RAM.
- Un Mínimo de 30 MB de espacio disponible en Disco Duro.
- Módem de 19200 bps o superior.
- Línea telefónica analógica.
- Dispositivo apuntador (Mouse).
- MS-Internet Explorer 4.0 o Netscape 3.0 o superiores.

Si tu equipo es una **Macintosh**:

- Procesador Power PC o superior.
- Sistema Operativo Mac OS 8 o superior.
- Mínimo de 16 MB en memoria RAM.
- Un mínimo de 40 MB de espacio disponible en Disco Duro.
- Módem de 19200 bps o superior.
- Línea telefónica analógica.
- MS-Internet Explorer 4.0 o Netscape 3.0 o superiores.

6. ¿Qué pasa cuando me acabo las horas de una tarjeta?

Puedes agregar tiempo al usuario que ya tienes registrado, acudiendo a comprar una tarjeta en cualquiera de nuestros puntos de venta.

Después de comprar una nueva tarjeta sólo:

Ve a la página www.toditocard.com.

Presiona "Activa tu tarjeta".

Selecciona "Agregar tiempo a una cuenta existente (si ya habías activado una tarjeta anteriormente)" y sigue los pasos que se te indican en la pantalla.

Si ya no tienes saldo en tu tarjeta, solo cambia tu nombre de usuario en tu acceso telefónico de redes por crearcuenta@todito.com y escribe la contraseña todito (En minúsculas), esto te permitirá navegar en la página de www.toditocard.com, para que puedas agregar más tiempo a tu cuenta con una nueva tarjeta siguiendo los pasos indicados arriba.

No olvides que únicamente con el nombre de usuario que tengas registrado en <http://www.toditocard.com/> podrás navegar por todo el Internet. Cambia nuevamente el nombre de usuario y contraseña en tu acceso telefónico a redes una vez que hayas añadido más tiempo a tu cuenta **todito Card**.

7. ¿Cómo sé cuanto tiempo me queda?

Ve a www.toditocard.com

Selecciona la opción consulta tu saldo y sigue los pasos que se indica en pantalla.

8. ¿En que ciudades puedo conectarme con todito Card?

Tenemos servicio en:

Acapulco	Coatzacoalcos	Matamoros	Poza Rica	Texcoco
Aguascalientes	Córdoba	Mazatlán	Puebla	Tijuana
Campeche	Cuautla	Mérida	Puerto Vallarta	Tlaxcala
Cancún	Cuernavaca	Mexicali	Querétaro	Toluca
Celaya	Culiacán	Monterrey	Reynosa	Torreón
Chihuahua	Durango	Morelia	Saltillo	Tula
Ciudad de México	Guadalajara	Nuevo Laredo	San Luis Potosí	Tulancingo
Ciudad del Carmen	Guanajuato	Oaxaca	Tampico	Tuxtla Gutiérrez
Ciudad Juárez	Hermosillo	Ocotlán	Tehuacan	Veracruz
Ciudad Obregón	Irapuato	Orizaba	Tepatitlán	Villahermosa
Ciudad Valles	Jalapa	Pachuca	Tepic	Zacatecas
Ciudad Victoria	León			

*Recuerda verificar que el número configurado le corresponda a la ciudad en la que te conectarás a Internet, así las llamadas serán locales.

*En caso de no ubicar en el catálogo, el nombre de la ciudad en donde realizarás la conexión a Internet, comunícate a nuestro centro de atención al cliente al teléfono 01 800-080-0000.

Ver números de acceso.

9. ¿Qué pasa si mi cuenta vence en 10 días con 300 minutos por usar y activo a mi cuenta una tarjeta en paquete de 30 días de uso ilimitado?

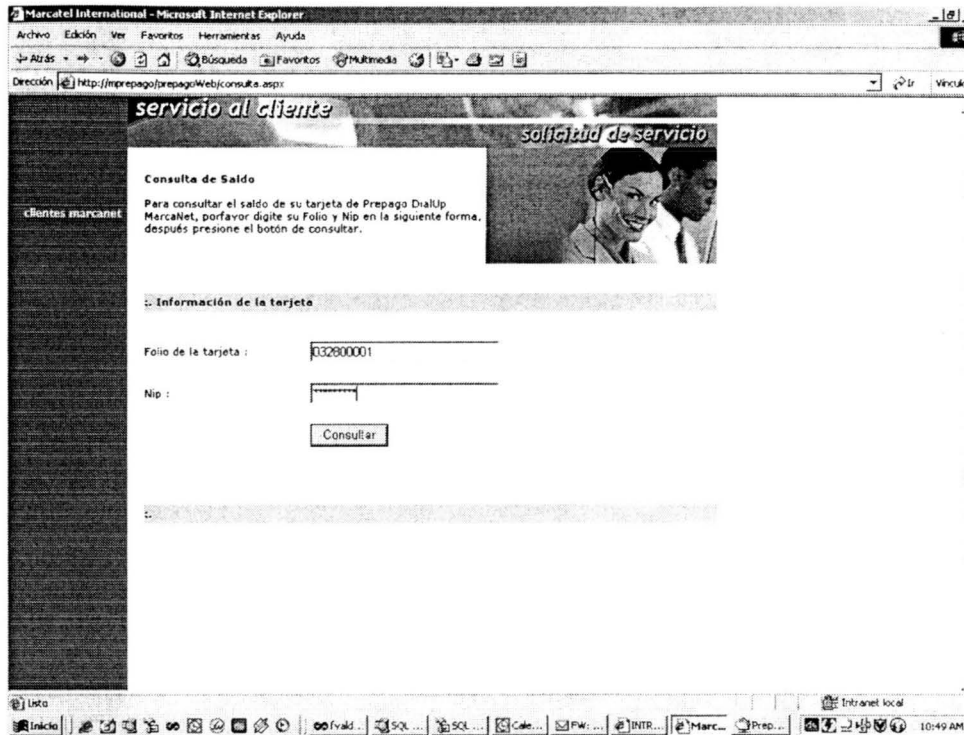
Al agregarse una tarjeta activada en paquete de días a una cuenta de horas, podrás navegar ilimitadamente por el periodo comprado (30 días por ejemplo). Al término del periodo ilimitado, tendrás el mismo plazo (10 días) para utilizar los minutos (300 horas) que tenías antes de activar tu todito card activada en paquete de días (en este caso 300 minutos).

10. ¿Qué pasa si estoy usando una todito Card activada en paquete de días que vence en 5 días y agrego a mi cuenta una tarjeta todito Card, por ejemplo de 900 minutos?

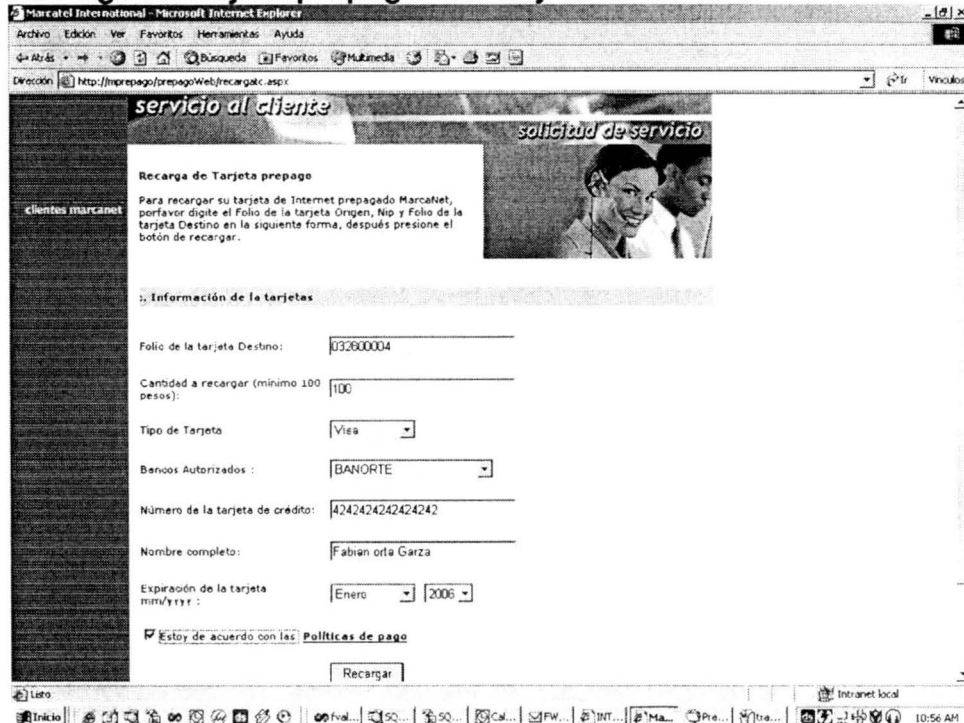
Podrás navegar ilimitadamente por el periodo que le quedaba a la tarjeta activada en paquete de días (5 días). Al término del periodo ilimitado, tendrás 3 meses para utilizarlos minutos (900 horas) que habías agregado a tu cuenta.

ANEXO A

Interfaces



Recarga de tarjeta prepago con tarjeta de crédito.



Traspaso de saldo de tarjeta prepago con tarjeta de crédito.

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page for 'servicio al cliente' (customer service). The page title is 'Recarga de Tarjeta prepago' (Prepaid Card Recharge). The main content area contains the following text:

Recarga de Tarjeta prepago
 Para recargar su tarjeta de Internet prepago MarcaNet, por favor digite el Folio de la tarjeta Origen, Nip y Folio de la tarjeta Destino en la siguiente forma, después presione el botón de recargar.

Below the text is a section titled 'Información de la tarjetas' (Card Information) with the following fields:

- Folio de la tarjeta Origen:
- Nip:
- Folio de la tarjeta Destino:

A 'Recargar' (Recharge) button is located below the destination card number field. The browser's address bar shows the URL 'http://mprepago/repagoWeb/recarga.aspx'. The taskbar at the bottom shows the system tray with the time '10:55 AM' and the date '10/5/2004'.

Monitoreo de saldos cliente

The screenshot shows a window titled 'MarcaNet Prepago'. The window contains a section titled 'Información de la Tarjeta' (Card Information) with the following fields:

- Tarjeta: **Desconectado**
- Saldo:
- Minutos restantes:
- Expiración del saldo:

At the bottom right of the window, there is a status indicator that reads 'Desconectado'.

Reporte de actividades con tarjeta de crédito

Reporte de actividades con tarjeta de crédito

Reporte de Actividad Diaria con Tarjeta de Crédito - Microsoft Internet Explorer

Address: http://localhost/Reportes/RepActividadTC.aspx

MARCATEL INTERNACIONAL

= Reporte de Actividad Diaria con Tarjeta de Crédito =
Del 20 Octubre del 2003

Actividad	Número de Tarjeta	Hora	Cantidad
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:20:43 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:23:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	45692301	29/08/2003 11:23:59 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:24:31 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:25:22 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	34560345	29/08/2003 11:26:07 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	29/08/2003 11:40:01 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	23450000	03/09/2003 11:58:34 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	65349078	17/09/2003 10:41:44 a.m.	0
Error: tarjeta de crédito inválida - System.Exception	39485234	17/09/2003 10:41:58 a.m.	0
Approved	65349078	17/09/2003 10:48:02 a.m.	160.9000
Approved	32094345	17/09/2003 10:55:52 a.m.	100.0000
Approved	65349023	17/09/2003 11:02:21 a.m.	250.0000
Approved	65201392	17/09/2003 11:12:42 a.m.	50.0000
Approved	23455079	17/09/2003 11:19:06 a.m.	124.9800
Approved	23455079	17/09/2003 11:22:28 a.m.	400.0000
Approved	65961179	17/09/2003 11:27:13 a.m.	340.0000
Approved	45693000	17/09/2003 11:36:21 a.m.	250.0000

Done Local intranet

Reporte de usuarios conectados al servicio Internet prepago.

Reporte de Usuarios Conectados - Microsoft Internet Explorer

Address: http://localhost/Reportes/WebForm1.aspx

MARCATEL INTERNACIONAL

= Reporte de Usuarios Conectados =

Cuenta	Hora	Tarifa
0328000001	06/06/2003 01:11:16 p.m.	0.7
0328000003	06/06/2003 05:37:19 p.m.	1.15
0328000012	06/06/2003 08:19:47 p.m.	0.5

Marcatel International 20003


Done Local intranet

Reporte de Actividad Diaria - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost/Reportes/RepActividadDiaria.aspx

MARCA TEL INTERNACIONAL



= Reporte de Actividad Diaria =
Del 20 Octubre del 2003

Cuenta	Hora	Tarifa
acerotek	06/11/2003 04:07:00 p.m.	0.65
acerotek	06/11/2003 04:07:00 p.m.	0.65
acerotek	06/11/2003 04:14:13 p.m.	0.65
acerotek	06/11/2003 04:14:13 p.m.	0.65
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
0328000002	06/11/2003 04:14:14 p.m.	0.7
acerotek	06/11/2003 04:14:34 p.m.	0.65
acerotek	06/11/2003 04:14:34 p.m.	0.65
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
0328000002	06/11/2003 04:14:34 p.m.	0.7
acerotek	06/11/2003 04:15:03 p.m.	0.65
acerotek	06/11/2003 04:15:03 p.m.	0.65

Done Local intranet

ANEXO B

Código Fuente del Sistema

Script SQL para generación de tablas

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[FK_Log_Actividades_Actividades]'
) and OBJECTPROPERTY(id, N'IsForeignKey') = 1)
ALTER TABLE [dbo].[Log_Actividades] DROP
CONSTRAINT FK_Log_Actividades_Actividades
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Actividades]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Actividades]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ActividadesTC]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[ActividadesTC]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[AreaGeografica]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[AreaGeografica]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[ClienteSerie]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[ClienteSerie]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[DefHorario]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[DefHorario]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Denominacion]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Denominacion]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[DominioAreaGeografica]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[DominioAreaGeografica]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[DominioTipoDia]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[DominioTipoDia]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Estaus_NIP_Dialup]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Estaus_NIP_Dialup]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Horario]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Horario]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Log_Actividades]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Log_Actividades]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Nips]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Nips]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Nips_DialUp]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Nips_DialUp]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Serie]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Serie]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Tarifa]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Tarifa]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Tarifas]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Tarifas]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Tarjeta]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Tarjeta]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[TipoDia]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[TipoDia]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[UsersConnected]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[UsersConnected]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[nipDisponibles]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[nipDisponibles]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[radiusaccounting]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[radiusaccounting]
GO
```

```
CREATE TABLE [dbo].[Actividades] (
[ActividadID] [smallint] IDENTITY (1, 1)
NOT NULL ,
```

```

        [ActividadDes] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ActividadesTC] (
    [transaccionID] [bigint] IDENTITY (10000,
1) NOT NULL ,
    [LogID] [bigint] NOT NULL ,
    [Login] [varchar] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [cantidad] [money] NOT NULL ,
    [ini_fecTransaccion] [datetime] NOT NULL
,
    [fin_fecTransaccion] [datetime] NOT NULL
,
    [tipoTarjeta] [smallint] NOT NULL ,
    [banco] [smallint] NOT NULL ,
    [referencia] [bigint] NOT NULL ,
    [respuesta] [varchar] (100) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [nombre] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [tarjetaCredito] [varchar] (8) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[AreaGeografica] (
    [AreaID] [smallint] NOT NULL ,
    [Descripcion] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [NoImporta] [bit] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ClienteSerie] (
    [SerieID] [int] NOT NULL ,
    [CustomerID] [numeric](9, 0) NOT NULL ,
    [OrdenID] [int] NOT NULL ,
    [TarjetaID] [tinyint] NOT NULL ,
    [FecAsigna] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[DefHorario] (
    [ID] [smallint] NOT NULL ,
    [TipoTipoDiald] [smallint] NOT NULL ,
    [Horalnicio] [varchar] (12) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [HoraFin] [varchar] (12) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Horario] [tinyint] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Denominacion] (
    [DenominacionID] [tinyint] NOT NULL ,
    [DenominacionDes] [money] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[DominioAreaGeografica] (
    [AreaID] [smallint] NOT NULL ,
    [Numero] [varchar] (20) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[DominioTipoDia] (
    [TipoDiald] [smallint] NOT NULL ,
    [fecha] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Estatus_NIP_Dialup] (
    [estatus] [smallint] IDENTITY (1, 1) NOT
NULL ,
    [descripcion] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Horario] (
    [Horario] [tinyint] NOT NULL ,
    [Descripcion] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Log_Actividades] (
    [LogID] [bigint] IDENTITY (1, 1) NOT NULL
,
    [ActividadID] [smallint] NOT NULL ,
    [LogDes] [varchar] (100) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [FecLog] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Nips] (
    [SerieID] [int] NOT NULL ,
    [Folio] [int] NOT NULL ,
    [TarjetaNum] [varchar] (15) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Nips] [varchar] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Fecha] [datetime] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Nips_DialUp] (
    [SerieID] [int] NOT NULL ,
    [Folio] [int] NOT NULL ,
    [Login] [varchar] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Saldo] [money] NOT NULL ,
    [Estatus] [smallint] NOT NULL ,
    [Fecexpiracion] [datetime] NOT NULL ,
    [Fecactivacion] [datetime] NULL ,
    [No_recargas] [int] NOT NULL ,
    [ProductoID] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Serie] (
    [SerieID] [int] NOT NULL ,
    [FolioIni] [int] NOT NULL ,
    [FolioFin] [int] NOT NULL ,
    [NumTarjeta] [int] NOT NULL ,
    [FecActivacion] [datetime] NOT NULL ,
    [FecVencimiento] [datetime] NOT NULL ,
    [TarjetaID] [tinyint] NOT NULL ,
    [PlanID] [int] NULL ,
    [IN800ID] [tinyint] NOT NULL ,
    [SAC800ID] [tinyint] NOT NULL ,
    [PlataformaID] [tinyint] NOT NULL ,
    [EstatusID] [tinyint] NOT NULL ,
    [FecEstatus] [datetime] NOT NULL ,
    [InvFisico] [bit] NULL

```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Tarifa] (
    [Productoid] [int] NOT NULL ,
    [Origen] [smallint] NOT NULL ,
    [Destino] [smallint] NOT NULL ,
    [Id] [smallint] NOT NULL ,
    [Vigencialni] [datetime] NOT NULL ,
    [VigenciaFin] [datetime] NOT NULL ,
    [Tarifa] [money] NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Tarifas] (
    [TarjetaID] [smallint] NOT NULL ,
    [tarifaMin] [money] NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Tarjeta] (
    [TarjetaID] [tinyint] NOT NULL ,
    [DistribuidorID] [tinyint] NOT NULL ,
    [DenominacionID] [tinyint] NOT NULL ,
    [TarjetaDes] [varchar] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Reorden] [smallint] NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[TipoDia] (
    [TipoDiald] [smallint] NOT NULL ,
    [Descripcion] [varchar] (30) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[UsersConnected] (
    [id_user] [int] IDENTITY (1, 1) NOT NULL ,
    [User_Name] [varchar] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [SessionID] [bigint] NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[nipDisponibles] (
    [folio] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [nip] [varchar] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
)
```

```
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[radiusaccounting] (
    [SessionID] [bigint] NOT NULL ,
    [LoggedAt] [datetime] NULL ,
    [Group_Name] [varchar] (64) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [User_Name] [varchar] (64) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Acct_Status_type] [int] NULL ,
    [Calling_station_ID] [varchar] (32)
COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
    [Call_station_ID] [varchar] (32) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL
)
```

```
) ON [PRIMARY]
GO
```

Stored Procedures

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_DelUsuariosIniciales]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_DelUsuariosIniciales]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsAcc]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsAcc]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsActividadesDiarias]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsActividadesDiarias]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsConsecutivo]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsConsecutivo]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsDialUP]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsDialUP]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsLog]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsLog]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsLote]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsLote]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsUsuariosConectados]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsUsuariosConectados]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsUsuariosConectadosRadius]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure
[dbo].[Sp_InsUsuariosConectadosRadius]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_InsUsuariosIniciales]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_InsUsuariosIniciales]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_Instarjeta]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_Instarjeta]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelDetalleTarjeta]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
```

```
drop procedure [dbo].[Sp_SelDetalleTarjeta]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelFolio]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelFolio]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelFolioNip]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelFolioNip]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelInfoPlugin]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelInfoPlugin]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelInfoTarjeta]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelInfoTarjeta]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelIntentos]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelIntentos]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelNipDisponible]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelNipDisponible]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelSaldo]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelSaldo]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelSesionesValidas]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelSesionesValidas]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_SelTarjDisp]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_SelTarjDisp]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdCobroTarifa]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_UpdCobroTarifa]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdEstatusTarj]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_UpdEstatusTarj]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdPago]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_UpdPago]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdRecargaTarjetaTarjeta]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure
[dbo].[Sp_UpdRecargaTarjetaTarjeta]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdSaldo]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_UpdSaldo]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_UpdTarifas]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_UpdTarifas]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[Sp_checa_tarj_disp]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[Sp_checa_tarj_disp]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_DelUsuarioSaldo]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_DelUsuarioSaldo]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_DelUsuariosDesconectados]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure
[dbo].[sp_DelUsuariosDesconectados]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_DelUsuariosDesconectadosRa
dus]') and OBJECTPROPERTY(id, N'IsProcedure')
= 1)
drop procedure
[dbo].[sp_DelUsuariosDesconectadosRadius]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[prepago].[sp_SMTPMail]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [prepago].[sp_SMTPMail]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_SelUsuariosConectados]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_SelUsuariosConectados]
GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_cobroUsuarios]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_cobroUsuarios]
GO
```

```

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[sp_verificaSalDOS]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[sp_verificaSalDOS]
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_DelUsuariosIniciales
AS
BEGIN
    BEGIN TRAN myTran
        Delete UsersConnected
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
borrar los usuarios iniciales'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_InsAcc
AS
BEGIN
    BEGIN TRAN myTran
        INSERT Into radiusaccounting
        VALUES(getdate(),'DU-
Normal',",",1,'0000000000')
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar de insertar en Acc'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_InsActividadesDiarias
    @UserName varchar(64),
    @sessionID varchar(64),
    @Origen varchar(32),
    @Destino varchar(32),
    @Tarifa money
AS
    BEGIN TRAN myTran
        Insert Into ActividadDiaria
            values(@UserName,@SessionID,@Orige
n,@Destino,getdate(),@Tarifa)
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar de cobrar tarifa'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_InsConsecutivo
    @nombre varchar(50),
    @cantidad money,
    @tarjetaCredito varchar(8),
    @folio varchar(10),
    @tipoTarjeta smallint,
    @banco smallint
AS
DECLARE
    @id_log bigint
BEGIN
    BEGIN TRAN myTran
        SET nocount on
        set @id_log=( SELECT
max(Logid) from log_actividades)
        INSERT INTO ActividadesTC

            VALUES(@id_log,@folio,@cantidad,getDa
te(),getdate(),@tipoTarjeta,@banco,0,"",@nombre,@
tarjetaCredito)
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100
'Error al tratar de insertar en RecargaTC'
                ROLLBACK TRAN
myTran
            END
        ELSE
            BEGIN
                RETURN 2
            END
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_InsDialUP

```

```

AS
BEGIN
DECLARE
    @num bigint,
    @serieID int,
    @Folio int,
    @Denominacion money
set @num=0
BEGIN TRAN myTran
    DECLARE tarjetas_dialup
CURSOR FOR
    SELECT SerieID,Folio FROM
Nips
    OPEN tarjetas_dialup
    FETCH NEXT FROM
tarjetas_dialup
    INTO @serieID,@Folio
    WHILE @@FETCH_STATUS =
0
    BEGIN
        SET @Denominacion=(SELECT
D.denominacionDes from Denominacion D,Tarjeta
T,Serie S
        WHERE S.SerieID=@serieID
and S.TarjetaID=T.TarjetaID and
T.DenominacionID=D.denominacionID)
        INSERT INTO
Nips_DialUp
        VALUES(@serieID,@Folio,RIGHT('0000'+
convert(varchar,@SerieID),4)+
RIGHT('000000'+convert(varchar,@Folio),6),@deno
minacion,0,',',0,0)
        Set @num=@num+1
        PRINT 'Row numero: '+
convert(varchar,@num)
    FETCH NEXT FROM
tarjetas_dialup
    INTO @serieID,@Folio
    END
    CLOSE tarjetas_dialup
    DEALLOCATE tarjetas_dialup
    IF @@ERROR<>0
    BEGIN
        RAISERROR 2000100 'Error al
tratar de insertar en InsDialUp'
        ROLLBACK TRAN myTran
        RETURN
    END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO
CREATE Procedure Sp_InsLog
    @Tipo smallint,
    @folioOrigen varchar(10),
    @folioDestino varchar(10),
    @monto money,
    @Descripcion varchar(100)
AS
BEGIN TRAN myTran
    INSERT INTO Log_Actividades
        VALUES(@Tipo,@folioOrigen,@FolioDesti
no,@monto,@Descripcion,getdate())
    IF @@ERROR<>0
    BEGIN
        RAISERROR 2000100 'Error al
tratar de insertar en InsDialUp'
        ROLLBACK TRAN myTran
        RETURN
    END
    COMMIT TRAN
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO
CREATE Procedure Sp_InsLote
AS
DECLARE
    @NumLote varchar(4)
    Set @NumLote =right('0000' +
convert(varchar,(select ISNULL(max(id_lote),'0') from
Lotes)+1),4)
BEGIN
    BEGIN TRAN myTran
        INSERT Into Lotes
        VALUES(@NumLote,'MiLote')
        SELECT @NumLote
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de insertar en Lotes'
            ROLLBACK TRAN myTran
            RETURN
        END
        COMMIT TRAN
    END
    GO
    SET QUOTED_IDENTIFIER OFF
    GO
    SET ANSI_NULLS ON
    GO
    SET QUOTED_IDENTIFIER ON
    GO
    SET ANSI_NULLS OFF
    GO
    CREATE Procedure Sp_InsUsuariosConectados
AS
BEGIN
    BEGIN TRAN myTran
        INSERT Into UsersConnected
        SELECT
        left(User_Name,10),SessionID from
        radiusaccounting
        WHERE Acct_Status_Type=1
        IF @@ERROR<>0

```

```

        BEGIN
        RAISERROR 2000100 'Error al
tratar de insertar en Acc'
        ROLLBACK TRAN myTran
        RETURN
        END
        COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure
Sp_InsUsuariosConectadosRadius
@SessionId as varchar(64),
@UserName as varchar(64),
@CallingId as varchar(32),
@CalledId as varchar(32),
@LoggedAt as datetime

AS
BEGIN
        BEGIN TRAN myTran
                IF Not EXISTS(Select sessionid
from usersconnected where sessionid=@SessionId)
                BEGIN
                        INSERT Into
UsersConnected

                        VALUES(@UserName,@SessionID,@Call
ingID,@CalledId,@LoggedAt,0)
                END
                IF @@ERROR<>0
                BEGIN
                        RAISERROR 2000100 'Error al
tratar de insertar en Acc'
                        ROLLBACK TRAN myTran
                        RETURN
                END
                COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_InsUsuariosIniciales
@sessionID varchar(64),
@userName varchar(64),
@origen varchar(32),
@destino varchar(32)

AS
BEGIN
        BEGIN TRAN myTran
                INSERT Into UsersConnected
                VALUES(@userName,@sessionID,@orige
n,@destino,getDate(),0)
                IF @@ERROR<>0
                BEGIN
                        RAISERROR 2000100 'Error al
tratar de insertar en Acc los usuarios iniciales'
                        ROLLBACK TRAN myTran
                        RETURN
                END
                COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_Instarjeta
@Numtarjeta varchar(10),
@Lote varchar(4),
@Nip varchar(10),
@fechaExpiracion datetime,
@saldo money

AS
BEGIN
        BEGIN TRAN myTran
                INSERT INTO Tarjetas

                VALUES(@Numtarjeta,@Nip,getDate(),'1/1
/1900',@fechaExpiracion,0,@saldo,0,0)

                INSERT INTO nipDisponibles
VALUES(@Numtarjeta,@Nip)

                INSERT INTO Lote_tarjetas
VALUES(@Lote,@Numtarjeta)

                IF @@ERROR<>0
                BEGIN
                        RAISERROR 2000100 'Error al
tratar de insertar en Lotes'
                        ROLLBACK TRAN myTran
                        RETURN
                END
                SELECT @@Identity
                COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelDetalleTarjeta
@userName varchar(64),
@sessionID varchar(64)

AS
BEGIN
        BEGIN TRAN myTran

```

```

/*
    Select
    @userName,N.Saldo,N.fecExpiracion,U.Tarifa
    from Nips_Dialup N,usersconnected U
    where U.sessionID=@sessionID and
    U.user_name=@username
    and U.user_name=N.login
*/
Select top 1
@userName,N.Saldo,N.fecExpiracion,U.Tarifa
from Nips_Dialup N,usersconnected U
where U.user_name=@username
and U.user_name=N.login
IF @@ERROR<>0
    BEGIN
        RAISERROR 2000100 'Error al
tratar seleccionar el detalle de la tarjeta'
        ROLLBACK TRAN myTran
        RETURN
    END
COMMIT TRAN

END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelFolio
    @folio varchar(10)
AS
BEGIN
    BEGIN TRAN myTran
        IF EXISTS(Select Login from
Nips_Dialup where Login=@folio)
            Select Convert(bit,1)
        ELSE
            Select Convert(bit,0)
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar de Validar Tarjeta'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelFolioNip
    @folio varchar(10),
    @nip varchar(10)
AS
BEGIN
    BEGIN TRAN myTran
        IF EXISTS(Select D.Login from
Nips_Dialup D,Nips N where D.Login=@folio and
D.SerialID=N.SerialID and D.Folio=N.Folio and
N.Tarjetanum=@nip)
            Select Convert(bit,1)
        ELSE
            Select Convert(bit,0)
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar de Validar Tarjeta'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelInfoPlugin
    @IP varchar(15)
AS
BEGIN
    BEGIN TRAN myTran
        Select
login,saldo,convert(integer,(saldo/3)),fecexpiracion
from Nips_Dialup D,Nips N where
D.SerialID=N.SerialID and D.Folio=N.Folio and d.login
='acerotek'
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar seleccionar Informacion de la tarjeta para
plugin'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelInfoTarjeta
    @folio varchar(10)
AS
BEGIN
    BEGIN TRAN myTran
        Select saldo,fecexpiracion from
Nips_Dialup D,Nips N where D.SerialID=N.SerialID
and D.Folio=N.Folio and d.login = @folio
        IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar seleccionar Informacion de la tarjeta'

```

```

        ROLLBACK TRAN myTran
        RETURN
    END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelIntentos
    @nombre varchar(50),
    @tarjetaCredito varchar(8)
AS
DECLARE
    @noIntentos int,
    @today datetime
SET @today =(select
convert(datetime,convert(varchar,datepart(mm,getdate()))+'/' +
convert(varchar,datepart(dd,getdate()))+'/' +
convert(varchar,datepart(yy,getdate()))))
    SET @noIntentos=(SELECT
ISNULL(count(login),0) from Actividadestc
                                WHERE
nombre=@nombre AND
tarjetaCredito=@tarjetaCredito

                                AND Ini_FecTransaccion >@today

                                AND Ini_FecTransaccion<
DateAdd(d,1,@today))
BEGIN
    Select @noIntentos
    BEGIN TRAN myTran
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de seleccionar Intentos'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_SelNipDisponible
    @Nip varchar(10)
AS
DECLARE
    @numRegistros smallint
set @numRegistros = (Select count(Nip)
from Nipdisponibles where nip=@Nip)
BEGIN
    BEGIN TRAN myTran
        IF @numRegistros = 0
            Select "Exitoso"
        ELSE
            Select "Repetido"
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de insertar en Lotes'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelSaldo
    @folio varchar(10)
AS
BEGIN
    BEGIN TRAN myTran
        Select saldo from Nips_Dialup
        WHERE login = @folio
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar seleccionar el saldo de la tarjeta'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelSesionesValidas
    @UserName varchar(64)
AS
BEGIN
    BEGIN TRAN myTran
        Select count(*) from Nips_DialUp
        WHERE Login=@UserName
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar seleccionar las cuentas validas'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_SelTarjDisp
    @denominacion as integer
as
BEGIN
    BEGIN TRAN myTran
select top 1 nips_dialup.login ,nips.tarjetanum
from ((tarjeta Inner Join serie on tarjeta.tarjetaid =
serie.tarjetaid)
inner join nips on serie.serieid =
nips.serieid)
inner join nips_dialup on (nips.serieid =
nips_dialup.serieid and nips.folio = nips_dialup.folio)
where distribuidorid = 2
and estatus = 0
and denominacionid = (select
denominacion.denominacionid from denominacion
where denominaciondes = @denominacion)

        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar seleccionar el saldo de la tarjeta'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_UpdCobroTarifa
    @login varchar(10),
    @sessionID varchar(64)
AS
DECLARE
    @tarifa money,
    @saldo money
set @tarifa=(Select max(tarifa) from
usersconnected where sessionid=@sessionid and
user_name=@login
)
set @saldo=(Select saldo-
convert(money,@tarifa) from Nips_dialUp WHERE
login=@login)

    IF @saldo<=0
    Set @saldo=0
    BEGIN TRAN myTran

        UPDATE Nips_DialUp
SET saldo=@saldo
WHERE login=@login
IF @@ERROR<>0
    BEGIN
        RAISERROR 2000100 'Error al
tratar de cobrar tarifa'
        ROLLBACK TRAN myTran
        RETURN
    END
    COMMIT TRAN
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_UpdEstatusTarj
    @folio as varchar(10)
as
BEGIN
    BEGIN TRAN myTran
update nips_dialup set estatus = 1 where
nips_dialup.login = @folio

        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar seleccionar el saldo de la tarjeta'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_UpdPago
    @consecutivo bigint,
    @referencia bigint,
    @respuesta varchar(50),
    @fecIni datetime,
    @fecFin datetime
AS
BEGIN
    BEGIN TRAN myTran
        SET nocount on
        UPDATE ActividadesTC
        SET referencia=@referencia,
        respuesta=@respuesta, Ini_fecTransaccion=@fecIni,
        fin_fecTransaccion=@fecFin
        WHERE
        transaccionID=@consecutivo
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de Modificar en RecargaTC'
            ROLLBACK TRAN myTran
            RETURN 2
        END
        SELECT @@Identity
    COMMIT TRAN

```

```

END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE
Sp_UpdRecargaTarjetaTarjeta
    @folioOrigen varchar(10),
    @folioDestino varchar(10)
AS
DECLARE
    @saldoOrigen money,
    @SaldoDestino money
SET @saldoOrigen=(Select saldo from
Nips_DialUp where Login=@folioOrigen)
SET @saldoDestino=(Select saldo from
Nips_DialUp where Login=@folioDestino)
BEGIN
    BEGIN TRAN myTran
        UPDATE Nips_DialUp
        SET
saldo=@saldoOrigen+@saldoDestino
        WHERE Login=@folioDestino
        UPDATE Nips_DialUp
        SET saldo=((Select saldo from
Nips_DialUp where Login=@folioOrigen)-
@saldoOrigen)
        WHERE Login=@folioOrigen
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de recargar Tarjeta con Tarjeta'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE Sp_UpdSaldo
    @folio varchar(10),
    @saldo money
AS
DECLARE
    @norecargas int,
    @FecExp datetime,
    @saldoNuevo money
SET @norecargas=(Select no_recargas+1
from Nips_DialUp where login=@folio)
SET @saldoNuevo=(Select saldo+@saldo
from Nips_DialUp where login=@folio)
IF @saldo <500
    BEGIN
        SET @FecExp=DateAdd(m,3,getdate())
        END
    ELSE
        BEGIN
            SET @FecExp=DateAdd(m,6,getdate())
            END
        BEGIN TRAN myTran
            UPDATE Nips_DialUp
            SET saldo=@saldoNuevo,
fecExpiracion=@fecExp, no_recargas=@norecargas
            WHERE login = @folio
            IF @@ERROR<>0
            BEGIN
                RAISERROR 2000100 'Error al
tratar de modificar el saldo de la tarjeta'
                ROLLBACK TRAN myTran
                RETURN
            END
        COMMIT TRAN
    END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

CREATE Procedure Sp_UpdTarifas
    @areaDestino varchar(10),
    @login varchar(10),
    @sessionID varchar(64)
AS
DECLARE
    @productoID int,
    @tarifa money,
    @horaTarifa varchar(50),
    @minTarifa varchar(50),
    @horafinal datetime
set @ProductoID=(select T.PlanID from
nips_DialUp ND,Nips N,Serie S,Tarjeta T
where ND.Login=@login and
ND.FOLio=N.Folio and ND.SerialID=n.SerialID
AND N.Serial=S.SerialID AND
S.TarjetaID=T.TarjetaID)
set
@horaTarifa=convert(varchar,datepart(hh,getdate()))
set
@minTarifa=convert(varchar,datepart(mi,getdate()))
set @horafinal=
convert(datetime,'1/1/1900')
set
@horafinal=dateadd(hh,convert(smollint,@horaTarifa
),@horafinal)
set
@horafinal=dateadd(mi,convert(smollint,@minTarifa)
,@horafinal)
set @tarifa=(Select
ISNULL(min(TF.Tarifa),0.5) from DominioTipoDia
D,TipoDia T,DefHorario H,Tarifa TF,AreaGeografica
AG,DominioAreaGeografica DAG where
datepart(day,D.fecha)=datepart(day,getdate()) and
datepart(mm,D.fecha)=datepart(mm,getdate()) and
datepart(yy,D.fecha)=datepart(yy,getdate())

```

```

        and T.TipoDiald=D.TipoDiald
and H.TipoDiald=T.TipoDiald and H.id=TF.Id and
H.Horainicio<@horafinal and H.Horafin>@horafinal
AND ProductoID=@ProductoID and
TF.Destino=AG.Areald and AG.Areald=DAG.Areald
and DAG.numero=@areaDestino
    )

```

```
BEGIN TRAN myTran
```

```

        UPDATE UsersConnected
        SET Tarifa=@tarifa
        WHERE User_name=@login
and sessionID=@sessionID
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de actualizar tarifa'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN

```

```

GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

```

```

CREATE PROCEDURE Sp_checa_tarj_disp
    @denominacion as integer
as
BEGIN
    BEGIN TRAN myTran
select count ( nips_dialup.login )
from ((tarjeta Inner Join serie on tarjeta.tarjetaid =
serie.tarjetaid)
inner join nips on serie.serieid =
nips.serieid)
inner join nips_dialup on (nips.serieid =
nips_dialup.serieid and nips.folio = nips_dialup.folio)
where distribuidorid = 2
and estatus = 0
and denominacionid = (select
denominacion.denominacionid from denominacion
where denominaciondes = @denominacion)

```

```

        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar seleccionar el saldo de la tarjeta'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN

```

```

END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

```

```

CREATE PROCEDURE sp_DelUsuarioSaldo
    @login varchar(10),
    @SessionID varchar(64)
AS
BEGIN
    BEGIN TRAN myTran
        DELETE UsersConnected
        WHERE user_name=@login
        and SessionID=@SessionID
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
borrar de usuario '
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN

```

```

END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

```

```

CREATE PROCEDURE
sp_DelUsuariosDesconectados
AS
DECLARE
    @user_name varchar(64),
    @id int

```

```

BEGIN
    BEGIN TRAN myTran
        DECLARE usuarios_cursor
CURSOR FOR
        SELECT User_name FROM
radiusaccounting
        WHERE Acct_status_type=2
        OPEN usuarios_cursor
        FETCH NEXT FROM
usuarios_cursor
        INTO @user_name
        WHILE @@FETCH_STATUS =
0
        BEGIN
            SET @id=(SELECT top 1
ISNULL(id_user,0) from UsersConnected
        WHERE
User_name=@user_name)
        FETCH NEXT FROM
usuarios_cursor
        INTO @user_name
        DELETE
UsersConnected
        WHERE id_user=@id
        END
        CLOSE usuarios_cursor
        DEALLOCATE usuarios_cursor
    END
    IF @@ERROR<>0

```

```

        BEGIN
        RAISERROR 2000100 'Error al
tratar de insertar en Acc'
        ROLLBACK TRAN myTran
        RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE
sp_DelUsuariosDesconectadosRadius
@UserName varchar(64),
@SessionId varchar(32)
AS
BEGIN
    BEGIN TRAN myTran
        DELETE UsersConnected
        WHERE
User_Name=@UserName
        and SessionID=@SessionId
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
tratar de borrar en UsersConnected'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO

setuser N'prepago'
GO

CREATE Procedure sp_SMTPMail
    @SenderName varchar(100),
    @SenderAddress varchar(100),
    @RecipientName varchar(100),
    @RecipientAddress varchar(100),
    @Subject varchar(200),
    @Body varchar(8000),
    @MailServer varchar(100) = 'localhost'
AS
    SET nocount on

    declare @oMail int --Object reference
    declare @resultcode int

        EXEC @resultcode = sp_OACreate
        'Outlook.Mail', @oMail OUT
        print @resultcode
        if @resultcode = 0
        BEGIN
            EXEC @resultcode =
            sp_OASetProperty @oMail, 'RemoteHost',
            @mailserver
            EXEC @resultcode =
            sp_OASetProperty @oMail, 'FromName',
            @SenderName
            EXEC @resultcode =
            sp_OASetProperty @oMail, 'FromAddress',
            @SenderAddress

            EXEC @resultcode = sp_OAMethod
            @oMail, 'AddRecipient', NULL, @RecipientName,
            @RecipientAddress

            EXEC @resultcode = sp_OASetProperty
            @oMail, 'Subject', @Subject

            EXEC @resultcode = sp_OASetProperty
            @oMail, 'BodyText', @Body

            EXEC @resultcode = sp_OAMethod
            @oMail, 'SendMail', NULL
            print 'Mail enviado'

            EXEC sp_OADestroy @oMail
            END

        SET nocount off

GO
setuser
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

CREATE PROCEDURE sp_SelUsuariosConectados
AS
BEGIN
    BEGIN TRAN myTran
        SELECT
User_name,SessionID,Calling_Station,Called_station
,Tarifa FROM usersconnected
        IF @@ERROR<>0
        BEGIN
            RAISERROR 2000100 'Error al
Seleccionar UsuariosConectados'
            ROLLBACK TRAN myTran
            RETURN
        END
    COMMIT TRAN
END
GO

```



```

        <a href="consulta.aspx"><IMG src="imgs/consulta.jpg"
onMouseOver="Javascript:
src='imgs/consulta_up.jpg';"
onMouseOut="Javascript:
src='imgs/consulta.jpg';"></a>
        <br>
        <a href="compras.aspx"><IMG src="imgs/compra.jpg"
onMouseOver="Javascript:
src='imgs/compra_up.jpg';"
onMouseOut="Javascript:
src='imgs/compra.jpg';"></a>
        <br>
        <a href="recarga_menu.aspx"><IMG src="imgs/recarga.jpg"
onMouseOver="Javascript:
src='imgs/recarga_up.jpg';"
onMouseOut="Javascript:
src='imgs/recarga.jpg';"></a>
        </td>
        <td align="left">
        <table height="100%" cellSpacing="0"
cellPadding="0" border="0">
        <tr>
        <td colspan="3"><IMG src="imgs/banner.gif"></td>
        </tr>
        <tr>
        <td width="50"></td>
        <td colspan="2" height="30"></td>
        <td class="Normal"
vAlign="top"><br>
        <br>
        <b>Servicios Marcatel</b>
        <br>
        <br>
        Utilice
nuestros servicios en Línea para consultar,
comprar o recargar sust
tarjetas de
Internet Prepago.
        </td>
        <td align="right"><IMG
src="imgs/foto_registro.gif"></td>
        </tr>

```

```

        <tr>
        <td width="50"></td>
        <td colspan="2" height="30"></td>
        <td colspan="2" height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
        <tr>
        <td colspan="2" height="30"></td>
        </tr>
        <tr>
        <td colspan="2" height="30"></td>
        <td colspan="2" height="30"></td>
        <td colspan="2" class="NormalBold"
bgColor="#becdda" colspan="2" height="20">..
Servicios en
Línea
        </td>
        </tr>
        <tr>
        <td colspan="2" height="30"></td>
        <td colspan="2" height="30"></td>
        <td colspan="2" class="Normal" width="129"></td>
        <td align="left"><li><a
href="consulta.aspx">Consulta
saldo</li></td>
        </tr>
        <tr>
        <td colspan="2" height="20"></td>
        </tr>

```

```

        bgColor="#becdda"
        height="20">.</td>
        colSpan="2"

</tr>

</table>

<td class="Normal" width="129"></td>

<td align="left"><li><a
href="compras.aspx">Compra </a>de tarjetas de
Internet

prepagado ( con
tarjeta de crédito )</li></td>

</tr>

</table>

<td width="100%"></td>

</tr>

</table>

</td>

<td width="100%"></td>

</tr>

</table>

</form>

</body>

</HTML>

Inicio Servicios Cliente Code Behind

Public Class Inicio
    Inherits System.Web.UI.Page

    #Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()>
    Private Sub InitializeComponent()

        End Sub

    Private Sub Page_Init(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Init
        'CODEGEN: This method call is required by the
        Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

End Class

Menu de Recarga de Tarjeta ASP

<%@ Page Language="vb"
AutoEventWireup="false"
Codebehind="recarga_menu.aspx.vb"
Inherits="PrepagoWeb.Inicio_recarga" %>
<HTML>
    <HEAD>
        <title>Marcatel
        International</title>
        <meta http-equiv="Content-
        Type" content="text/html; charset=iso-8859-1">
        <td class="NormalBold"

```

```

        <LINK href="Styles.css"
type="text/css" rel="stylesheet">
    </HEAD>
    <body leftMargin="0" topMargin="0">
        <form runat="server">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
                <tr>
                    <td vAlign="top" width="130"
background="imgs/barra.gif"><br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <a href="consulta.aspx"><IMG
src="imgs/consulta.jpg"
onMouseOver="Javascript:
src='imgs/consulta_up.jpg';"
onMouseOut="Javascript:
src='imgs/consulta.jpg';"></a>
                    <br>
                    <a href="compras.aspx"><IMG
src="imgs/compra.jpg"
onMouseOver="Javascript:
src='imgs/compra_up.jpg';"
onMouseOut="Javascript:
src='imgs/compra.jpg';"></a>
                    <br>
                    <a href="recarga.aspx"><IMG
src="imgs/recarga.jpg"
onMouseOver="Javascript:
src='imgs/recarga_up.jpg';"
onMouseOut="Javascript:
src='imgs/recarga.jpg';"></a>
                    </td>
                    <td align="left">
                        <table height="100%" cellSpacing="0"
cellPadding="0" border="0">
                            <tr>
                                <td colspan="3"><IMG src="imgs/banner.gif"></td>
                                </tr>
                                <tr>
                                </tr>
                                <tr>
                                    <td width="50"></td>
                                    <td class="Normal"
vAlign="top"><br>
                                </tr>
                            </table>
                        <br>
                    </td>
                    <td colspan="2">
                        <b>Servicios Marcatel</b>
                        <br>
                        <br>
                        Utilice
                        nuestros servicios en Línea para consultar,
                        comprar o recargar sust
                        <br>
                        tarjetas de
                        Internet Prepago.
                    </td>
                    <td vAlign="top"
align="right"><IMG
src="imgs/foto_registro.gif"></td>
                </tr>
                <tr>
                    <td width="50"></td>
                    <td colspan="2" vAlign="top"
colSpan="2">
                        <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
                            <tr>
                                <td colspan="2" height="30"></td>
                                </tr>
                                <tr>
                                    <td colspan="2" class="NormalBold"
bgColor="#becdda" colSpan="2" height="20">..
Recarga de
                                </td>
                                <td colspan="2" height="30"></td>
                                </tr>
                            </table>
                        <br>
                    </td>
                </tr>
            </table>
        </form>
    </body>

```



```

<LINK href="Styles.css"
type="text/css" rel="stylesheet">
</HEAD>
<body leftMargin="0" topMargin="0">
<form id="Form1"
runat="server">
<%
If (not session("compra_efectuada")) then
%>
<table height="100%"
cellSpacing="0" cellPadding="0" width="100%"
border="0">
<tr>
<td vAlign="top" width="130"
background="imgs/barra.gif"><br>
<br>
<br>
<br>
<br>
<br>
<IMG src="imgs/clientes.gif">
</td>
<td align="left">
<table height="100%" cellSpacing="0"
cellPadding="0" border="0">
<tr>
<td vAlign="top"
colSpan="3"><IMG src="imgs/banner.gif"></td>
</tr>
<tr>
<td width="50"></td>
<td class="Normal"
vAlign="top">
<P><br>
<br>
<b>Compra &nbsp;de &nbsp;Tarjeta
prepago</b>&nbsp;
<br>
<br>
Para &nbsp;comprar su tarjeta de Internet
prepago MarcaNet, por favor

```

```

digite &nbsp;el Folio de la tarjeta Origen,
Nip y &nbsp;Folio de la tarjeta

Destino &nbsp;en la siguiente forma,
después presione el botón de recargar.

</P>
</td>
<td vAlign="top"
align="right"><IMG src="imgs/foto_registro.gif"></td>
</tr>
<tr>
<td width="50"></td>
<td vAlign="top"
colSpan="2">
<table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
<tr>
<td width="239" height="30"></td>
</tr>
<tr>
<td class="NormalBold"
bgColor="#becdda" colSpan="2" height="20">..
Información de
tarjeta de crédito:</td>
</tr>
<tr>
<td align="middle" colSpan="2"
height="30"><asp:label id="LblInfo" runat="server"
CssClass="error">LblInfo</asp:label><br>
</td>
</tr>
<tr>
<td colSpan="2" height="20"></td>

```

```

        </tr>

        <tr>

            <td class="Normal" width="239"
                height="23">Seleccione la tarjeta que desea

                    compar.&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>

                <td align="left"
                    height="23"><asp:dropdownlist id="cboProducto"
                        runat="server" Width="134px">

                            <asp:ListItem
                                Value="00" Selected="True">Eliga un
                                producto</asp:ListItem>

                            <asp:ListItem
                                Value="0">50</asp:ListItem>

                            <asp:ListItem
                                Value="1">100</asp:ListItem>

                            <asp:ListItem
                                Value="3">200</asp:ListItem>

                        </asp:dropdownlist><asp:customvalidator
                            id="ValProducto" runat="server" CssClass="error"
                            Width="240px" ErrorMessage="* Seleccione una
                            tarjeta prepagada" ControlToValidate="cboMes"
                            Display="Dynamic"
                            OnServerValidate="ValidaProducto"></asp:customva
                            lidator></td>

                    </tr>

                <tr>

                    <td colspan="2" height="20"></td>

                </tr>

                <td class="Normal" width="239">Tipo de
                    Tarjeta:</td>

                <td align="left"><asp:dropdownlist
                    id="cboTipoTarjeta"
                    Runat="Server"></asp:dropdownlist><asp:customval
                    idator id="ValTipoTarjeta" runat="server"

                        CssClass="error" ErrorMessage="* Seleccione un
                        tipo de tarjeta" ControlToValidate="cboTipoTarjeta"
                        Display="Dynamic"
                        OnServerValidate="ValidaTipoTarjeta"></asp:custom
                        validator></td>

                <tr>

                    <td colspan="2" height="20"></td>

                </tr>

                <td colspan="2" height="20"></td>

                <td class="Normal" width="239">Bancos
                    Autorizados :</td>

                <td align="left"><asp:dropdownlist
                    id="cboBancos"
                    Runat="Server"></asp:dropdownlist><asp:customval
                    idator id="ValBanco" runat="server"
                    CssClass="error" ErrorMessage="* Seleccione un
                    banco" ControlToValidate="cboTipoTarjeta"
                    Display="Dynamic"
                    OnServerValidate="ValidaBanco"></asp:customvalid
                    ator></td>

                </tr>

                <tr>

                    <td colspan="2" height="20"></td>

                </tr>

                <td colspan="2" height="20"></td>

                <td class="Normal" width="239">Número
                    de la tarjeta de crédito:</td>

                <td align="left"><asp:textbox
                    id="tarjetaCredito" Width="200"
                    Runat="server"></asp:textbox><asp:requiredfieldvali
                    dator id="ValFolioOrg" runat="server"
                    CssClass="Error" ErrorMessage=" * Favor de
                    introducir el número de tarjeta"
                    ControlToValidate="tarjetaCredito"
                    Display="Dynamic"></asp:requiredfieldvalidator></td>

                </tr>

```

<tr>	<tr>		
<td colspan="2" height="20"></td>	<td colspan="2" height="20"></td>		
</tr>	</tr>	</tr>	</tr>
<tr>	<tr>	<tr>	<tr>
<td class="Normal" width="239">Nombre completo:</td>	<td class="Normal" width="239"></td>	<td align="left"></td>	<td align="left"></td>
<td align="left"><asp:textbox id="nombre" Width="200" Runat="server"></asp:textbox><asp:requiredfieldvalidator id="ValNombre" runat="server" CssClass="Error" ErrorMessage=" * Favor de introducir el nombre completo" ControlToValidate="nombre" Display="Dynamic"></asp:requiredfieldvalidator></td>	<td align="middle" colspan="2" height="30"><asp:button id="Btncomprar" Runat="server" Text="Comprar"></asp:button></td>	</tr>	
</tr>	</tr>	</tr>	</tr>
<tr>	<tr>	<tr>	<tr>
<td colspan="2" height="20"></td>	<td colspan="2" height="20"></td>		
</tr>	</tr>	<td width="239" height="30"></td>	</tr>
<tr>	<tr>	<tr>	<tr>
<td class="Normal" width="239">Expiración de la tarjeta mm/yyyy :</td>	<td class="NormalBold" bgColor="#becdda" colspan="2" height="20">.</td>	</tr>	
<td align="left"><asp:dropdownlist id="cboMes" Runat="Server"></asp:dropdownlist> <asp:dropdownlist id="cboAnio" Runat="Server"></asp:dropdownlist>	</tr>	</table>	</td>
<asp:customvalidator id="ValMes" runat="server" CssClass="error" ErrorMessage="* Seleccione un mes" ControlToValidate="cboMes" Display="Dynamic" OnServerValidate="ValidaMes"></asp:customvalidator><asp:customvalidator id="ValAnio" runat="server" CssClass="error" ErrorMessage="* Seleccione un año" ControlToValidate="cboAnio" Display="Dynamic" OnServerValidate="ValidaAnio"></asp:customvalidator></td>	</tr>	<tr>	<td height="100%"></td>
</tr>	</tr>	</table>	</tr>
</tr>	</tr>	<td width="100%"></td>	</tr>

```

        </tr>
    </table>
    <%else
    %>
    <table height="100%"
cellSpacing="0" cellPadding="0" width="100%"
    <tr>
        <td vAlign="top" width="130"
background="imgs/barra.gif" height="204"><br>
        <br>
        <br>
        <br>
        <br>
        <br>
        <IMG src="imgs/clientes.gif">
        </td>
        <td align="left" height="204">
            <table height="100%" cellSpacing="0"
cellPadding="0" border="0">
                <tr>
                    <td vAlign="top"
colSpan="3"><IMG src="imgs/banner.gif"></td>
                    </tr>
                    <tr>
                        <td width="50"></td>
                        <td class="Normal"
vAlign="top" align="left" colSpan="2" height="100%">
                            <P><br>
                                Se realizó la compra de tarjeta
                                exitosamente, con los siguientes datos:
                                <br>
                                <br>
                                <b>Compra de Tarjeta
                                prepago</b>
                                <br>
                                <br>
                                Para comprar su tarjeta de Internet
                                prepago MarcaNet, por favor
                                digite el Folio de la tarjeta Origen,
                                Nip y Folio de la tarjeta
                                Destino en la siguiente forma,
                                después presione el botón de
                    </td>
                </tr>
            </table>
        </td>
    </tr>
    <td align="right"><IMG src="imgs/foto_registro.gif"></td>
    </tr>
    <tr>
        <td colspan="3" height="30"></td>
        <td colspan="3">
            <td width="50"></td>
            <td class="Normal"
align="middle" colSpan="2"><b>Compra de
Tarjeta de Internet
                prepago</b>
            </td>
        </tr>
        <tr>
            <td colspan="3" height="100%"></td>
            <td width="50"
vAlign="top" align="left" colSpan="2" height="100%">
                <P><br>
                    Se realizó la compra de tarjeta
                    exitosamente, con los siguientes datos:
                    <br>
                    <br>
                    <b>Compra de Tarjeta
                    prepago</b>
                    <br>
                    <br>
                    Para comprar su tarjeta de Internet
                    prepago MarcaNet, por favor
                    digite el Folio de la tarjeta Origen,
                    Nip y Folio de la tarjeta
                    Destino en la siguiente forma,
                    después presione el botón de
            </td>
        </tr>
    </table>
    <asp:label id="LblCantidad"
runat="server">Saldo de Tarjeta
: </asp:label>
    <asp:label id="LblProducto" runat="server"
CssClass="error"></asp:label>
    <P><b>FAVOR DE APUNTA
R</b>
S</P>

```



```

Protected WithEvents ValNombre As
System.Web.UI.WebControls.RequiredFieldValidator
Protected WithEvents tarjetaCredito As
System.Web.UI.WebControls.TextBox
Protected WithEvents cboProducto As
System.Web.UI.WebControls.DropDownList
Protected WithEvents ValProducto As
System.Web.UI.WebControls.CustomButton
Protected WithEvents cboMes As
System.Web.UI.WebControls.DropDownList
Protected WithEvents cboAnio As
System.Web.UI.WebControls.DropDownList
Protected WithEvents ValMes As
System.Web.UI.WebControls.CustomButton
Protected WithEvents Btncomprar As
System.Web.UI.WebControls.Button
Protected WithEvents LblCantidad As
System.Web.UI.WebControls.Label
Protected WithEvents LblProducto As
System.Web.UI.WebControls.Label
Protected WithEvents Lb1 As
System.Web.UI.WebControls.Label
Protected WithEvents LblUsuario As
System.Web.UI.WebControls.Label
Protected WithEvents LblDestinotxt As
System.Web.UI.WebControls.Label
Protected WithEvents lbpas As
System.Web.UI.WebControls.Label
Protected WithEvents LinkButton1 As
System.Web.UI.WebControls.LinkButton
Protected WithEvents Button1 As
System.Web.UI.WebControls.Button
Protected WithEvents Email As
System.Web.UI.WebControls.TextBox
Protected WithEvents ValEmail As
System.Web.UI.WebControls.RequiredFieldValidator
Protected WithEvents ValAnio As
System.Web.UI.WebControls.CustomButton

#Region " Web Form Designer Generated Code "

' This call is required by the Web Form Designer.
<System.Diagnostics.DebuggerStepThrough(>
Private Sub InitializeComponent()

End Sub

Private Sub Page_Init(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
'CODEGEN: This method call is required by the
Web Form Designer
'Do not modify it using the code editor.
InitializeComponent()
End Sub

#End Region

Private Sub Page_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
'Put user code to initialize the page here
LblInfo.Text = ""
If Not IsPostBack Then
LlenaList()
End If
If Session("compra_efectuada") Then
cargar_labels()
ValProducto.Enabled = False
ValTipoTarjeta.Enabled = False
ValBanco.Enabled = False
ValFolioOrg.Enabled = False
ValNombre.Enabled = False
ValMes.Enabled = False
ValAnio.Enabled = False

End If

End Sub

Private Sub LlenaList()
'Llena los combos de Banco y Fecha de
expiracion

cboTipoTarjeta.Items.Add(New ListItem("Eliga
tipo", "00"))
cboTipoTarjeta.Items.Add(New ListItem("Visa",
"01"))
cboTipoTarjeta.Items.Add(New
ListItem("MasterCard", "02"))

cboBancos.Items.Add(New ListItem("Eliga su
banco", "00"))
cboBancos.Items.Add(New ListItem("AFIRME",
"02"))
cboBancos.Items.Add(New ListItem("BAMSA /
BOFAMERICA", "04"))
cboBancos.Items.Add(New
ListItem("BANCRECER", "05"))
cboBancos.Items.Add(New
ListItem("BANJERCITO", "06"))
cboBancos.Items.Add(New ListItem("BANK
ONE", "07"))
cboBancos.Items.Add(New
ListItem("BANORTE", "08"))
cboBancos.Items.Add(New
ListItem("BANREGIO", "09"))
cboBancos.Items.Add(New ListItem("BANSI",
"10"))
cboBancos.Items.Add(New ListItem("BBVA
BANCOMER", "11"))
cboBancos.Items.Add(New ListItem("BITAL /
INTERNACIONAL", "12"))
cboBancos.Items.Add(New ListItem("BOSTON",
"13"))
cboBancos.Items.Add(New ListItem("DEL
BAJIO", "14"))
cboBancos.Items.Add(New ListItem("DEL
SURESTE", "15"))
cboBancos.Items.Add(New
ListItem("BANJERCITO", "06"))
cboBancos.Items.Add(New
ListItem("INBURSA", "16"))
cboBancos.Items.Add(New
ListItem("INDUSTRIAL", "17"))
cboBancos.Items.Add(New
ListItem("INTERACCIONES", "18"))
cboBancos.Items.Add(New ListItem("INVEX",
"19"))
cboBancos.Items.Add(New ListItem("IXE",
"20"))
cboBancos.Items.Add(New ListItem("MIFEL",
"21"))
cboBancos.Items.Add(New
ListItem("NAFINSA", "22"))
cboBancos.Items.Add(New
ListItem("SANTANDER MEXICANO", "23"))
cboBancos.Items.Add(New
ListItem("SCOTIABANK", "24"))

```

```

        cboBancos.Items.Add(New ListItem("SERFIN",
"25"))
        cboBancos.Items.Add(New
ListItem("BANAMEX", "26"))
        cboBancos.Items.Add(New
ListItem("CITIBANK", "27"))
        cboBancos.Items.Add(New ListItem("BANCO
AZTECA", "28"))

        cboMes.Items.Add(New ListItem("Mes", "00"))
        cboMes.Items.Add(New ListItem("Enero", "01"))
        cboMes.Items.Add(New ListItem("Febrero",
"02"))
        cboMes.Items.Add(New ListItem("Marzo", "03"))
        cboMes.Items.Add(New ListItem("Abril", "04"))
        cboMes.Items.Add(New ListItem("Mayo", "05"))
        cboMes.Items.Add(New ListItem("Junio", "06"))
        cboMes.Items.Add(New ListItem("Julio", "07"))
        cboMes.Items.Add(New ListItem("Agosto",
"08"))
        cboMes.Items.Add(New ListItem("Septiembre",
"09"))
        cboMes.Items.Add(New ListItem("Octubre",
"10"))
        cboMes.Items.Add(New ListItem("Noviembre",
"11"))
        cboMes.Items.Add(New ListItem("Diciembre",
"12"))

        cboAnio.Items.Add(New ListItem("Año", "00"))
        cboAnio.Items.Add(New ListItem("2003",
"2003"))
        cboAnio.Items.Add(New ListItem("2004",
"2004"))
        cboAnio.Items.Add(New ListItem("2005",
"2005"))
        cboAnio.Items.Add(New ListItem("2006",
"2006"))
        cboAnio.Items.Add(New ListItem("2007",
"2007"))
        cboAnio.Items.Add(New ListItem("2008",
"2008"))
        cboAnio.Items.Add(New ListItem("2009",
"2009"))
        cboAnio.Items.Add(New ListItem("2010",
"2010"))
        cboAnio.Items.Add(New ListItem("2011",
"2011"))
        cboAnio.Items.Add(New ListItem("2012",
"2012"))
        cboAnio.Items.Add(New ListItem("2013",
"2013"))
    End Sub

    Private Sub Btncomprar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Btncomprar.Click
        Dim tarjeta As New ObjTarjeta.Tarjeta()
        Dim login, pass, respuesta As String

        Try
            If (Not ValTipoTarjeta.IsValid) OrElse (Not
ValBanco.IsValid) OrElse (Not ValMes.IsValid)
OrElse (Not ValAnio.IsValid) OrElse (Not
ValProducto.IsValid) Then
                'Validacion de combos
                ElseIf tarjeta.comprarTarjeta(nombre.Text,
cboMes.SelectedItem.Value & "/" &
cboAnio.SelectedItem.Value,
CInt(cboTipoTarjeta.SelectedItem.Value),
CInt(cboProducto.SelectedItem.Text),
CInt(cboBancos.SelectedItem.Value),
tarjetaCredito.Text, respuesta,
CInt(cboProducto.SelectedItem.Text), login, pass)
Then
                    'Recarga la tarjeta y devuelve un True ->
Transaccion válida
                    Session("compra_efectuada") = True
                    Session("temp_login") = login
                    Session("temp_pass") = pass
                    Session("temp_producto") =
cboProducto.SelectedItem.Text

                    'Response.Redirect("Compras_resultado.aspx?produ
cto=" & cboProducto.SelectedItem.Text & "&login="
& login & "&pass=" & pass)

                    'Deshabilitar validaciones de los pagos
                    ValProducto.Enabled = False
                    ValTipoTarjeta.Enabled = False
                    ValBanco.Enabled = False
                    ValFolioOrg.Enabled = False
                    ValNombre.Enabled = False
                    ValMes.Enabled = False
                    ValAnio.Enabled = False
                    cargar_labels()
                Else
                    'Si la respuesta de la recarga es un false -
> Nó Valida
                    LblInfo.Text = respuesta
                    tarjeta.guardarLog(1, "", "", 0, " Error en la
compra de tarjeta ")
                    End If
                    Catch ex As Exception
                        Dim mail As New PrepagoWeb.Mail()
                        mail.mandarMail("Error en pagina
compra.aspx ." & ex.ToString)
                        tarjeta.guardarLog(1, "", "", 0, " Error en la
compra de tarjeta ")
                    End Try
                    End Sub

                    Public Sub cargar_labels()
                        If Session("compra_efectuada") Then
                            LblProducto.Text =
Session("temp_producto")
                            LblUsuario.Text = Session("temp_login")
                            lblpass.Text = Session("temp_pass")
                            Dim x, i As Short
                            x = Page.Validators.Count()
                            For i = 0 To x - 2

                                Page.Validators.Remove(Page.Validators(0))
                                    Next
                                End If
                                    Email.Text = "tucorreo@mail.com"
                                End Sub

                                #Region "Validacion de los Combos
                                CustomValidator"
                                    Sub ValidaTipoTarjeta(ByVal source As Object,
                                    ByVal args As ServerValidateEventArgs)
                                        args.IsValid =
(cboTipoTarjeta.SelectedItem.Value <> "00")
                                    End Sub
                                        Sub ValidaBanco(ByVal source As Object, ByVal
                                        args As ServerValidateEventArgs)
                                            args.IsValid = (cboBancos.SelectedItem.Value
<> "00")
                                        End Sub
                                End Sub

```

```

Sub ValidaMes(ByVal source As Object, ByVal
args As ServerValidateEventArgs)
    args.IsValid = (cboMes.SelectedItem.Value <>
"00")
End Sub
Sub ValidaAnio(ByVal source As Object, ByVal
args As ServerValidateEventArgs)
    args.IsValid = (cboAnio.SelectedItem.Value <>
"00")
End Sub
Sub ValidaProducto(ByVal source As Object,
ByVal args As ServerValidateEventArgs)
    args.IsValid = (cboProducto.SelectedItem.Value
<> "00")
End Sub
#End Region

```

```

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim Mymail As New Mail()
    Dim msg As New StringBuilder()
    msg.Append("***Este es un mensaje generado
automaticamente por el sistema*** & CrLf)
    msg.Append("***Favor de no responder*** &
CrLf)
    msg.Append(CrLf)
    msg.Append(CrLf)
    msg.Append("Ha adquirido una tarjeta con la
siguiente informacion:" & CrLf)
    msg.Append(CrLf)
    msg.Append(CrLf)
    msg.Append("Saldo de Tarjeta: " &
LblProducto.Text & CrLf)
    msg.Append("Nombre de Usuario: " &
LblUsuario.Text & CrLf)
    msg.Append("Contraseña: " & lblpass.Text &
CrLf)
    Mymail.mandarMail_cliente(Email.Text,
msg.ToString)
    compra_efectuada = True
End Sub
End Class

```

Resultado de la Compra ASP

```

<%@ Page Language="vb"
AutoEventWireup="false"
Codebehind="Compras_resultado.aspx.vb"
Inherits="PrepagoWeb.Compras_resultado" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>Marcatel
International</title>
        <meta http-equiv="Content-
Type" content="text/html; charset=iso-8859-1">

```

```

        <LINK href="Styles.css"
type="text/css" rel="stylesheet">
    </HEAD>
    <body leftMargin="0" topMargin="0">
        <form runat="server">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
                <tr>
                    <td vAlign="top" width="130"
background="imgs/barra.gif" height="204"><br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <br>
                    <IMG src="imgs/clientes.gif">
                    </td>
                    <td align="left" height="204">
                        <table height="100%" cellSpacing="0"
cellPadding="0" border="0">
                            <tr>
                                <td vAlign="top"
colSpan="3"><IMG src="imgs/banner.gif"></td>
                                </tr>
                                <tr>
                                    <td width="50"></td>
                                    <td class="Normal"
vAlign="top" width="265">
                                        <P><br>
                                        <br>
                                        <b>Compra&nbsp;de&nbsp;Tarjeta
prepago</b>&nbsp;&nbsp;&nbsp;<br>
                                        <br>
                                        Para&nbsp;comprar su tarjeta de
Internet prepago MarcaNet, por favor
                                        digite&nbsp;el Folio de la tarjeta
Origen, Nip y&nbsp;Folio de la tarjeta
                                        Destino&nbsp;en la siguiente forma,
después presione el botón de

```



```

        <asp:linkbutton      id="LinkButton1"
runat="server"
CssClass="error">Aquí</asp:linkbutton></P>

        </td>

    </tr>

</table>

</td>

<td width="100%" height="204"></td>
</tr>
</table>
</form>
</body>
</HTML>

```

Resultado de la Compra Code Behind

```

Public Class Compras_resultado
    Inherits System.Web.UI.Page
    Protected WithEvents LblDestino As
System.Web.UI.WebControls.Label
    Protected WithEvents LinkButton1 As
System.Web.UI.WebControls.LinkButton
    Protected WithEvents LinkButton2 As
System.Web.UI.WebControls.LinkButton
    Protected WithEvents lblpass As
System.Web.UI.WebControls.Label
    Protected WithEvents LblCantidad As
System.Web.UI.WebControls.Label
    Protected WithEvents LblProducto As
System.Web.UI.WebControls.Label
    Protected WithEvents Lb1 As
System.Web.UI.WebControls.Label
    Protected WithEvents LblUsuario As
System.Web.UI.WebControls.Label
    Protected WithEvents LblDestinoxt As
System.Web.UI.WebControls.Label

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()>
    Private Sub InitializeComponent()

        End Sub

    Private Sub Page_Init(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Init
        'CODEGEN: This method call is required by the
Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'Put user code to initialize the page here
        If Not IsPostBack Then
            LblProducto.Text =
Request.QueryString("producto")

```

```

        LblUsuario.Text =
Request.QueryString("login")
        lblpass.Text = Request.QueryString("pass")
        End If
        End Sub

    Private Sub LinkButton1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles LinkButton1.Click
        Response.Redirect("consulta.aspx")
    End Sub

    Private Sub LinkButton2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles LinkButton2.Click
        lblpass.Text = Request.QueryString("pass")
        If LinkButton2.Text = "Presiona aqui para ver tu
contraseña" Then
            LinkButton2.Text = "Presiona aqui para
esconder tu contraseña"
            lblpass.Visible = True
        Else
            LinkButton2.Text = "Presiona aqui para ver tu
contraseña"
            lblpass.Visible = False
        End If
    End Sub

End Class

```

Consulta de Saldo ASP

```

<%@      Page      Language="vb"
AutoEventWireup="false"
Codebehind="Consulta.aspx.vb"
Inherits="PrepagoWeb.Consulta" %>
<HTML>
    <HEAD>
        <title>Marcatel
International</title>
        <meta http-equiv="Content-
Type" content="text/html; charset=iso-8859-1">
        <LINK href="Styles.css"
type="text/css" rel="stylesheet">
    </HEAD>
    <body leftMargin="0" topMargin="0">
        <form runat="server">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
                <tr>
                    <td vAlign="top" width="130"
background="imgs/barra.gif"><br>
                    <br>
                    <br>
                    <br>

```

```

        <br>
        <a href="consulta.aspx"><IMG
src="imgs/consulta.jpg"
onMouseOver="Javascript:
src='imgs/consulta_up.jpg';"
onMouseOut="Javascript:
src='imgs/consulta.jpg';"></a>
        <br>
        <a href="compras.aspx"><IMG
src="imgs/compra.jpg"
onMouseOver="Javascript:
src='imgs/compra_up.jpg';"
onMouseOut="Javascript:
src='imgs/compra.jpg';"></a>
        <br>
        <a href="recarga.aspx"><IMG
src="imgs/recarga.jpg"
onMouseOver="Javascript:
src='imgs/recarga_up.jpg';"
onMouseOut="Javascript:
src='imgs/recarga.jpg';"></a>
    </td>
    <td align="left">
        <table height="100%" cellSpacing="0"
cellPadding="0" border="0">
            <tr>
                <td colspan="3"><IMG src="imgs/banner.gif"></td>
            </tr>
            <tr>
                <td width="50"></td>
                <td class="Normal"
vAlign="top"><br>
                    <br>
                    <b>Consulta de Saldo</b>
                    <br>
                    <br>
                    Para
consultar el saldo de su tarjeta de Prepago
DialUp MarcaNet, porfavor
                    <br>
                    digite su
Folio y Nip en la siguiente forma, después
presione el botón de
                    <br>
                    consultar.
                </td>
            </tr>
        </table>
    </td>
    <td align="right"><IMG
src="imgs/foto_registro.gif"></td>
    </tr>
    <tr>
        <td colspan="2">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">
                <tr>
                    <td width="200" height="30"></td>
                    <td colspan="2" class="NormalBold"
bgColor="#becdda" colSpan="2" height="20">:.
Información de
                        la tarjeta</td>
                </tr>
                <tr>
                    <td colspan="2">
                        <td colspan="2" class="Normal"
height="30"><asp:label
id="lblvalidacion"
runat="server"
Width="593px"></asp:label></td>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <td class="Normal" width="200">Folio
de la tarjeta :</td>
                    </td>
                </tr>
                <tr>
                    <td align="left"><asp:textbox id="folio"
Width="200"
Runat="server"></asp:textbox><asp:requiredfiel
dvalidator id="ValConsulta" runat="server"
CssClass="Error" ErrorMessage=" * Favor de
introducir el folio"
ControlToValidate="folio"></asp:requiredfieldvali
dator></td>
                </tr>
            </table>
        </td>
    </tr>

```



```

InitializeComponent()
End Sub
#End Region
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    lblvalidacion.Text = ""
End Sub
Private Sub BtnConsultar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnConsultar.Click
    Dim tarjeta As New ObjTarjeta.Tarjeta()
    Try
        If (tarjeta.validarTarjeta(folio.Text, pin.Text))
        Then
            Dim saldo As Decimal
            Dim fecha_exp As DateTime
            tarjeta.consultarTarjeta(folio.Text, saldo, fecha_exp)
            Response.Redirect("Consulta_resultado.aspx" & "?saldo=" & Trim(CStr(saldo)) & "&fechaexp=" & Trim(CStr(fecha_exp)), False)
        Else
            lblvalidacion.Text = "Usuario o contraseña no válida, intentar de nuevo."
            tarjeta.guardarLog(1, folio.Text, "", 0, "Error en consulta de tarjeta " & folio.Text)
        End If
        Catch ex As Exception
            Dim mail As New PrepagoWeb.Mail()
            mail.mandarMail("Error en pagina consulta.aspx :" & ex.ToString)
            tarjeta.guardarLog(1, folio.Text, "", 0, "Error en consulta de tarjeta " & folio.Text)
        End Try
    End Sub
End Class

Resultado de Consulta de Saldo ASP

<%@ Page Language="vb"
AutoEventWireup="false"
Codebehind="Consulta_resultado.aspx.vb"
Inherits="PrepagoWeb.Template" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>Marcatel
International</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <LINK href="Styles.css" type="text/css" rel="stylesheet">
    </HEAD>
    <body leftMargin="0" topMargin="0">
        <form runat="server">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="756" border="0">
                <tr>
                    <td vAlign="top" width="130"
background="imgs/barra.gif" height="204"><br>
                    <td vAlign="top"
align="right"><IMG
src="imgs/foto_registro.gif"></td>
                </tr>
            </table>
        </form>
    </body>
</HTML>

```



```

Private Sub Page_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    lblsaldo.Text = "$" &
Request.QueryString("Saldo")
    lblFecExp.Text =
Request.QueryString("fechaexp")
End Sub

```

```
End Class
```

Traspaso de Saldo de Tarjeta a Tarjeta ASP

```

<%@ Page Language="vb"
AutoEventWireup="false"
Codebehind="Recarga.aspx.vb"
Inherits="PrepagoWeb.Recarga" %>
<HTML>

```

```

    <HEAD>
        <title>Marcatel
International</title>
        <meta http-equiv="Content-
Type" content="text/html; charset=iso-8859-1">
        <LINK href="Styles.css"
type="text/css" rel="stylesheet">
    </HEAD>

```

```

    <body leftMargin="0" topMargin="0">
        <form runat="server">
            <table
height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">

```

```

                <tr>
                    <td vAlign="top" width="130"
background="imgs/barra.gif"><br>

```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<br>
```

```
<IMG src="imgs/clientes.gif">
```

```
</td>
```

```
<td align="left">
```

```

        <table height="100%" cellSpacing="0"
cellPadding="0" border="0">

```

```

            <tr>

```

```

                <td vAlign="top"
colSpan="3"><IMG src="imgs/banner.gif"></td>

```

```

            </tr>

```

```

            <tr>

```

```

                <td width="50"></td>

```

```

                <td class="Normal"
vAlign="top">

```

```
<P><br>
```

```
<br>
```

```

        <b>Recarga&nbsp;de&nbsp;Tarjeta
prepago</b>&nbsp;

```

```
<br>
```

```
<br>
```

```

        Para recargar su tarjeta de Internet
prepago MarcaNet, porfavor

```

```

        digite&nbsp;el Folio de la tarjeta
Origen, Nip y&nbsp;Folio de la tarjeta

```

```

        Destino&nbsp;en la siguiente forma,
después presione el botón de recargar.

```

```
</P>
```

```
</td>
```

```
<td vAlign="top"
```

```

align="right"><IMG
src="imgs/foto_registro.gif"></td>

```

```
</tr>
```

```
<tr>
```

```
<td width="50"></td>
```

```
<td vAlign="top"
```

```
colSpan="2">
```

```

        <table

```

```

height="100%" cellSpacing="0" cellPadding="0"
width="100%" border="0">

```

```
<tr>
```

```
<td height="30"></td>
```

```
</tr>
```

```
<tr>
```

```

        <td class="NormalBold"
bgColor="#becdda" colSpan="2" height="20">..
Información de

```

```

        la tarjetas</td>

```

```
</tr>
```

```

        <tr>
            <td align="middle" colspan="2"
                height="30"><asp:label
                id="LblInfo"
                runat="server"
                CssClass="error">LblInfo</asp:label></td>
        </tr>
        <tr>
            <td class="Normal" width="200">Folio
            de la tarjeta Destino:</td>
            <td align="left"><asp:textbox
                id="folioDestino" Width="200" Runat="server"
                MaxLength="10"></asp:textbox><asp:requiredfie
                ldvalidator id="ValDolioDes" runat="server"
                CssClass="Error"
                ControlToValidate="folioDestino"
                ErrorMessage=" * Favor de introducir el
                folio"></asp:requiredfieldvalidator></td>
        </tr>
        <tr>
            <td class="Normal" width="200">Folio
            de la tarjeta Origen:</td>
            <td align="left"><asp:textbox
                id="folioOrigen" Width="200" Runat="server"
                MaxLength="10"></asp:textbox><asp:requiredfie
                ldvalidator id="ValFolioOrg" runat="server"
                CssClass="Error"
                ControlToValidate="folioOrigen" ErrorMessage="
                * Favor de introducir el
                folio"></asp:requiredfieldvalidator></td>
        </tr>
        <tr>
            <td colspan="2" height="20"></td>
            <td align="left"><asp:button
                id="BtnRecargar" Runat="server"
                Text="Recargar"></asp:button></td>
        </tr>
        <tr>
            <td class="Normal" width="200">Nip
            :</td>
            <td align="middle" colspan="2"
                height="30"></td>
        </tr>
        <tr>
            <td align="left"><asp:textbox id="nip"
                Width="200" Runat="server"
                TextMode="Password"
                MaxLength="10"></asp:textbox><asp:requiredfie
                ldvalidator id="Valpassword" runat="server"
                CssClass="Error" ControlToValidate="nip"
                ErrorMessage=" * Favor de introducir su
                Nip"></asp:requiredfieldvalidator></td>
            <td colspan="2" height="30"></td>
        </tr>
        <tr>
            <td colspan="2" height="20"></td>
            <td align="left" class="NormalBold"
                bgColor="#becdda"
                height="20">:</td>
        </tr>
    
```

```

</tr>
</table>
</td>
</tr>
<tr>
<td
height="100%"></td>
</tr>
</table>
</td>
<td width="100%"></td>
</tr>
</table>
</form>
</body>
</HTML>

```

Traspaso de Saldo de Tarjeta a Tarjeta Code Behind

```

Imports System.Threading

Public Class Recarga
    Inherits System.Web.UI.Page
    Protected WithEvents folioOrigen As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValFolioOrg As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents nip As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents Valpassword As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents folioDestino As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValDolioDes As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents BtnRecargar As
    System.Web.UI.WebControls.Button
    Protected WithEvents LblInfo As
    System.Web.UI.WebControls.Label

    #Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()>
    Private Sub InitializeComponent()

        End Sub

    Private Sub Page_Init(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Init
        'CODEGEN: This method call is required by the
    Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

```

```

Private Sub Page_Load(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    LblInfo.Text = ""
End Sub

Private Sub BtnRecargar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles BtnRecargar.Click
    Dim Tarjeta As New ObjTarjeta.Tarjeta()
    Dim MiSaldo As Decimal

    Try
        If Tarjeta.obtenerSaldo(folioOrigen.Text) <= 0
        Then
            LblInfo.Text = "La tarjeta no tiene saldo."
            ElseIf folioOrigen.Text = folioDestino.Text
            Then
                LblInfo.Text = "Porfavor seleccione una
                tarjeta de destino diferente a la de origen."
                ElseIf Tarjeta.validarTarjeta(folioOrigen.Text,
                nip.Text) AndAlso
                Tarjeta.obtenerSaldo(folioOrigen.Text) > 0 Then
                    MiSaldo =
                    Tarjeta.obtenerSaldo(folioOrigen.Text)
                    Tarjeta.recargarTarjeta(folioOrigen.Text,
                    folioDestino.Text)

                    Response.Redirect("recarga_resultado.aspx?origen="
                    & folioOrigen.Text & "&destino=" &
                    folioDestino.Text & "&monto=" & Misaldo, False)
                Else
                    LblInfo.Text = "La tarjeta y/o Nip
                    incorrecto(a)."
                    Tarjeta.guardarLog(1, folioOrigen.Text,
                    folioDestino.Text, 0, "Error en traspaso de tarjeta de
                    " & folioOrigen.Text & " a " & folioDestino.Text)
                End If
                Catch ex As Exception
                    Dim mymail As New Mail()
                    mymail.mandarMail("Error en pagina
                    Recarga.aspx : " & ex.ToString)
                    Tarjeta.guardarLog(1, folioOrigen.Text,
                    folioDestino.Text, 0, "Error en traspaso de tarjeta de
                    " & folioOrigen.Text & " a " & folioDestino.Text)
                    mymail = Nothing
                    Tarjeta = Nothing
                End Try
            End Sub

        End Class

```

Resultado de Traspaso de Saldo ASP

```

<%@ Page Language="vb"
AutoEventWireup="false"
Codebehind="Recarga_resultado.aspx.vb"
Inherits="PrepagoWeb.Recarga_resultado" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>Marcatel
        International</title>
        <meta http-equiv="Content-
        Type" content="text/html; charset=iso-8859-1">
        <LINK href="Styles.css"
        type="text/css" rel="stylesheet">
    </HEAD>

```



```

</tr>

<tr>

<td class="Normal" width="200">Cantidad
a recargar (mínimo 100 pesos):</td>

<td align="left"><asp:textbox id="monto"
Width="200" Runat="server"
MaxLength="7"></asp:textbox><asp:requiredfieldvali
dator id="ValMonto" runat="server" CssClass="Error"
Display="Dynamic" ControlToValidate="monto"
ErrorMessage=" * Favor de introducir la
cantidad"></asp:requiredfieldvalidator><asp:regularex
pressionvalidator id="ValCantidad" runat="server"
CssClass="error" Display="Dynamic"
ControlToValidate="monto" ErrorMessage="*
Cantidad no válida"
ValidationExpression="\d+(\.d{2})?"></asp:regularex
pressionvalidator></td>

</tr>

<tr>

<td colspan="2" height="20"></td>

</tr>

<tr>

<td class="Normal" width="200">Tipo de
Tarjeta</td>

<td align="left"><asp:dropdownlist
id="cboTipoTarjeta"
Runat="Server"></asp:dropdownlist><asp:customval
idator id="ValTipoTarjeta" runat="server"
CssClass="error" Display="Dynamic"
ControlToValidate="cboTipoTarjeta"
ErrorMessage="* Seleccione un tipo de tarjeta"
OnServerValidate="ValidaTipoTarjeta"></asp:custom
validator></td>

</tr>

<tr>

<td colspan="2" height="20"></td>

</tr>

<tr>

<td class="Normal" width="200">Bancos
Autorizados :</td>

<td align="left"><asp:dropdownlist
id="cboBancos"
Runat="Server"></asp:dropdownlist><asp:customval
idator id="ValBanco" runat="server"
CssClass="error" Display="Dynamic"
ControlToValidate="cboTipoTarjeta"
ErrorMessage="* Seleccione un banco"
OnServerValidate="ValidaBanco"></asp:customvalid
ator></td>

</tr>

<tr>

<td colspan="2" height="20"></td>

</tr>

<tr>

<td class="Normal" width="200">Número
de la tarjeta de crédito:</td>

<td align="left"><asp:textbox
id="tarjetaCredito" Width="200" Runat="server"
MaxLength="16"></asp:textbox><asp:requiredfieldva
lidator id="ValFolioOrg" runat="server"
CssClass="Error" Display="Dynamic"
ControlToValidate="tarjetaCredito" ErrorMessage="*
Favor de introducir el número de
tarjeta"></asp:requiredfieldvalidator></td>

</tr>

<tr>

<td colspan="2" height="20"></td>

</tr>

<tr>

<td class="Normal" width="200">Nombre
completo:</td>

<td align="left"><asp:textbox id="nombre"
Width="200" Runat="server"
MaxLength="50"></asp:textbox><asp:requiredfieldva
lidator id="ValNombre" runat="server"
CssClass="Error" Display="Dynamic"

```



```

        height="100%"></td>
    </tr>
</table>
</td>
<td width="100%"></td>
</tr>
</table>
</form>
</body>
</HTML>

```

Recarga de Saldo con Tarjeta de Crédito Code Behind

```

Imports System.Data.SqlClient

Public Class RecargaTC
    Inherits System.Web.UI.Page
    Protected WithEvents lblInfo As
    System.Web.UI.WebControls.Label
    Protected WithEvents folioOrigen As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValFolioOrg As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents folio As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValFolio As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents monto As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValMonto As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents ValCantidad As
    System.Web.UI.WebControls.RegularExpressionValidator
    Protected WithEvents cboTipoTarjeta As
    System.Web.UI.WebControls.DropDownList
    Protected WithEvents cboBancos As
    System.Web.UI.WebControls.DropDownList
    Protected WithEvents cboMes As
    System.Web.UI.WebControls.DropDownList
    Protected WithEvents cboAnio As
    System.Web.UI.WebControls.DropDownList
    Protected WithEvents ValTipoTarjeta As
    System.Web.UI.WebControls.CustomValidator
    Protected WithEvents ValBanco As
    System.Web.UI.WebControls.CustomValidator
    Protected WithEvents nombre As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents ValNombre As
    System.Web.UI.WebControls.RequiredFieldValidator
    Protected WithEvents ValAnio As
    System.Web.UI.WebControls.CustomValidator
    Protected WithEvents ValMes As
    System.Web.UI.WebControls.CustomValidator
    Protected WithEvents tarjetaCredito As
    System.Web.UI.WebControls.TextBox
    Protected WithEvents LinkPoliticAs As
    System.Web.UI.WebControls.LinkButton
    Protected WithEvents checkPoliticAs As
    System.Web.UI.WebControls.CheckBox
    Protected WithEvents lblIntentos As
    System.Web.UI.WebControls.Label

```

```

    Protected WithEvents BtnRecargar As
    System.Web.UI.WebControls.Button

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()>
    Private Sub InitializeComponent()

        End Sub

    Private Sub Page_Init(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Init
        'CODEGEN: This method call is required by the
    Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private Sub Page_Load(ByVal sender As
    System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        'Put user code to initialize the page here
        lblInfo.Text = ""
        lblIntentos.Text = ""
        If Not IsPostBack Then
            LlenerLista()
        End If
        'Intentos()
    End Sub

    Private Sub Intentos()
        Dim MyFunctions As New Functions()

        Select Case
myfunctions.obtenerIntentos(nombre.Text,
tarjetaCredito.Text)
            Case 0
                lblIntentos.Text = "PRIMER INTEN
T O"
            Case 1
                lblIntentos.Text = "SEGUNDO INT
ENTO"
            Case 2
                lblIntentos.Text = "TERCER INTE
NT O"
            Case 3
                lblIntentos.Text = "USUARIO BLO
QUEADO"
        End Select
        myfunctions = Nothing
    End Sub

    Private Sub LlenerLista()
        'Llena los combos de Banco y Fecha de
    expiracion

        cboTipoTarjeta.Items.Add(New ListItem("Eliga
    tipo", "00"))
        cboTipoTarjeta.Items.Add(New ListItem("Visa",
    "01"))
        cboTipoTarjeta.Items.Add(New
    ListItem("MasterCard", "02"))

        cboBancos.Items.Add(New ListItem("Eliga su
    banco", "00"))

```

```

        cboBancos.Items.Add(New ListItem("AFIRME",
"02"))
        cboBancos.Items.Add(New ListItem("BAMSA /
BOFAMERICA", "04"))
        cboBancos.Items.Add(New
ListItem("BANCRECER", "05"))
        cboBancos.Items.Add(New
ListItem("BANJERCITO", "06"))
        cboBancos.Items.Add(New ListItem("BANK
ONE", "07"))
        cboBancos.Items.Add(New
ListItem("BANORTE", "08"))
        cboBancos.Items.Add(New
ListItem("BANREGIO", "09"))
        cboBancos.Items.Add(New ListItem("BANSI",
"10"))
        cboBancos.Items.Add(New ListItem("BBVA
BANCOMER", "11"))
        cboBancos.Items.Add(New ListItem("BITAL /
INTERNACIONAL", "12"))
        cboBancos.Items.Add(New ListItem("BOSTON",
"13"))
        cboBancos.Items.Add(New ListItem("DEL
BAJIO", "14"))
        cboBancos.Items.Add(New ListItem("DEL
SURESTE", "15"))
        cboBancos.Items.Add(New
ListItem("BANJERCITO", "06"))
        cboBancos.Items.Add(New
ListItem("INBURSA", "16"))
        cboBancos.Items.Add(New
ListItem("INDUSTRIAL", "17"))
        cboBancos.Items.Add(New
ListItem("INTERACCIONES", "18"))
        cboBancos.Items.Add(New ListItem("INVEX",
"19"))
        cboBancos.Items.Add(New ListItem("IXE",
"20"))
        cboBancos.Items.Add(New ListItem("MIFEL",
"21"))
        cboBancos.Items.Add(New
ListItem("NAFINSA", "22"))
        cboBancos.Items.Add(New
ListItem("SANTANDER MEXICANO", "23"))
        cboBancos.Items.Add(New
ListItem("SCOTIABANK", "24"))
        cboBancos.Items.Add(New ListItem("SERFIN",
"25"))
        cboBancos.Items.Add(New
ListItem("BANAMEX", "26"))
        cboBancos.Items.Add(New
ListItem("CITIBANK", "27"))
        cboBancos.Items.Add(New ListItem("BANCO
AZTECA", "28"))

        cboMes.Items.Add(New ListItem("Mes", "00"))
        cboMes.Items.Add(New ListItem("Enero", "01"))
        cboMes.Items.Add(New ListItem("Febrero",
"02"))
        cboMes.Items.Add(New ListItem("Marzo", "03"))
        cboMes.Items.Add(New ListItem("Abril", "04"))
        cboMes.Items.Add(New ListItem("Mayo", "05"))
        cboMes.Items.Add(New ListItem("Junio", "06"))
        cboMes.Items.Add(New ListItem("Julio", "07"))
        cboMes.Items.Add(New ListItem("Agosto",
"08"))
        cboMes.Items.Add(New ListItem("Septiembre",
"09"))
        cboMes.Items.Add(New ListItem("Octubre",
"10"))

        cboMes.Items.Add(New ListItem("Noviembre",
"11"))
        cboMes.Items.Add(New ListItem("Diciembre",
"12"))

        cboAnio.Items.Add(New ListItem("Año", "00"))
        cboAnio.Items.Add(New ListItem("2003",
"2003"))
        cboAnio.Items.Add(New ListItem("2004",
"2004"))
        cboAnio.Items.Add(New ListItem("2005",
"2005"))
        cboAnio.Items.Add(New ListItem("2006",
"2006"))
        cboAnio.Items.Add(New ListItem("2007",
"2007"))
        cboAnio.Items.Add(New ListItem("2008",
"2008"))
        cboAnio.Items.Add(New ListItem("2009",
"2009"))
        cboAnio.Items.Add(New ListItem("2010",
"2010"))
        cboAnio.Items.Add(New ListItem("2011",
"2011"))
        cboAnio.Items.Add(New ListItem("2012",
"2012"))
        cboAnio.Items.Add(New ListItem("2013",
"2013"))
        End Sub

Private Sub BtnRecargar_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles BtnRecargar.Click
    Dim Tarjeta As New ObjTarjeta.Tarjeta()
    Dim respuesta As String
    Dim MyFunctions As New Functions()

    Try

        If (Not ValTipoTarjeta.IsValid) OrElse (Not
ValBanco.IsValid) OrElse (Not ValMes.IsValid)
OrElse (Not ValAnio.IsValid) Then
            'Validacion de combos
            Elseif
MyFunctions.obtenerIntentos(nombre.Text,
tarjetaCredito.Text) > 2 Then
                'Valida que el usuario no este bloqueado
por intentos fallidos
                LblInfo.Text = "El usuario ha sido
bloqueado temporalmente(24 hrs.) debido al número
de intentos fallidos."
                Elseif Not Tarjeta.validarTarjeta(folio.Text)
Then
                    'El folio no existe
                    LblInfo.Text = "El folio de tarjeta prepago
no existe."
                    Elseif Not (CDec(monto.Text) >= 100
AndAlso CDec(monto.Text) <= 2500) Then
                        'La cantidad cubra el rango de 100 a 2500
                        LblInfo.Text = "La cantidad no cubre el
límite mínimo requerido."
                        Elseif Not checkPolíticas.Checked Then
                            'El clienet no esta de acuerdo con las
políticas
                            LblInfo.Text = "Favor de leer las políticas y
marcar el checkbox una vez que este de acuerdo
con ellas."
                            Elseif Tarjeta.recargarTarjeta(nombre.Text,
folio.Text, cboMes.SelectedItem.Value & "/" &
cboAnio.SelectedItem.Value,

```

```

CInt(cboTipoTarjeta.SelectedItem.Value),
CDec(monto.Text),
CInt(cboBancos.SelectedItem.Value),
tarjetaCredito.Text, respuesta) Then
    'Recarga la tarjeta y devuelve un True ->
    Transaccion válida

Response.Redirect("recargaTC_resultado.aspx?mon
to=" & monto.Text & "&folio=" & folio.Text, False)
Else
    'Si la respuesta de la recarga es un false -
    > Nó Válida
    LblInfo.Text = respuesta
    Tarjeta.guardarLog(1, folio.Text, "",
monto.Text, "Error en la recarga de tarjeta " &
folio.Text & " con tarjeta de crédito " &
tarjetaCredito.Text & " por " & monto.Text)
End If
Catch ex As Exception
    Tarjeta.guardarLog(1, folio.Text, "",
monto.Text, "Error en la recarga de tarjeta " &
folio.Text & " con tarjeta de crédito " &
tarjetaCredito.Text & " por " & monto.Text)
    LblInfo.Text = "La transacción no pudo ser
procesada, favor de intentar más tarde."
End Try
End Sub

#Region "Validacion de los Combos
CustomValidator"
Sub ValidaTipoTarjeta(ByVal source As Object,
ByVal args As ServerValidateEventArgs)
    args.IsValid =
(cboTipoTarjeta.SelectedItem.Value <> "00")
End Sub
Sub ValidaBanco(ByVal source As Object, ByVal
args As ServerValidateEventArgs)
    args.IsValid = (cboBancos.SelectedItem.Value
<> "00")
End Sub
Sub ValidaMes(ByVal source As Object, ByVal
args As ServerValidateEventArgs)
    args.IsValid = (cboMes.SelectedItem.Value <>
"00")
End Sub
Sub ValidaAnio(ByVal source As Object, ByVal
args As ServerValidateEventArgs)
    args.IsValid = (cboAnio.SelectedItem.Value <>
"00")
End Sub
#End Region

End Class

Resultado Recarga de Saldo con Tarjeta de
Crédito ASP

<%@
    Page
    Language="vb"
    AutoEventWireup="false"
    Codebehind="recargaTC_resultado.aspx.vb"
    Inherits="PrepagoWeb.recargaTC_resultado" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<HTML>
    <HEAD>
        <title>Marcatel
        International</title>
        <meta http-equiv="Content-
        Type" content="text/html; charset=iso-8859-1">

```

```

    <LINK href="Styles.css"
    type="text/css" rel="stylesheet">
</HEAD>
<body leftMargin="0" topMargin="0">
    <form runat="server"
    ID="Form1">
        <table
        height="100%" cellSpacing="0" cellPadding="0"
        width="100%" border="0">
            <tr>
                <td vAlign="top" width="130"
                background="imgs/barra.gif" height="204"><br>
                <br>
                <br>
                <br>
                <br>
                <br>
                <IMG src="imgs/clientes.gif">
            </td>
            <td align="left" height="204">
                <table height="100%" cellSpacing="0"
                cellPadding="0" border="0">
                    <tr>
                        <td vAlign="top"
                        colSpan="3"><IMG src="imgs/banner.gif"></td>
                    </tr>
                    <tr>
                        <td width="50"></td>
                        <td class="Normal"
                        vAlign="top" width="265">
                            <P><br>
                            <br>
                            <b>Recarga&nbsp;de&nbsp;Tarjeta
                            prepago</b>&nbsp;
                            <br>
                            <br>
                            Para recargar su tarjeta de Internet
                            prepago MarcaNet, porfavor
                            digite&nbsp;el Folio de la tarjeta
                            Origen, Nip y&nbsp;Folio de la tarjeta

```



```

End Sub
#End Region

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'Put user code to initialize the page here
    LblDestino.Text = Request.QueryString("folio")
    LblMonto.Text = Request.QueryString("monto")
End Sub

Private Sub LinkButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LinkButton1.Click
    Response.Redirect("consulta.aspx")
End Sub

End Class

Clase Tarjeta

Imports System.Data.SqlClient
Imports System.Web.Mail

Public Class Tarjeta

#Region " Metodos y funciones de la tarjeta"

Public Sub recargarTarjeta(ByVal folioOrigen As String, ByVal folioDestino As String)
    Dim myConnection As SqlConnection
    Dim Dcant_traspaso As String
    Try
        Dcant_traspaso = obtenerSaldo(folioOrigen)
        myConnection = ConectarBD()
        myConnection.Open()
        Dim myCommand As SqlCommand = New SqlCommand("Sp_UpdRecargaTarjetaTarjeta", myConnection)
        myCommand.Parameters.Add(New SqlParameter("@folioOrigen", SqlDbType.VarChar)).Value = folioOrigen
        myCommand.Parameters.Add(New SqlParameter("@folioDestino", SqlDbType.VarChar)).Value = folioDestino
        myCommand.CommandType = CommandType.StoredProcedure
        myCommand.ExecuteNonQuery()
        myConnection.Close()
        guardarLog(2, folioOrigen, folioDestino, Dcant_traspaso, "Traspaso de " & folioOrigen & " a " & folioDestino & " por " & Dcant_traspaso)
        Catch e As Exception
            mandarMail("Error en la recarga de Tarjeta a tarjeta: " & e.ToString)
        End Try
    End Sub

Public Function recargarTarjeta(ByVal nombre As String, ByVal folio As String, ByVal fecExp As String, ByVal tipoTarjeta As Int16, ByVal monto As Decimal, ByVal banco As Int16, ByVal tarjetaCredito As String, ByVal respuesta As String) As Boolean
    Dim consecutivo As Int32
    Dim APIPayWorks As New ConectorPayWorks()

    consecutivo = insertarConsecutivo(folio, monto, tipoTarjeta, banco, nombre, tarjetaCredito)
    APIPayWorks = ConectorPayWorks.EfectuarCargo(tarjetaCredito, fecExp, consecutivo, monto, "00000", "177.77.3.66:80")
    If Not APIPayWorks.EstatusPago = 0 Then
        GuardarPago(consecutivo, APIPayWorks.CodigoAutorizacionProcesador, APIPayWorks.MensajeRetornoProcesador, APIPayWorks.TiempoInicioTransaccion, APIPayWorks.TiempoFinTransaccion)
    End If
    If APIPayWorks.CodigoAutorizacionProcesador > 0 Then
        Dim myConnection As SqlConnection

        Try
            myConnection = ConectarBD()
            myConnection.Open()
            Dim myCommand As SqlCommand = New SqlCommand("Sp_UpdSaldo", myConnection)
            myCommand.Parameters.Add(New SqlParameter("@folio", SqlDbType.VarChar)).Value = folio
            myCommand.Parameters.Add(New SqlParameter("@saldo", SqlDbType.Money)).Value = monto
            myCommand.CommandType = CommandType.StoredProcedure
            myCommand.ExecuteNonQuery()
            myConnection.Close()
            guardarLog(3, folio, "", monto, "Recarga de tarjeta " & folio & " con tarjeta de crédito " & tarjetaCredito & " por " & monto)
            Return True
        Catch e As Exception
            mandarMail("Error al recargar el saldo de Tarjeta con TC : " & e.ToString)
            Throw New Exception(e.ToString)
            Return False
        End Try
    End If
    Else
        GuardarPago(consecutivo, -1, APIPayWorks.ToString, Now(), Now())
        If APIPayWorks.TextoError = "" Then
            respuesta = "Por el momento el servicio no se encuentra disponible."
        Else
            respuesta = "Tarjeta inválida/expirada, favor de verificar su tarjeta."
        End If
        Return False
    End If
    Catch ex As Exception
        GuardarPago(consecutivo, -1, "Error: tarjeta de crédito inválida - " & ex.ToString, Now(), Now())
        mandarMail("Error al comprar tarjeta: " & ex.ToString)
        respuesta = "La tarjeta de crédito no se pudo procesar, favor de verificar los datos e intentar de nuevo."
    End Try
    Return False
End Function

Public Function validarTarjeta(ByVal folio As String) As Boolean
    Dim myConnection As SqlConnection

```

```

Dim MiValida As Boolean

Try
    myConnection = ConectarBD()
    myConnection.Open()
    Dim myCommand As SqlCommand = New
SqlCommand("Sp_SelFolio", myConnection)
    myCommand.Parameters.Add(New
SqlParameter("@folio", SqlDbType.VarChar)).Value
= folio
    myCommand.CommandType =
CommandType.StoredProcedure
    MiValida = myCommand.ExecuteScalar()
    myConnection.Close()
    Return MiValida

Catch e As Exception
    mandarMail("Error en Validacion de Folio: " &
e.ToString)
    Return False
End Try
End Function

Public Function validarTarjeta(ByVal folio As
String, ByVal Nip As String) As Boolean
    Dim myConnection As SqlConnection
    Dim MiValida As Boolean

    Try
        myConnection = ConectarBD()
        myConnection.Open()
        Dim myCommand As SqlCommand = New
SqlCommand("Sp_SelFolioNip", myConnection)
        myCommand.Parameters.Add(New
SqlParameter("@folio", SqlDbType.VarChar)).Value
= folio
        myCommand.Parameters.Add(New
SqlParameter("@nip", SqlDbType.VarChar)).Value =
Nip
        myCommand.CommandType =
CommandType.StoredProcedure
        MiValida = myCommand.ExecuteScalar()
        myConnection.Close()
        Return MiValida

    Catch e As Exception
        mandarMail("Error en Validacion de Folio: " &
e.ToString)
        Return False
    End Try
End Function

Public Sub consultarTarjeta(ByVal folio As String,
ByRef saldo As Decimal, ByRef fecExpiracion As
DateTime)
    Dim myConnection As SqlConnection
    Dim MyReader As SqlDataReader
    Try
        myConnection = ConectarBD()
        myConnection.Open()
        Dim myCommand As SqlCommand = New
SqlCommand("Sp_SelInfoTarjeta", myConnection)
        myCommand.Parameters.Add(New
SqlParameter("@folio", SqlDbType.VarChar)).Value
= folio
        myCommand.CommandType =
CommandType.StoredProcedure
        MyReader = myCommand.ExecuteReader()
        Try
            If MyReader.Read() Then
                saldo = MyReader.GetValue(0)
                fecExpiracion = MyReader.GetValue(1)
            Else
                saldo = 0
                fecExpiracion = "1/1/1900"
            End If
            guardarLog(5, folio, "", 0, "Consulta de
tarjeta " & folio)
            MyReader = Nothing
            myConnection.Close()
            Catch e As Exception
                mandarMail("Error al Consultar tarjeta -
datos DB: " & e.ToString)
            End Try
            Catch e As Exception
                mandarMail("Error al Consultar tarjeta: " &
e.ToString)
            End Try
        End Sub

    Public Function obtenerSaldo(ByVal folio As
String) As Decimal
        Dim myConnection As SqlConnection
        Dim MiSaldo As String

        Try
            myConnection = ConectarBD()
            myConnection.Open()
            Dim myCommand As SqlCommand = New
SqlCommand("Sp_SelSaldo", myConnection)
            myCommand.Parameters.Add(New
SqlParameter("@folio", SqlDbType.VarChar)).Value
= folio
            myCommand.CommandType =
CommandType.StoredProcedure
            MiSaldo = myCommand.ExecuteScalar()
            myConnection.Close()
            Return MiSaldo

        Catch e As Exception
            mandarMail("Error al Consultar tarjeta - datos
DB: " & e.ToString)
            Return 0
        End Try
    End Function

    Public Function comprarTarjeta(ByVal nombre As
String, ByVal fecExp As String, ByVal tipoTarjeta As
Int16, ByVal tipoMonto As Int16, ByVal banco As
Int16, ByVal tarjetaCredito As String, ByRef
respuesta As String, ByVal denominacion As
Decimal, ByRef login As String, ByRef pass As
String) As Boolean
        Dim consecutivo As Int32
        Dim APIPayWorks As New
ConectorPayWorks()
        Dim monto As Decimal

        'Buscar si existe el tipo de tarjeta y devuelve el
monto si hay disponible
        If Not (chechar_inventario(denominacion)) Then
            respuesta = "No hay tarejtas disponibles de
$" & CStr(denominacion)
            mandarMail("Tarjetas para venta en internet
con denominacion: " & denominacion & " estan
agotadas")
            Return False
        End If

```

```

Try
    consecutivo =
insertarConsecutivo("0000000000", 0.0, tipoTarjeta,
banco, nombre, tarjetaCredito)
    APIPayWorks =
ConectorPayWorks.EfectuarCargo(tarjetaCredito,
fecExp, consecutivo, denominacion, "00000",
"177.77.3.66:80")
    If Not APIPayWorks.EstatusPago = 0 Then
        GuardarPago(consecutivo,
APIPayWorks.CodigoAutorizacionProcesador,
APIPayWorks.MensajeRetornoProcesador,
APIPayWorks.TiempoInicioTransaccion,
APIPayWorks.TiempoFinTransaccion)
        If
APIPayWorks.CodigoAutorizacionProcesador > 0
Then
            Dim myConnection As SqlConnection

                Try
                    'Se obtiene la tarjeta disponible y se
modifica el estatus
                    obtener_tarjeta(denominacion, login,
pass)

                    actualizar_estatus_tarj(login)
                    guardarLog(4, login, "",
denominacion, "Compra de Tarjeta " & login & " por "
& denominacion)

                        Return True
                    Catch e As Exception
                        mandarMail("Error al comprar tarjeta
con Tarjeta con TC : " & e.ToString)
                        Return False
                    End Try
                End If
            Else
                GuardarPago(consecutivo, -1,
APIPayWorks.ToString, Now(), Now())
                If APIPayWorks.TextoError = "" Then
                    respuesta = "Por el momento el servicio
no se encuentra disponible."
                Else
                    respuesta = "Tarjeta inválida/expirada,
favor de verificar su tarjeta."
                End If
                Return False
            End If
        Catch ex As Exception
            GuardarPago(consecutivo, -1, "Error: tarjeta
de crédito inválida - " & ex.ToString, Now(), Now())
            mandarMail("Error al comprar tarjeta: " &
ex.ToString)
            respuesta = "La tarjeta de crédito no se pudo
procesar, favor de verificar los datos e intentar de
nuevo."
            Return False
        End Try
    End Function

#End Region

#Region "Sub metodos y funciones"

Private Function ConectarBD() As SqlConnection

    Return New SqlConnection("user
id=prepago;password=xdcc;initial
catalog=Prepago;data source=177.77.3.77;")
End Function

Public Sub guardarLog(ByVal tipo As Int16, ByVal
folioOrigen As String, ByVal folioDestino As String,
ByVal monto As Double, ByVal Descripcion As
String)
    Dim myConnection As SqlConnection
    Try
        myConnection = ConectarBD()
        myConnection.Open()
        Dim myCommand As SqlCommand = New
SqlCommand("Sp_InsLog", myConnection)
        myCommand.Parameters.Add(New
SqlParameter("@Tipo", SqlDbType.SmallInt)).Value
= tipo
        myCommand.Parameters.Add(New
SqlParameter("@folioOrigen",
SqlDbType.VarChar)).Value = folioOrigen
        myCommand.Parameters.Add(New
SqlParameter("@folioDestino",
SqlDbType.VarChar)).Value = folioDestino
        myCommand.Parameters.Add(New
SqlParameter("@monto", SqlDbType.Money)).Value
= monto
        myCommand.Parameters.Add(New
SqlParameter("@Descripcion",
SqlDbType.VarChar)).Value = Descripcion
        myCommand.CommandType =
CommandType.StoredProcedure
        myCommand.ExecuteNonQuery()
        myConnection.Close()

    Catch e As Exception
        mandarMail("Error al guardar Log: " &
e.ToString)
    End Try
End Sub

Private Sub mandarMail(ByVal msg As String)
    'Envial el email con la lista de conciliacion como
Attachment
    Dim myMail As New MailMessage()

    Dim myBody As String
    Try
        myBody = New String("")
        SmtMail.SmtServer = "MTELMY16"
        myMail.From = "Error@marcatel.net"
        myMail.To = "forta@marcatel.net"
        myMail.Subject = "Error en el objeto Tarjeta"
        myMail.Priority = MailPriority.High
        myMail.Body = msg
        SmtMail.Send(myMail)
    Catch ex As Exception
        Throw New Exception("Error al enviar mail")
    End Try
End Sub

Private Function insertarConsecutivo(ByVal folio
As String, ByVal cantidad As Decimal, ByVal
tipoTarjeta As Int16, ByVal banco As Int16, ByVal
nombre As String, ByVal tarjetaCredito As String) As
Int32
    Dim myConnection As SqlConnection =
ConectarBD()

    Try

```

```

        myConnection.Open()
        Dim myCommand As New
    SqlCommand("Sp_InsConsecutivo", myConnection)
        Dim NoTarjeta As String
        Dim consecutivo As Int32

        NoTarjeta = Mid(tarjetaCredito, 1, 4)
        NoTarjeta = NoTarjeta & Mid(tarjetaCredito,
13, 4)

        myCommand.CommandType =
    CommandType.StoredProcedure
        myCommand.Parameters.Add(New
    SqlParameter("@nombre",
    SqlDbType.VarChar)).Value = nombre
        myCommand.Parameters.Add(New
    SqlParameter("@cantidad",
    SqlDbType.Money)).Value = cantidad
        myCommand.Parameters.Add(New
    SqlParameter("@tarjetaCredito",
    SqlDbType.VarChar)).Value = CStr(NoTarjeta)
        myCommand.Parameters.Add(New
    SqlParameter("@folio", SqlDbType.VarChar)).Value
    = folio
        myCommand.Parameters.Add(New
    SqlParameter("@tipoTarjeta",
    SqlDbType.SmallInt)).Value = tipoTarjeta
        myCommand.Parameters.Add(New
    SqlParameter("@banco",
    SqlDbType.SmallInt)).Value = banco
        consecutivo = myCommand.ExecuteScalar()
        myConnection.Close()
        Return consecutivo
    Catch ex As Exception
        mandarMail("Error al obtener consecutivo: "
& ex.ToString)
        Throw New Exception(ex.ToString)
        Return 0
    End Try
    End Function

    Private Sub GuardarPago(ByVal consecutivo As
    Int32, ByVal referencia As Int32, ByVal respuesta As
    String, ByVal feclnicio As DateTime, ByVal fecFinal
    As DateTime)
        Dim myConnection As SqlConnection =
    ConectarBD()

        Try
            myConnection.Open()
            Dim myCommand As New
    SqlCommand("Sp_UpdPago", myConnection)

            myCommand.CommandType =
    CommandType.StoredProcedure
            myCommand.Parameters.Add(New
    SqlParameter("@consecutivo",
    SqlDbType.BigInt)).Value = consecutivo
            myCommand.Parameters.Add(New
    SqlParameter("@referencia",
    SqlDbType.BigInt)).Value = referencia
            myCommand.Parameters.Add(New
    SqlParameter("@respuesta",
    SqlDbType.VarChar)).Value = respuesta
            myCommand.Parameters.Add(New
    SqlParameter("@feclni",
    SqlDbType.DateTime)).Value = feclnicio
            myCommand.Parameters.Add(New
    SqlParameter("@fecFin",
    SqlDbType.DateTime)).Value = fecFinal

            myCommand.ExecuteNonQuery()
            myConnection.Close()

            Catch ex As Exception
                mandarMail("Error al grabar el pago: " &
    ex.ToString)
            End Try
        End Sub

        ' 1. actualizar estatus de las tarjetas
        Private Sub actualizar_estatus_tarj(ByVal folio As
    String)
            Dim myConnection As SqlConnection =
    ConectarBD()

            Try
                myConnection.Open()
                Dim myCommand As New
    SqlCommand("Sp_UpdEstatusTarj", myConnection)

                myCommand.CommandType =
    CommandType.StoredProcedure
                myCommand.Parameters.Add(New
    SqlParameter("@folio", SqlDbType.VarChar)).Value
    = folio
                myCommand.ExecuteNonQuery()
                myConnection.Close()
                Catch ex As Exception
                    mandarMail("Error al grabar el pago: " &
    ex.ToString)
                End Try
            End Sub

            '--2. checar existencia de tarjetas para compra
            Private Function checar_inventario(ByVal
    denominacion As Integer) As Boolean
                Dim myconnection As SqlConnection
                Dim cantidad As Long
                Try
                    myconnection = ConectarBD()
                    myconnection.Open()
                    Dim myCommand As SqlCommand = New
    SqlCommand("Sp_checa_tarj_disp", myconnection)
                    myCommand.Parameters.Add(New
    SqlParameter("@denominacion",
    SqlDbType.VarChar)).Value = denominacion
                    myCommand.CommandType =
    CommandType.StoredProcedure
                    cantidad = myCommand.ExecuteScalar()
                    myconnection.Close()

                    If cantidad = 0 Then
                        Return False
                    Else
                        Return True
                    End If

                    Catch e As Exception
                        mandarMail("Error al checar denominacion
    en existencia : " & e.ToString)
                        Throw New Exception(e.ToString)
                        Return False
                    End Try
                End Function

            '--3. regresar tarjeta disponible
            Private Sub obtener_tarjeta(ByVal denominacion
    As Integer, ByRef login As String, ByRef pass As
    String)
                Dim myconnection As SqlConnection

```

```

Dim MyReader As SqlDataReader

Try
    myconnection = ConectarBD()
    myconnection.Open()
    Dim myCommand As SqlCommand = New
SqlCommand("Sp_SelTarjDisp", myconnection)
    myCommand.Parameters.Add(New
SqlParameter("@denominacion",
SqlDbType.VarChar)).Value = denominacion
    myCommand.CommandType =
CommandType.StoredProcedure
    MyReader = myCommand.ExecuteReader()

    If MyReader.Read() Then
        login = MyReader.GetValue(0)
        pass = MyReader.GetValue(1)
    End If
    myconnection.Close()

Catch e As Exception
    mandarMail("Error al checar denominacion
en existencia : " & e.ToString)
    Throw New Exception(e.ToString)
End Try

End Sub
#End Region

End Class

Código Monitoreo de Saldos, Servidor

Imports System
Imports System.Data.SqlClient
Imports System.Net.Sockets
Imports System.Net
Imports System.Threading

Public Class Server

    Private myListener As TcpListener
    Private port As Int16 = 2003
    Private MySocket As Socket
    Dim thas As Thread

    Public Sub Server()
        Try
            myListener = New TcpListener(port)
            myListener.Start()
            thas = New Thread(New
ThreadStart(AddressOf ThAcceptSocket))
            thas.Start()
        Catch e As Exception
            Console.WriteLine("Error al iniciar el
servidor:" & e.ToString())
        End Try
    End Sub

    Private Sub ThAcceptSocket()
        Do While True
            MySocket = myListener.AcceptSocket
            Dim th As Thread = New Thread(New
ThreadStart(AddressOf StartListen))
            th.Start()
        Loop
    End Sub

    Private Sub StartListen()
        Dim TmpSocket As Socket = MySocket
        Dim Timeout As Int16 = 0
        Do While True And Timeout < 3
            Try
                If TmpSocket.Connected Then
                    Dim receive(64) As Byte
                    Dim i As Int32 =
TmpSocket.Receive(receive, receive.Length, 0)
                    Dim unwanted() As Char = {"", " ", " "}
                    Dim rece As String =
System.Text.Encoding.ASCII.GetString(receive)
                    aceptarComando(rece, TmpSocket)
                    If i > 0 Then
                        Exit Do
                    Else
                        Thread.Sleep(1000)
                    End If
                    Timeout = Timeout + 1
                Else
                    Exit Do
                End If
                TmpSocket.Close()
            Catch e As Exception
            End Try
        Loop
    End Sub

    Public Sub Server_Pause()
        myListener.Stop()
    End Sub

    Public Sub Server_Stop()
        If Not myListener Is Nothing Then
            myListener.Stop()
            If Not thas Is Nothing Then thas.Abort()
        End Sub

    Public Sub Server_Start()
        myListener.Start()
    End Sub

    Private Sub aceptarComando(ByVal paquete As
String, ByVal mysocket As Socket)
        Dim comando As String = New
String(Mid(paquete, 1, 3))
        Dim parametro As String = New
String(Mid(paquete, 5, Len(paquete) - 4))
        Dim respuesta As String
        Dim flg_cmd As Boolean

        Select Case comando
            Case "MIN"
                flg_cmd = True
                respuesta = "MIN " &
consultarInfo(parametro)
            Case "INF"
                'Devolver Info de la tarjeta
                flg_cmd = True
            Case Else
                flg_cmd = False
        End Select
        Try
            If flg_cmd Then
                Dim byteDateLine() As Byte =
Text.Encoding.ASCII.GetBytes(respuesta.ToCharArray())
                mysocket.Send(byteDateLine,
byteDateLine.Length, 0)
            End If
        End Try
    End Sub

```

```

    End If
    Catch e As Exception
    End Try
End Sub

Private Function consultarInfo(ByVal lp As String)
As String
    Dim myConnection As SqlConnection
    Dim informacion As String
    Dim MyReader As SqlDataReader
    Try
        myConnection = ConectarBD()
        myConnection.Open()
        Dim myCommand As SqlCommand = New
SqlCommand("Sp_SellInfoPlugin", myConnection)
        myCommand.Parameters.Add(New
SqlParameter("@IP", SqlDbType.VarChar)).Value =
lp
        myCommand.CommandType =
CommandType.StoredProcedure
        MyReader = myCommand.ExecuteReader
    Try
        If MyReader.Read() Then
            informacion = MyReader.GetValue(0) &
", " & MyReader.GetValue(1) & ", " &
MyReader.GetValue(2) & ", " &
MyReader.GetValue(3)
        End If
        MyReader = Nothing
        myConnection.Close()
        Return informacion
    Catch e As Exception
        Return ""
    End Try
    Catch e As Exception
        Return ""
    End Try
End Function

Private Function ConectarBD() As SqlConnection
    Return New SqlConnection("user
id=prepago;password=xdcc;initial
catalog=Prepago;data source=177.77.3.77;")
End Function

End Class

```

Código Monitoreo de Saldos, Cliente

```

Imports System
Imports System.Net.Sockets
Imports System.Net
Imports System.Threading

Public Class Cliente
    Private name As String
    Private mySocket As Socket
    Private port As Int16 = 2003
    Private readData As Boolean = False
    Public nts As NetworkStream
    Private tcpc As MyTcpClient

    Private Sub aceptarComando(ByVal paquete As
String, ByVal nts As NetworkStream, ByVal tarjeta
As System.Windows.Forms.Label, ByVal saldo As
System.Windows.Forms.Label, ByVal minutos As
System.Windows.Forms.Label, ByVal expiracion As
System.Windows.Forms.Label, ByVal mylcon As
NotifyIcon)

```

```

        Dim comando As String = New
String(Mid(paquete, 1, 3))
        Dim parametro As String = New
String(Mid(paquete, 5, Len(paquete) - 4))
        Dim respuesta As String
        Dim flg_cmd As Boolean

        Select Case comando
            Case "MIN"
                flg_cmd = False
                mostrarInfo(True, parametro, tarjeta, saldo,
minutos, expiracion, mylcon)
            Case "INF"
                'Devolver Info de la tarjeta
                flg_cmd = False
            Case Else
                flg_cmd = False
        End Select
        If flg_cmd Then
            Try
                If nts.CanWrite Then
                    Dim sender As String = respuesta
                    Dim sends() As Byte =
System.Text.Encoding.ASCII.GetBytes(sender.ToCh
arArray())
                    nts.Write(sends, 0, sends.Length)
                    nts.Flush()
                End If
            Catch e As Exception
            End Try
        End If
    End Sub

```

```

    Private Sub mostrarInfo(ByVal opcion As Boolean,
ByVal info As String, ByVal tarjeta As
System.Windows.Forms.Label, ByVal saldo As
System.Windows.Forms.Label, ByVal minutos As
System.Windows.Forms.Label, ByVal expiracion As
System.Windows.Forms.Label, ByVal mylcon As
NotifyIcon)
        If opcion Then
            Dim StringArr() As String = Split(info, ",")
            tarjeta.Text = StringArr(0).ToString
            saldo.Text = StringArr(1).ToString
            minutos.Text = StringArr(2).ToString & "
minuto(s)"
            expiracion.Text = StringArr(3).ToString
            mylcon.Text = "Tiempo restante : " &
minutos.Text & " minuto(s)"
        Else
            tarjeta.Text = "Desconectado"
            saldo.Text = ""
            minutos.Text = ""
            expiracion.Text = ""
            mylcon.Text = "Desconectado"
        End If
    End Sub

```

```

    Public Sub obtenerInfo(ByVal tarjeta As
System.Windows.Forms.Label, ByVal saldo As
System.Windows.Forms.Label, ByVal minutos As
System.Windows.Forms.Label, ByVal expiracion As
System.Windows.Forms.Label, ByVal mylcon As
NotifyIcon, ByVal status As StatusBar)
        Dim readData As Boolean = False
        Dim Timeout As Int16 = 0
        Try
            status.Panels.Item(1).Text = "Conectando..."
            tcpc = New MyTcpClient("177.77.3.77", port)
            If tcpc.IsConnected Then

```

```

nts = tcpc.GetStream()
mylcon.Icon = New Icon("m.ico")
status.Panels.Item(1).Text = "Conectado"
If nts.CanWrite Then
    Dim oAddr As System.Net.IPAddress
    Dim sAddr As String
    oAddr = New
System.Net.IPAddress(System.Net.Dns.GetHostByName(System.Net.Dns.GetHostName()).AddressList(0).Address)
    sAddr = oAddr.ToString
    Dim sender As String = "MIN " & sAddr
    Dim sends() As Byte =
System.Text.Encoding.ASCII.GetBytes(sender.ToArray())
    nts.Write(sends, 0, sends.Length)
    nts.Flush()
End If
Do While (Not readData And nts.CanRead)
And Timeout < 5
    If nts.DataAvailable Then
        Dim rcd(128) As Byte
        Dim i As Int32 = nts.Read(rcd, 0, 128)
        Dim ree As String =
System.Text.Encoding.ASCII.GetString(rcd)
        Dim unwanted() As Char = {" ", " ", " ", " "}
        aceptarComando(ree, nts, tarjeta,
saldo, minutos, expiracion, mylcon)
        readData = True
    End If
    Thread.Sleep(1000)
    Timeout = Timeout + 1
Loop
tcpc.Close()
nts.Close()
End If
Catch e As Exception
    mylcon.Icon = New Icon("mOff.ico")
    mostrarInfo(False, "", tarjeta, saldo, minutos,
expiracion, mylcon)
    status.Panels.Item(1).Text = "Desconectado"
    If Not (tcpc Is Nothing) Then tcpc.Close()
End Try
End Sub

End Class

Public Class MyTcpClient
    Inherits TcpClient

    Public Sub New(ByVal host As String, ByVal port
As Integer)
        Me.Connect(host, port)
    End Sub

    Public Function IsConnected() As Boolean
        Return Me.Client.Connected
    End Function
End Class

```

Modulo de tarificación

Tarificación.vb

```

Imports System.Data.SqlClient

Public Class Tarificacion

    'Variables de la clase
    Private numActividad As Int16 = 1

```

```

Private secsProceso As Int32

    Public Sub tarificaUsuarios(ByVal control As
System.Windows.Forms.TextBox, ByVal
fechaProceso As DateTime)

*****
*****
        'Metodo principal cuyas funciones principales
son:
        '-Inicializar la lista la lista de usuarios a
tarificar(Solo la primera vez - al inicio)
        '-Insertar los usuario recién conectados
        '-Actualizar las tarifas de los usuarios en la lista
de Usuarios a Tarificar
        '-Tarificar (Insertar al listado de transacciones
diarias)
        '-Cobrar los minutos consumidos
        '-Remover los usuarios que se desconectaron
        '-Validar los saldos y desconectar usuarios sin
saldo (Interfase plataforma)

*****
*****

        Try
            'Obtiene los usuarios conectados en el ultimo
periodo de tiempo y los agrega a la lista de Usuarios
a Tarificar
            insertarUsuariosConectados(fechaProceso)

            'Actualiza tarifas de los usuarios de la lista de
Usuarios a Tarificar
            actualizarTarifas()

            'Inserta los usuarios de la lista de Usuarios a
Tarificar en las lista de Actividad Diaria
            tarificar()

            'Recorre la lista de Usuarios a Tarificar,
descontando el saldo por minuto de cada tarjeta
            cobrarSaldos(control)

            'Obtiene los usuarios desconectados en el
ultimo periodo de tiempo y los borra de la lista de
Usuarios a Tarificar
            removerUsuariosdesconectados(control)

            'Valida cada usuario de la lista de Usuarios a
Tarificar, desconectando los usuarios cuyo saldo no
sea suficiente para el cobro de minuto
            validaSaldos(control)

            Catch ex As Exception
                Dim errorMsg As New ErrorCs()
                errorMsg.SendError("Error al tarificar
usuarios : " & ex.ToString)
            End Try
        End Sub

```

#Region "Pasos de tarificación"

```

'Listado de procesos de tarificación

    Public Sub inicializarLista(ByVal fechaProceso As
DateTime)
        'Se conecta al Servidor TNT mediante el
protocolo SMNP, para obtener los usuarios
conectados
        Dim obj_SNMP As New
Plataforma_SNMP_TNT_class()

```

```

Dim sesiones(.) As String
Dim tamArreglo As Int16 = 0

Try
    fechaProceso = Now()

obj_SNMP.obtener_Usuarios("200.56.224.25")
    sesiones = obj_SNMP.listadoSesiones

    depuraListaSesiones(sesiones)

    Dim connStr As SqlConnection
    connStr = conectarBD()
    connStr.Open()
    Dim cmdRadius As SqlCommand = New
SqlCommand("Sp_DelUsuariosIniciales", connStr)
    cmdRadius.CommandType =
CommandType.StoredProcedure
    cmdRadius.ExecuteNonQuery()
    cmdRadius.CommandText =
"Sp_InsUsuariosIniciales"

    For tamArreglo = 0 To sesiones.GetLength(1)
- 1
        If sesiones(1, tamArreglo) <> "" Then
            cmdRadius.Parameters.Add(New
SqlParameter("@sessionID",
SqlDbType.VarChar)).Value = sesiones(0,
tamArreglo)
            cmdRadius.Parameters.Add(New
SqlParameter("@userName",
SqlDbType.VarChar)).Value = sesiones(1,
tamArreglo)
            cmdRadius.Parameters.Add(New
SqlParameter("@origen",
SqlDbType.VarChar)).Value = sesiones(2,
tamArreglo)
            cmdRadius.Parameters.Add(New
SqlParameter("@destino",
SqlDbType.VarChar)).Value = sesiones(3,
tamArreglo)
            cmdRadius.ExecuteNonQuery()
            cmdRadius.Parameters.Clear()
        End If
        Next
        connStr.Close()
        Catch ex As Exception
            Dim errorMsg As New ErrorCs()
            errorMsg.SendError("Error al Insertar
usuarios Iniciales: " & ex.ToString)
        End Try
    End Sub

    Private Sub insertarUsuariosConectados(ByRef
fechaProceso As DateTime)
        Dim connStrRadius, connStr As SqlConnection
        Dim sqlReader As SqlDataReader

        Try
            connStrRadius = conectarBDRadius()
            connStr = conectarBD()
            connStrRadius.Open()
            connStr.Open()

            'Inserta los usuarios recién conectados, a la
lista de usuarios a tarificar ( Acct_Status_Id=1 )
            Dim cmdRadius As SqlCommand = New
SqlCommand("Sp_SelUsuariosConectados",
connStrRadius)

            cmdRadius.CommandType =
CommandType.StoredProcedure
            secsProceso = DateDiff(DateInterval.Second,
fechaProceso, Now())
            fechaProceso = Now()
            cmdRadius.Parameters.Add(New
SqlParameter("@secsProceso",
SqlDbType.BigInt)).Value = secsProceso
            sqlReader = cmdRadius.ExecuteReader
            Dim cmdPrepago As SqlCommand = New
SqlCommand("Sp_InsUsuariosConectadosRadius",
connStr)
            cmdPrepago.CommandType =
CommandType.StoredProcedure
            While sqlReader.Read
                cmdPrepago.Parameters.Add(New
SqlParameter("@SessionId",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(0)
                cmdPrepago.Parameters.Add(New
SqlParameter("@UserName",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(1)
                cmdPrepago.Parameters.Add(New
SqlParameter("@CallingId",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(3)
                cmdPrepago.Parameters.Add(New
SqlParameter("@CalledId",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(4)
                cmdPrepago.Parameters.Add(New
SqlParameter("@CalledId",
SqlDbType.VarChar)).Value = "1111111111"
                cmdPrepago.Parameters.Add(New
SqlParameter("@LoggedAt",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(2)
                cmdPrepago.ExecuteNonQuery()
                cmdPrepago.Parameters.Clear()
            End While
            sqlReader.Close()
            connStrRadius.Close()
            connStr.Close()
        Catch ex As Exception
            Throw New Exception("Error al insertar
usuarios conectados de Radius: " & ex.ToString)
        End Try
    End Sub

    Private Sub actualizarTarifas()
        Dim connStrUsuarios, connStrTarifas As
SqlConnection
        Dim sqlReader As SqlDataReader
        Dim horaTarifa As String

        Try
            connStrUsuarios = conectarBD()
            connStrUsuarios.Open()

            'Obtiene los registros de los usuarios
conectados a ser tarificados
            Dim cmdUsuarios As SqlCommand = New
SqlCommand("Sp_SelUsuariosConectados",
connStrUsuarios)
            cmdUsuarios.CommandType =
CommandType.StoredProcedure
            sqlReader = cmdUsuarios.ExecuteReader
            connStrTarifas = conectarBD()
            connStrTarifas.Open()

```

```

'Store procedure que actualiza la tarifa de
cada usuario
Dim cmdTarifas As SqlCommand = New
SqlCommand("Sp_Updtarifas", connStrTarifas)
Do While sqlReader.Read()
    cmdTarifas.Parameters.Clear()
    cmdTarifas.CommandType =
CommandType.StoredProcedure
    cmdTarifas.Parameters.Add(New
SqlParameter("@areaDestino",
SqlDbType.VarChar)).Value =
sqlReader.GetValue(3)
    cmdTarifas.Parameters.Add(New
SqlParameter("@login", SqlDbType.VarChar)).Value
= Trim(sqlReader.GetValue(0))
    cmdTarifas.Parameters.Add(New
SqlParameter("@sessionID",
SqlDbType.VarChar)).Value =
Trim(sqlReader.GetValue(1))
    cmdTarifas.ExecuteNonQuery()
Loop
connStrTarifas.Close()
connStrUsuarios.Close()

Catch ex As Exception
    Throw New Exception("Error al actualizar
tarifas: " & ex.ToString)
End Try
End Sub

Private Sub tarificar()
'Llena el log de actividades diarias

Dim connStrUsuarios, connStrActividades As
SqlConnection
Dim sqlReader As SqlDataReader

Try
    connStrUsuarios = conectarBD()
    connStrUsuarios.Open()

'Obtiene los registros de los usuarios
conectados a ser tarificados
Dim cmdUsuarios As SqlCommand = New
SqlCommand("Sp_SelUsuariosConectados",
connStrUsuarios)
    cmdUsuarios.CommandType =
CommandType.StoredProcedure
    sqlReader = cmdUsuarios.ExecuteReader
    connStrActividades = conectarBD()
    connStrActividades.Open()

'Store procedure que inserta cada transacion
el ActividadDiaria
Dim cmdActividades As SqlCommand = New
SqlCommand("Sp_InsActividadesDiarias",
connStrActividades)
Do While sqlReader.Read()
    cmdActividades.Parameters.Clear()
    cmdActividades.CommandType =
CommandType.StoredProcedure

        cmdActividades.Parameters.Add(New
SqlParameter("@userName",
SqlDbType.VarChar)).Value =
Trim(sqlReader.GetValue(0))
        cmdActividades.Parameters.Add(New
SqlParameter("@sessionID",

```

```

SqlDbType.VarChar)).Value =
Trim(sqlReader.GetValue(1))
        cmdActividades.Parameters.Add(New
SqlParameter("@origen",
SqlDbType.VarChar)).Value =
Trim(esNulo(sqlReader.GetValue(2)))
        cmdActividades.Parameters.Add(New
SqlParameter("@destino",
SqlDbType.VarChar)).Value =
Trim(esNulo(sqlReader.GetValue(3)))
        cmdActividades.Parameters.Add(New
SqlParameter("@tarifa", SqlDbType.Money)).Value =
Trim(sqlReader.GetValue(4))

            cmdActividades.ExecuteNonQuery()
Loop
connStrActividades.Close()
connStrUsuarios.Close()

Catch ex As Exception
    Throw New Exception("Error al Insertar
actividades: " & ex.ToString)
End Try
End Sub

Private Sub cobrarSaldo(ByVal control As
System.Windows.Forms.TextBox)
'Recorre la lista de usuarios, descontando el
saldo por minuto de cada tarjeta

Dim connStrUsuarios, connStrCobro As
SqlConnection
Dim sqlReader As SqlDataReader
Dim horaTarifa As String

Try
    connStrUsuarios = conectarBD()
    connStrUsuarios.Open()

'Obtiene los registros de los usuarios
conectados a ser tarificados
Dim cmdUsuarios As SqlCommand = New
SqlCommand("Sp_SelUsuariosConectados",
connStrUsuarios)
    cmdUsuarios.CommandType =
CommandType.StoredProcedure
    sqlReader = cmdUsuarios.ExecuteReader
    connStrCobro = conectarBD()
    connStrCobro.Open()

'Store procedure que cobra el minuto a cada
usuario
Dim cmdCobro As SqlCommand = New
SqlCommand("Sp_UpdCobroTarifa", connStrCobro)
Do While sqlReader.Read()
    cmdCobro.Parameters.Clear()
    cmdCobro.CommandType =
CommandType.StoredProcedure
    cmdCobro.Parameters.Add(New
SqlParameter("@login", SqlDbType.VarChar)).Value
= Trim(sqlReader.GetValue(0))
    cmdCobro.Parameters.Add(New
SqlParameter("@sessionID",
SqlDbType.VarChar)).Value =
Trim(sqlReader.GetValue(1))
    cmdCobro.ExecuteNonQuery()
    control.AppendText(CStr(numActividad) &
" - " & "Cobro : a la cuenta " &
sqlReader.GetValue(1) & " por concepto de 1 minuto

```

```

de Internet Prepagado" &
Microsoft.VisualBasic.ControlChars.CrLf)
    numActividad = numActividad + 1
    Loop
    connStrCobro.Close()
    connStrUsuarios.Close()

    Catch ex As Exception
        Throw New Exception("Error al cobrar
Saldos: " & ex.ToString)
    End Try
End Sub

Private Sub
removerUsuariosdesconectados(ByVal control As
System.Windows.Forms.TextBox)
    Dim connStrRadius, connStr As SqlConnection
    Dim sqlreader As SqlDataReader

    Try
        connStrRadius = conectarBDRadius()
        connStr = conectarBD()
        connStrRadius.Open()
        connStr.Open()

        Dim cmdRadius As SqlCommand = New
SqlCommand("Sp_SelUsuariosDesconectados",
connStrRadius)
        cmdRadius.CommandType =
CommandType.StoredProcedure
        cmdRadius.Parameters.Add(New
SqlParameter("@secsProceso",
SqlDbType.BigInt)).Value = secsProceso
        Dim cmdPrepago As SqlCommand = New
SqlCommand("Sp_DelUsuariosDesconectadosRadiu
s", connStr)
        cmdPrepago.CommandType =
CommandType.StoredProcedure

        sqlreader = cmdRadius.ExecuteReader

        'Remueve Usuarios Desconectados de la
lista de usuarios a tarificar( Acct_Status_Id=2 )
        While sqlreader.Read
            cmdPrepago.Parameters.Add(New
SqlParameter("@UserName",
SqlDbType.VarChar)).Value = sqlreader.GetValue(1)
            cmdPrepago.Parameters.Add(New
SqlParameter("@SessionId",
SqlDbType.VarChar)).Value =
Trim(sqlreader.GetValue(1))
            cmdPrepago.ExecuteNonQuery()
            cmdPrepago.Parameters.Clear()
            control.AppendText(CStr(numActividad) &
"- " & "Desconexion : El usuario " &
Trim(sqlreader.GetString(0)) & " se desconecto." &
Microsoft.VisualBasic.ControlChars.CrLf)
            numActividad = numActividad + 1
        End While
        connStrRadius.Close()
        connStr.Close()
        Catch ex As Exception
            Throw New Exception("Error al remover
usuarios (desconexiones de Radius): " &
ex.ToString)
        End Try
    End Sub

    Private Sub validaSaldos(ByVal control As
System.Windows.Forms.TextBox)

```

```

        'Valida cada usuario de la lista, desconectando
los usuarios cuyo saldo no sea suficiente para el
cobro de minuto

        Dim connStrUsuarios, connStrVerif As
SqlConnection
        Dim sqlReader As SqlDataReader
        Dim saldoValido As Boolean
        Dim obj_SNMP As New
Plataforma_SNMPTNT_class()
        Try
            connStrUsuarios = conectarBD()
            connStrUsuarios.Open()
            Dim cmdUsuarios As SqlCommand = New
SqlCommand("Sp_SelUsuariosConectados",
connStrUsuarios)
            cmdUsuarios.CommandType =
CommandType.StoredProcedure
            sqlReader = cmdUsuarios.ExecuteReader

            connStrVerif = conectarBD()
            connStrVerif.Open()
            Dim cmdVerif As SqlCommand = New
SqlCommand("Sp_verificaSaldos", connStrVerif)
            Do While sqlReader.Read()
                cmdVerif.Parameters.Clear()
                cmdVerif.CommandType =
CommandType.StoredProcedure
                cmdVerif.CommandText =
"Sp_verificaSaldos"
                cmdVerif.Parameters.Add(New
SqlParameter("@login", SqlDbType.VarChar)).Value
= Trim(sqlReader.GetString(0))
                saldoValido = cmdVerif.ExecuteScalar
                If Not saldoValido Then
                    'Desconexión Telnet del Usuario -----
                    aqui -----
                    ""obj_SNMP.desconectar_Usuario("",
Trim(sqlReader.GetString(1)))
                    'Desconexión Telnet del Usuario -----
                    aqui -----
                    cmdVerif.CommandText =
"Sp_DelUsuarioSaldo"
                    cmdVerif.Parameters.Add(New
SqlParameter("@SessionId",
SqlDbType.VarChar)).Value =
Trim(sqlReader.GetValue(1))
                    cmdVerif.ExecuteNonQuery()
                    control.AppendText(CStr(numActividad)
& "- " & "Desconexion : La cuenta " &
Trim(sqlReader.GetString(0)) & " ha sido
desconectada debido a falta de saldo." &
Microsoft.VisualBasic.ControlChars.CrLf)
                    numActividad = numActividad + 1
                End If
            Loop
            connStrUsuarios.Close()
            connStrVerif.Close()
            Catch ex As Exception
                Throw New Exception("Error al validar
Saldos:" & ex.ToString)
            End Try
        End Sub

#End Region

#Region "Funciones/Metodos Utilizadas por los
pasos de Tarificacion "
Private Function conectarBD() As SqlConnection

```

```

        'Conexión a la Base de Datos
        Return New SqlConnection("user
id=prepago;password=xdcc;initial
catalog=Prepago;data source=localhost;")
    End Function

    Private Function conectarBDRadius() As
    SqlConnection
        'Conexión a la Base de Datos de
    RadiusAccounting
        Return New SqlConnection("user
id=prepago;password=xdcc;initial
catalog=Temporal;data source=localhost;")
    End Function

    Private Function esNulo(ByRef valor As Object)
        If IsDBNull(valor) Then
            valor = ""
        End If
    End Function

    Private Sub depuraListaSesiones(ByRef
sesiones(,) As String)
        Dim tamArreglo As Int16 = 0
        Dim connStrRadius, connStr As SqlConnection

        Try
            connStrRadius = conectarBD()
            connStrRadius.Open()
            Dim cmdRadius As SqlCommand = New
    SqlCommand("Sp_SelSesionesValidas",
connStrRadius)
            cmdRadius.CommandType =
    CommandType.StoredProcedure

            For tamArreglo = 0 To sesiones.GetLength(1)
- 1
                cmdRadius.Parameters.Clear()
                cmdRadius.Parameters.Add(New
    SqlParameter("@UserName",
    SqlDbType.VarChar)).Value = sesiones(1,
tamArreglo)
                If cmdRadius.ExecuteScalar <= 0 Then
                    sesiones(1, tamArreglo) = ""
                End If
            Next
            connStrRadius.Close()
            Catch ex As Exception
                Throw New Exception("Error al Depurar
usuarios Iniciales: " & ex.ToString)
            End Try

        End Sub

#End Region

End Class

```

Plataforma_SNMPTNT.vb

```

Imports System
Imports System.Net.Sockets
Imports System.Net
Imports System.Threading

Interface Plataforma_SNMPTNT

    Sub obtener_Usuarios(ByVal IP As String)
    Sub desconectar_Usuario(ByVal IP As String,
ByVal SessionID As String)

End Interface

Public Class Plataforma_SNMPTNT_class
    Implements Plataforma_SNMPTNT

    Private puertoTelnet As Int16 = 23
    Private ntsTelnet As NetworkStream
    Private tcpCliente As ClienteTelnet
    Public listadoSesiones(,) As String

    *****
    *****
    "obtener_Usuarios:
    '-Se conecta al servidor TNT via SNMP y obtiene
    a los usuarios conectados en ese momento

    "desconectar_usuario:
    '-Recive el No. de sesion y se conecta al servidor
    TNT via Telnet y desconecta al usuario
    '-con ese No. de session

    *****
    *****

    Public Sub obtener_Usuarios(ByVal IP As String)
    Implements
    Plataforma_SNMPTNT.obtener_Usuarios
        Dim obj_SNMP As New SNMP_NET.SNMP()
        Dim key As String

        Try
            Dim tablahs As Hashtable =
obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.12.3.1.4", True)
            Dim tablahsOrigen As Hashtable =
obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.11.16.1.17", True)
            Dim tablahsDestino As Hashtable =
obj_SNMP.Walk(IP, "init$MrC+",
".1.3.6.1.4.1.529.11.16.1.16", True)

```



```

        Next
        Loop
        Do While enumDestino.MoveNext()
            For numRegistros = 0 To
                sesionesLocales.GetLength(1) - 1
                sesionTemporal =
Split(enumDestino.Key, ".")
                If
                    sesionTemporal.GetValue(sesionTemporal.Length -
1) = sesionesLocales(0, numRegistros) Then
                        sesionesLocales(3, numRegistros) =
enumDestino.Value
                            Exit For
                                End If
                                    Next
                                        Loop
                                            listadoSesiones = sesionesLocales
                                                Catch ex As Exception
                                                    Throw New Exception("Error al obtener
sesiones: " & ex.ToString)
                                                        End Try
                                                            End Sub

#End Region

End Class

Public Class ClienteTelnet
    Inherits TcpClient
    'Clase heredada por Plataforma SNMPTNT_class
    para manejar el cliente Telnet

    Public Sub New(ByVal host As String, ByVal port
As Integer)
        Try
            Me.Connect(host, port)
        Catch ex As Exception
            Dim errorMsg As New ErrorCs()
            errorMsg.SendError("Error al tratar de
conectarse via TELNET a la IP-" & host & " : " &
ex.ToString)
        End Try
    End Sub

    Public Function IsConnected() As Boolean
        Try
            Return Me.Client.Connected
        Catch ex As Exception
            Dim errorMsg As New ErrorCs()
            errorMsg.SendError("Error al tratar de
obtener estatus via TELNET : " & ex.ToString)
        End Try
    End Function
End Class

```

ErrorCs.vb

```

Imports System.Web.Mail

Public Class ErrorCs

    Public Sub SendError(ByVal mensaje As String)
        Dim Mail As New MailMessage()

        SmtplibMail.SmtplibServer = "MTELMY16"
        Mail.Body = "Error en el Servicio de tarificacion
de Internet prepagado"
        Mail.From = "Administrador Tarificacion"

```