



46
REJ
**UNIVERSIDAD NACIONAL
AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA DE FACTURACION CORPORATIVO
(SIFACOR) PARA TELCEL EN INFORMIX**

T E S I S

Que para obtener el Título de
INGENIERO EN COMPUTACION

p r e s e n t a n

Claudia Rodríguez Espino
Eva María Eugenia Sibaja Pastrana
Everardo Ochoa Suárez
Juan Manuel Corona Sánchez
Raúl González López



ASESOR: M. I. LAURO SANTIAGO CRUZ

México, D. F.

Octubre 1995

FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

***Queremos expresar nuestro especial agradecimiento a
aquellos sin cuya valiosa ayuda hubiera sido
imposible materializar este trabajo:***

***A nuestro director de Seminario
Por su apoyo y gran paciencia
M.I. Lauro Santiago Cruz***

A la Dirección de Sistemas TELCEL



**A DIOS POR DARLE A MIS PADRES LA
OPORTUNIDAD DE OTORGARME LOS
RECURSOS Y APOYO NECESARIOS
PARA DESARROLLARME
PROFESIONALMENTE.**

**A MIS PADRES QUIENES ME BRINDARON
SU APOYO Y DEDICACION, LOS CUALES
ME MOSTRARON EL CAMINO QUE SIGO
Y SEGUIRE EN LAS DIFERENTES ETAPAS
DE MI VIDA.**

**A MIS HERMANOS PATY, MAYI Y COKIS POR SOPORTARME
Y COMPRENDERME EN LOS MOMENTOS MAS DIFICILES
DE MI TRAYECTORIA ACADEMICA.**

**A LA UNIVERSIDAD, FACULTAD Y
PROFESORES POR COMPARTIR
CONMIGO LOS CONOCIMIENTOS
E INSTRUMENTOS NECESARIOS
PARA EL DESARROLLO DE
MI CARRERA.**

**A MIKE POR SU APOYO Y COMPRENSION
QUE ME AYUDARON AL TERMINO DE
MI CARRERA.**

CLAUDIA RDZ. E.

La vida me ha enseñado que debemos terminar lo que comenzamos.

Este trabajo es la realización de una de mis ilusiones, el premio a un esfuerzo que comenzó hace mucho tiempo y que no había podido consumir.

Pero todo lo invertido en este logro no significa nada en comparación con el cariño, apoyo, comprensión y amor que he recibido siempre de todos mis seres queridos.

Por eso, dedico mi trabajo a:

Mis padres Helia y Arsenio.

Mis hermanos Alejandro, Enrique y Rocío.

Eva



Octubre 1995

**A LA MEMORIA DE MI PADRE,
DE QUIEN SIEMPRE TUVE APOYO
TOTAL E INCONDICIONAL**

**A MI MADRE Y A MI HERMANA,
POR SUPUESTO**

**A MIS TÍOS GUADALUPE Y
RAFAEL SUÁREZ, POR SU
AYUDA Y MOTIVACIÓN**

**AL M.I. LAURO SANTIAGO CRUZ
POR SU DIRECCIÓN,
ASÍ COMO A MIS COMPAÑEROS Y
A MIS AMIGOS
A QUIENES DIRECTA E
INDIRECTAMENTE DEBO EN PARTE
LA REALIZACIÓN DE ÉSTE TRABAJO**

GRACIAS SEÑOR.

EVERARDO OCHOA SUÁREZ

Quiero expresar mi gratitud:

A Dios por permitirme haber concluido mis estudios satisfactoriamente.

A mis padres por enseñarme el sentido de la vida y guiarme exitosamente por el nada fácil camino de la preparación profesional.

A mis hermanos, cuñados y demás familiares que siempre me brindaron su apoyo y comprensión en todos los sentidos.

A los profesores que con sus conocimientos y experiencia me formaron profesionalmente.

A todos mis compañeros de estudio que me ayudaron para sobreponerme a obstáculos y dificultades para poder alcanzar mis metas de tipo educativo. En especial a mis amigos, más que compañeros: Raúl, Salvador, Fermin, César, Edgar, Daniel, Alfonso, Ernesto y Gabriel.

A mis compañeros de tesis, pues sin su apoyo este sueño no sería hoy una realidad.

Al personal del área de sistemas de Telcel por el apoyo y todas las facilidades prestadas en el desarrollo del presente trabajo.

Al M. I. Lauro Santiago Cruz por su ayuda, colaboración y por los muchos comentarios pertinentes en lo que se refiere a la forma de estructurar y elaborar este trabajo.

Juan Manuel

El espacio de esta página resulta realmente insuficiente para incluir a todas las personas a quienes quisiera agradecer, no así el espacio de la razón y el sentimiento en donde parientes y amigos están siempre presentes.

A TELCEL y su gente por el apoyo brindado a este proyecto de tesis.

A mis compañeros de seminario por su paciencia.

A mi madre.

Tal vez la única persona que nunca dudó de este logro y de quién siempre he recibido amor y apoyo a caudales.

A mi padre.

Por su apoyo, su ejemplo y aquella "Reserva del Patrón" que jamás olvide.

A mi hermano

Desde la meta se aprecian las cosas como nunca. las podrás ver desde el camino.

A Lourdes López mi tía consentida.

¡Mira lo hemos logrado! Sé que disfrutas tanto este logro como yo, gracias por tu cariño y apoyo.

A Roberto Valle Rizada mi tío abuelo.

Por los gratos momentos de niño y con admiración por el mayor don de gentes que he conocido.

A mis amigos y compañeros de carrera con quienes compartir aquél tiempo fué verdaderamente útil y divertido, en especial a: Salvador, César, Edgar, Daniel, Juan Manuel, Gabriel, Alfonso, Ernesto, Fermín, Rocío, Marta y Edith.

Y muy especialmente a Minerva.

Porque cualquier expresión hablada o escrita reflejarán pobremente la dimensión real del sentimiento, porque tienes mucho más que ver en este logro de lo que te imaginas, porque compartir contigo la vida es excitante y hermoso, por tu sonrisa, tu empuje y tu estrella, este logro lo dedico a ti.



Raúl González López.

Octubre 1995

ÍNDICE

SISTEMA DE FACTURACIÓN CORPORATIVO PARA TELCEL (SIFACOR)

INTRODUCCIÓN	1
CAPÍTULO UNO. CONCEPTOS BÁSICOS	6
1.1 GENERALIDADES DEL ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN	6
¿QUÉ ES EL ANÁLISIS DE SISTEMAS?	10
TIPOS DE ANÁLISIS DE INFORMACIÓN	12
HERRAMIENTAS PARA EL ANÁLISIS	14
DISEÑO DE LOS SISTEMAS DE INFORMACIÓN	15
PROCESO DE DISEÑO	16
EVALUACIÓN DE LA CALIDAD DE SOFTWARE	18
LA ETAPA DE DESARROLLO	18
LENGUAJES DE CUARTA GENERACIÓN	19
LENGUAJES NO ORIENTADOS HACIA LOS PROCEDIMIENTOS	20
LENGUAJES DE CONSULTA Y RECUPERACIÓN	20
GENERADORES DE REPORTES	20
GENERADORES DE APLICACIONES	20
GENERADORES DE PANTALLA	21
SISTEMA DE DICCIONARIO DE DATOS	21
BIBLIOTECAS DE CÓDIGO REUTILIZABLE	21
HERRAMIENTAS DE TIPO BACK-END	22
HERRAMIENTAS INTEGRALES	22
HERRAMIENTAS ASISTIDAS POR COMPUTADORA PARA LA INGENIERÍA DE SISTEMAS (CASE)	22
GENERADOR DE INTERFASES	22
GENERADORES DE CÓDIGO	22
PRUEBAS Y LIBERACIÓN	23
TÉCNICAS DE DISEÑO DE CASOS DE PRUEBA	24
PRUEBAS DE CAJA BLANCA	24
PRUEBAS DE CAJA NEGRA	24
ESTRATEGIAS DE PRUEBA	25
DEPURACIÓN	26
MANTENIMIENTO	27
MANTENIMIENTO CORRECTIVO	28
MANTENIMIENTO ADAPTATIVO	28

MANTENIMIENTO PERFECTIVO	29
MANTENIMIENTO PREVENTIVO	29
COSTO DEL MANTENIMIENTO	29
CONCLUSIONES	31
CAPÍTULO DOS. PLANEACIÓN	32
2.1. DEFINICIÓN DE PLANEACIÓN	33
2.2. MÉTODOS DE PLANEACIÓN	33
OPCIONES PARA LA REALIZACIÓN DE ESTIMACIONES	34
FACTORES QUE AFECTAN LA PRECISIÓN DE LAS ESTIMACIONES	35
QUÉ ES LO QUE SE ESTIMA	36
ANÁLISIS DE RIESGOS	36
IDENTIFICACIÓN DEL RIESGO	36
PROYECCIÓN DEL RIESGO	37
EVALUACIÓN DEL RIESGO	37
GESTIÓN Y SUPERVISIÓN DEL RIESGO	38
SEGUIMIENTO Y CONTROL	38
2.3. ALCANCES	38
2.4. PLANEACIÓN DEL SISTEMA DE FACTURACIÓN	39
CONCLUSIONES	44
CAPÍTULO TRES. ANÁLISIS	46
3.1. RECOPIACIÓN DE INFORMACIÓN	47
LIMITANTES DEL SISTEMA ACTUAL	50
RECOPIACIÓN DE INFORMACIÓN	51
3.2. NECESIDADES DEL SISTEMA	54
3.3. ALTERNATIVAS DE SOLUCIÓN	55
3.4. ANÁLISIS ESTRUCTURADO	55
3.5. FACTIBILIDAD DEL SISTEMA	70
FACTIBILIDAD TÉCNICA	70
FACTIBILIDAD ECONÓMICA	71
FACTIBILIDAD OPERATIVA	72
CONCLUSIONES	72
CAPÍTULO CUATRO. DISEÑO	73
4.1. ESPECIFICACIONES DEL SISTEMA PRINCIPAL	74
4.2. ESPECIFICACIONES DE LOS SUBSISTEMAS	74
MÓDULO DE CATÁLOGOS	75
DESCRIPCIÓN DE LOS CATÁLOGOS	75
REGIONES	75
TIENDAS	75
PROVEEDORES	75

CLIENTES	75
CUENTAS	75
PRODUCTOS Y SERVICIOS	75
PLANES TARIFARIOS	76
MÓDULO DE FACTURACIÓN	76
DESCRIPCIÓN DE LAS OPCIONES DE FACTURACIÓN	76
FACTURACIÓN	76
NOTAS DE CRÉDITO	76
NOTAS DE CARGO	76
FACTURACIÓN ROAMING	77
REIMPRESIÓN	77
REIMPRESIÓN MÚLTIPLE	77
CONTRATOS CELULARES	77
MÓDULO DE CONSULTAS Y REPORTES	77
CONSULTA DE FACTURACIÓN DIARIA	77
REPORTE DE INVENTARIOS	78
CONSULTA DE PRODUCTOS	78
REPORTE ACUMULADO DE VENTAS POR ASESOR	78
MÓDULO DE ADMINISTRACIÓN DEL SISTEMA	78
CATÁLOGO DE USUARIOS DEL SISTEMA	79
CATÁLOGO DE PROGRAMAS	79
PERFILES DE USUARIO	79
CONTROLADOR DE FOLIOS	79
4.3. DISEÑO DE LA BASE DE DATOS	79
BASE DE DATOS RELACIONAL	80
LLAVES PRIMARIAS Y FORÁNEAS	80
DIAGRAMAS DE ENTIDAD-RELACIÓN	82
DIAGRAMA DE RELACIÓN DE TABLAS	92
DICCIONARIO DE DATOS	94
DICCIONARIO DE ALMACENES DE DATOS	96
DEFINICIÓN DE TABLAS	98
4.4. SEGURIDAD DEL SISTEMA	103
4.5. DISEÑO DE PANTALLAS, CONSULTAS Y REPORTES	103
4.6. PSEUDOCÓDIGO	108
PROGRAMA: CATÁLOGO DE PERFILES DE USUARIO	108
PROGRAMA: CATÁLOGO DE CLIENTES	113
PROGRAMA: INICIALIZADOR DE FOLIOS	117
PROGRAMA: ELABORACIÓN DE DOCUMENTOS	119
PROGRAMA: REPORTE DE VENTAS POR ASESOR	121
CONCLUSIONES	123

CAPÍTULO CINCO. DESARROLLO **124**

5.1. LA HERRAMIENTA DE PROGRAMACIÓN INFORMIX	125
GENERALIDADES	125
ARQUITECTURA DE INFORMIX	125
INFORMIX-4GL Y LA ARQUITECTURA	

CLIENTE-SERVIDOR	127
HERRAMIENTAS	128
HERRAMIENTAS DE APLICACIÓN	128
PRODUCTOS OPEN CASE	129
SERVIDORES DE BASES DE DATOS	129
INSTRUCCIONES DE MANIPULACIÓN DE BASE DE DATOS	130
CONSULTAR REGISTROS	130
OPERADORES CONDICIONALES DE INFORMIX	131
MODIFICAR REGISTROS	132
INSERTAR REGISTROS	132
BORRAR REGISTROS	132
INTEGRIDAD	132
MÉTODO 1. USO DE LA UTILERÍA ARCHIVE	132
MÉTODO 2. EL USO DE ESPEJO	133
MÉTODO 3. LA MIGRACIÓN	133
5.2. PROGRAMACIÓN	134
MÓDULO DE CATÁLOGOS. PROGRAMA FTM75001.4GL.	
ALTA EN CATÁLOGO DE CLIENTES	135
MÓDULO DE FACTURACIÓN. PROGRAMA FTP10000.4GL.	
PEDIDOS	136
MÓDULO DE CONSULTAS Y REPORTES. PROGRAMA	
FTM75000.4GL. ALTAS, CONSULTAS Y REPORTES DE	
CLIENTES	141
MÓDULO DE ADMINISTRACIÓN DEL SISTEMA. PROGRAMA	
SPMA5000.4GL. MANTENIMIENTO DE VISTAS. PROGRAMA	144
SPMA6000.4GL. MANTENIMIENTO AL CATÁLOGO DE	
FOLIOS	150
CONCLUSIONES	152
CAPÍTULO SEIS. IMPLANTACIÓN, LIBERACIÓN Y MANTENIMIENTO	153
6.1. IMPLANTACIÓN	154
INSTALACIÓN DEL SISTEMA	155
AMBIENTES DE DESARROLLO Y PRODUCCION	155
COPIADO DE LA BASE DE DATOS AL AMBIENTE DE	
PRODUCCIÓN	155
TRANSFERENCIA DE ARCHIVOS EJECUTABLES AL	
AMBIENTE DE PRODUCCIÓN	156
FINALIZACIÓN DE LA DOCUMENTACIÓN DEL	
DESARROLLO	156
PRUEBAS Y EVALUACIÓN	157
PRUEBAS DE CONTROL Y UNIDAD	157
PRUEBA DE INTEGRACIÓN	157
PRUEBA DE RESISTENCIA	157
PRUEBAS DE VALIDACIÓN	158
DEPURACIÓN Y AJUSTES	159
PRUEBA DE ACEPTACIÓN	159
CAPACITACIÓN	159
6.2. LIBERACIÓN	161

PUESTA EN MARCHA	161
6.3. MANTENIMIENTO	162
MANTENIMIENTO CORRECTIVO	162
MANTENIMIENTO ADAPTATIVO	162
MANTENIMIENTO PERFECTIVO	162
MANTENIMIENTO PREVENTIVO	163
ACTUALIZACIONES	163
SEGURIDAD DE LA INFORMACIÓN	163
CONCLUSIONES	164
CONCLUSIONES GENERALES	165
APÉNDICES	169
APÉNDICE 1. LA NECESIDAD DE FACTURAR	170
APÉNDICE 2. REQUISITOS FISCALES PARA LA AUTOIMPRESIÓN DE COMPROBANTES FISCALES	173
APÉNDICE 3. FICHAS TÉCNICAS	174
BIBLIOGRAFÍA	177

INTRODUCCIÓN

TELCEL es una empresa dedicada a la telefonía celular. Entre sus actividades se cuenta la venta de aparatos telefónicos celulares, sus accesorios, así como la prestación de servicios inherentes a la comunicación inalámbrica.

La razón social de TELCEL es Radio Móvil Dipsa (Directorios Profesionales, S.A.), que es una compañía administrada por Telmex, una de las empresas que conforman el grupo Carso que cuenta entre otras filiales con Euzkadi, Hoteles Calinda y Cigatam.

A pesar de ser una empresa de reciente creación, pues se fundó hace seis años aproximadamente, el auge que han tenido los productos y servicios que ofrece se ha incrementado enormemente. Para dar idea de este aumento, el número de clientes o abonados de esta compañía pasó de 20 000 a 200 000 personas en los últimos tres años.

Es de mencionar que TELCEL es tan sólo una de las muchas empresas que se dedican en México a lo relacionado con servicios telefónicos celulares. Aunque se ha diversificado la clase de servicios que proporcionan, básicamente son los mismos en unas y otras. Por lo tanto, la competencia por captar el mercado y mantenerse vigentes es intensa y se lleva a cabo a través de extensas campañas publicitarias y agresivas estrategias de venta. Pero lo que hace la diferencia entre una compañía y otra es, desde el punto de vista del cliente, la clase de atención que recibe en sus tratos con la empresa, por este motivo, una de las preocupaciones fundamentales de TELCEL es la de proporcionar al cliente servicios de calidad y esmerada atención en sus oficinas.

Como es sabido, en los tratos comerciales que se efectúan en nuestros días, las facturas que expide la compañía prestadora de servicios son el único documento que relaciona a la empresa con su cliente y que avala la legitimidad de un bien o servicio.

Muchas compañías conscientes de esta situación buscan que estos documentos sean claros y atractivos, ya que reflejan la imagen de la misma ante sus clientes. Sin embargo, más importantes que el diseño o la elegancia de una factura, son la integridad de la información ahí contenida y el proporcionar un trámite expedito al cliente.

La integridad de la información de la facturación es fundamental para la empresa prestadora de servicios, dado que la información frágil o inconsistente ocasiona pérdidas parciales de datos, lo que puede generar severas sanciones fiscales. Por esa razón se justifica el cuidadoso trato que se le da a este tipo de información.

En sus inicios, TELCEL contaba con un solo Centro de Atención al Cliente, ubicado en la colonia Roma, en la Cd. De México. No hacían falta más oficinas de ese tipo ya que el número de clientes era sumamente reducido y la mayoría de ellos se concentraba en esa zona.

A fin de dar un servicio adecuado a los abonados de la empresa, se implantó un sistema computarizado para manejar un control de las ventas y otros datos de interés para TELCEL.

Pero en recientes sondeos se estableció que la cantidad de abonados celulares (clientes) de esta empresa, crecía a un ritmo de 12 % mensual. Este crecimiento real superó en mucho el que se tenía proyectado, por lo que las consecuencias en el sistema de facturación anterior no se hicieron esperar.

Este sistema comenzó a mostrar graves fallas de integridad en los datos y rendimiento al manejar volúmenes de datos cada vez mayores. Por ejemplo, la elaboración de una sola nota de venta típica - que despacha un aparato telefónico, algún accesorio y un servicio -, tarda alrededor de 22 minutos. Para calcular la ineficiencia del sistema, baste multiplicar este tiempo por la cantidad de clientes que acudían al Centro de Atención en ese mismo día. No es necesario mencionar que los clientes evitaban en lo posible asistir a este lugar.

En cuanto a la confiabilidad de la información, es de mencionar que el sistema de facturación actual permite la recaptura del número consecutivo de folio sin validar la existencia previa del documento. Por lo que es frecuente generar facturas con números de folio duplicados.

Un problema adicional al anterior, es que no existe liga alguna entre la facturación y la generación de notas de crédito o cargo. Por lo que hay que hacer un rastreo prácticamente manual de estas notas.

Además, es frecuente que los operarios inexpertos cometan errores sin que el sistema esté listo para detectarlos, lo que provoca graves fallas generales. Estas fallas ocasionan pérdidas económicas importantes y crean una mala imagen al público, por la falta de oportunidad y confiabilidad de la información.

Aunado a lo anteriormente descrito, cabe mencionar que se planea aumentar el número de Centros de Atención al Cliente de uno a nueve, que se distribuirán en toda la República, todos los cuales proporcionarían servicios similares a los que se prestan en el Centro de Atención del D.F.

Estos centros regionales estarán ubicados en ciudades clave y todos ellos se comunicarán a fin de mantener homogeneidad y modernidad en el servicio. Las ciudades que se tienen proyectadas son: Tijuana, Hermosillo, Chihuahua, Monterrey, Guadalajara, Querétaro, Puebla y Mérida.

Se cree que como promedio cada uno de estos Centros de Atención Regionales deberá generar de 150 a 200 facturas diariamente.

Ahora bien, en las oficinas de Atención a Clientes de TELCEL se elaboran notas de venta de los distintos productos y servicios que esta empresa ofrece tales como:

1. Aparatos telefónicos de distintas marcas.
2. Accesorios: pilas, cargadores, fundas, eliminadores de baterías, pilas de alta duración, pilas de baja duración, pilas de cargado rápido, antenas.

3. Servicios de: Activación, Conferencia tripartita, Buzones: Universal y Personalizado, Cambio de Número Telefónico y Tasación. Este último es el costo del tiempo aire de las llamadas del usuario, que se procesa en equipos Vax bajo otra aplicación.¹

Recientemente, la Secretaría de Hacienda y Crédito Público dio a conocer en el Diario Oficial una serie de condiciones con las que tienen que cumplir este tipo de aplicaciones comerciales, a fin de poder dar un cabal seguimiento a los procesos hacendarios en las declaraciones de impuestos de las empresas. Esta institución también determinó que la aplicación de apoyo a la facturación que corría en TELCEL, no cumplía con estos requisitos, por lo que era obligatorio modificarla.

Por todo lo anterior, se hizo necesaria, si no es que urgente, la sustitución del sistema de facturación actual por uno se acoplara a las necesidades de TELCEL en el corto y mediano plazo.

El presente trabajo tiene como objetivo la creación y establecimiento de un sistema que permita satisfacer cabalmente las necesidades del proceso de facturación que se lleva a cabo en las oficinas de TELCEL.

Uno de los principales objetivos de la nueva aplicación, a la cual le hemos nombrado SIFACOR por Sistema de Facturación Corporativa, es reducir los costos de operación y mejorar la presentación de la documentación que se entrega al cliente, cumpliendo además con los requisitos señalados por la S.H.C.P.

El SIFACOR deberá generar de forma rápida, confiable y veraz, la documentación que más comúnmente se hace llegar a las manos del cliente al momento de la contratación de una línea celular, o al adquirir algún otro bien o servicio, produciendo un documento con excelente presentación.

La operación del SIFACOR debe ser lo más sencilla posible y la seguridad de la información deberá cubrir los siguientes puntos fundamentales:

- 1. Eliminar la duplicidad de folios, trabajando bajo un ambiente multiusuario.**
- 2. Mantener fuertemente la integridad de los datos aun cuando el sistema sea operado por usuarios inexpertos.**

El sistema deberá estructurarse de tal forma que los usuarios cuenten con ayudas y accesos a catálogos en línea, tales como de Proveedores, Clientes, Productos, Servicios y Planes Tarifarios.

Deberá ser posible, además, la elaboración de reportes operativos y para la toma de decisiones como los siguientes: Reporte de Ventas por Asesor, Reporte de Ventas por Centro de Atención y Reporte de las Marcas de Productos más vendidos.

Se contempla que el SIFACOR crecerá en el futuro, es posible que se le añadan módulos específicos de orientación contable y financiera.

¹ Este tipo de servicio no será objeto de atención del presente trabajo.

Este proyecto se elaborará en base a la infraestructura de comunicaciones, el hardware y el software con que cuenta la empresa, que se especifica a continuación:

1. Equipo multiusuario Bull DPX2 , sistema operativo UNIX V. 4 de AT&T. Se proyecta adquirir otros ocho equipos similares para ubicarlos en cada uno de los centros regionales. La comunicación entre estos mainframes se establecerá vía satélite, microondas e incluso cable telefónico. Cabe mencionar que este equipo proporciona salida TCP/IP (Transmission Control Protocol/Internal Protocol) y ofrece determinadas facilidades en telecomunicación. Las terminales localizadas en la misma ciudad estarán conectadas a través de redes locales.
2. Recientemente TELCEL adquirió el manejador de bases de datos relacionales INFORMIX V. 5, que provee una serie de herramientas que se pretende utilizar en el desarrollo del sistema propuesto tales como: debugger , generador de reportes, generador de aplicaciones, interfases gráficas y principios de programación orientada a objetos. Acepta además SQL.
3. También, se utilizará un nuevo equipo de impresión para la emisión de facturas, se trata del equipo XEROX modelo 4213, que permite la carga de formatos predefinidos y que sólo requiere alimentación de papel membretado. La información cuadra automáticamente al imprimirse la factura y la calidad de impresión es excelente.

Cabe mencionar que TELCEL fue autorizado para emitir sus propios comprobantes fiscales.

En resumen, la estructura del presente trabajo consiste de seis capítulos y cuatro apéndices.

En el capítulo uno veremos las generalidades de análisis y diseño de sistemas de información, empezando por la definición de qué es el análisis, qué tipos existen, sus herramientas, para luego seguir con el proceso de diseño de sistemas. Pasaremos luego a la etapa de desarrollo o codificación. Posteriormente se dirigirá la atención a las pruebas del sistema, determinadas principalmente por las denominadas de caja blanca y caja negra, y como último punto, se considerará el mantenimiento como la fase final del proceso de ingeniería de software.

En el capítulo dos se considerarán los elementos que conforman la planeación de un proyecto de software así como sus alcances. En ese capítulo se hará una definición de lo que es la planeación y se describirán brevemente los principales métodos de planeación de sistemas de información. Asimismo, se explicarán los alcances y limitaciones de la planeación. Por último, se presentará la planeación del proyecto SIFACOR.

El capítulo tres se abocará al análisis de información. Se establecerán las necesidades y operaciones que la empresa quiere atender a través de un nuevo sistema de información. Se analizará el sistema de facturación actual REAL WORLD, mediante el establecimiento de un diagrama de contexto que mostrará la relación del sistema con su entorno, y permitirá evaluar la posibilidad de modificarlo para satisfacer los requerimientos de los usuarios, que se complementarán mediante los resultados de la recopilación de información apoyada en entrevistas principalmente. Basados en el análisis de la información obtenida, se propondrán alternativas de solución; una vez seleccionada una alternativa se estudiarán con detalle los procesos y flujos de información mediante el análisis estructurado, estableciendo así las funciones que son susceptibles de automatizarse y que se incluirán en el nuevo sistema.

Finalmente se hará una evaluación de la factibilidad del desarrollo de la aplicación en función de los recursos con que cuenta la organización.

El capítulo cuatro contendrá lo relativo al diseño de la base de datos de la aplicación, así como la estructura funcional de los módulos del sistema, tomando como base los requisitos detectados en el análisis.

El capítulo cinco versará sobre las características más importantes de las herramientas de programación de INFORMIX que se utilizarán para el desarrollo del SIFACOR.

El capítulo seis tratará acerca de las etapas que se llevarán a cabo en la última fase del ciclo de vida de desarrollo de sistemas; la primera de ellas corresponde a la implantación, proceso mediante el cual se hacen los preparativos para la primera operación "en vivo" del nuevo sistema, dicha etapa abarca la verificación e instalación del equipo en que funcionará la aplicación, el entrenamiento necesario del personal para que sea capaz de usar y operar el nuevo sistema, incluyendo las pruebas pertinentes para dejar al sistema libre de errores. En segundo término se continuará con la liberación, es decir, aquella actividad que ocurre cuando el sistema alcanza su mayor nivel de eficiencia en relación a los requerimientos que debe atender y, por lo tanto, es dejado en las manos de los usuarios finales para su operación cotidiana. El mantenimiento se centrará en los cambios asociados a la corrección de errores, las modificaciones debidas a la evolución del entorno y otros dirigidos a reforzar o ampliar al sistema.

El apéndice uno contiene una breve historia del proceso de facturación a través del tiempo.

En el apéndice dos se presentan los principales requisitos que emitió la Secretaría de Hacienda y Crédito Público para los autoemisores de comprobantes fiscales. Mismos que se tomaron en cuenta en el capítulo tres dentro del análisis de requisitos.

En el apéndice tres se presentan las esquemas o fichas técnicas del equipo BULL DPX2, que soportará a la aplicación; la impresora XEROX modelo 4213 y el lector óptico Symbol.

CAPÍTULO 1

CONCEPTOS BÁSICOS

En el presente capítulo veremos las generalidades del análisis y diseño de sistemas de información, empezando por la definición de qué es el análisis, que tipos existen, sus herramientas, para luego pasar a la etapa de diseño de los sistemas y contemplar algunos tipos de diseño. Hablaremos luego de la etapa de desarrollo o codificación, de las pruebas del sistema, determinadas principalmente por las denominadas de caja blanca y caja negra, y como último punto del mantenimiento como última fase del proceso de ingeniería de software.

1.1. GENERALIDADES DEL ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN

La información. Este elemento que hace algunas décadas no era considerado como un recurso básico de las organizaciones -como lo podrían ser los recursos humanos ó la materia prima-, comienza a cobrar su real dimensión a partir de que los responsables de la toma de decisiones perciben que ya no es un producto exclusivamente colateral de la operación de la empresa, sino que es en sí, uno de los promotores de la misma.

Con el fin de lograr la máxima utilidad de la información, ésta debe administrarse de manera correcta, como ocurriría con cualquier otro de los recursos de la empresa. Los directivos deben entender que existen costos que se asocian con la producción, distribución, seguridad, almacenamiento y recuperación de la información. Aunque la información se encuentra aparentemente siempre a nuestro alcance, su uso estratégico como un apoyo de la competitividad de nuestro negocio no debe considerarse como un elemento gratuito.

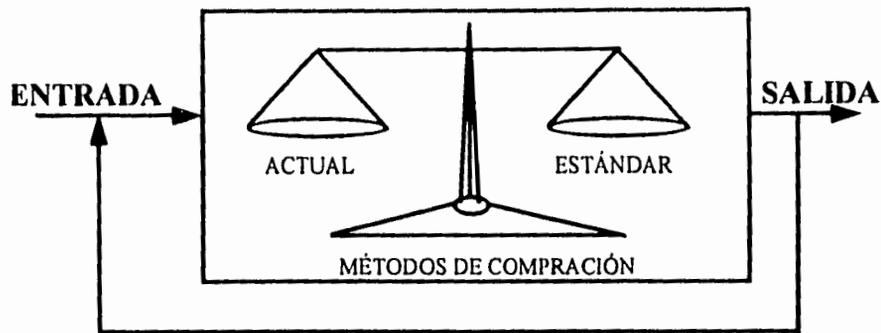
La disponibilidad actual de las computadoras ha generado todo un incremento y una diversificación de la información, tanto para la sociedad en general como para los negocios en particular. La administración de la información que se genera por computadora difiere en diversas formas de aquella que se obtiene manualmente. A menudo se tiene una mayor cantidad de información si ésta se genera utilizando sistemas computacionales; los costos para crear y mantener la información computarizada son aparentemente mayores; la información que genera la computadora puede llegar a multiplicarse a velocidades impresionantes. Con frecuencia la información así generada se trata con menos escepticismo que la obtenida por otros medios.

Sistema. Aunque esta palabra se aplica en infinidad de contextos, en informática se ha tornado como la base del desarrollo de aplicaciones. En el sentido más amplio, un sistema es un conjunto de componentes que se relacionan entre sí para lograr un objetivo común. Nuestra sociedad está rodeada de sistemas, por ejemplo, cualquier persona experimenta emociones físicas gracias a un complejo sistema nervioso formado por el cerebro, la médula espinal, los nervios y las células sensoriales especializadas que se encuentran debajo de la piel; estos elementos funcionan en conjunto para hacer que el sujeto experimente sensaciones de frío, calor, comezón, etc. Así mismo, las personas viven en un sistema económico en el que intercambian bienes y servicios por otros de valor comparable y en el que, al menos en teoría, los participantes obtienen un beneficio.

La *finalidad de un sistema* es la razón de su existencia y para alcanzar sus objetivos, los sistemas interactúan con su medio ambiente, el cual está formado por todos los elementos que se encuentran fuera de las fronteras de los mismos. Los sistemas que interactúan con su medio ambiente (que reciben entradas y producen salidas) se denominan sistemas abiertos. En contraste, aquellos que no interactúan con su medio ambiente se conocen como sistemas cerrados. Todos los sistemas actuales son abiertos, es así como los sistemas cerrados existen sólo como un concepto, aunque muy importante, como se verá más adelante.

El elemento de *control* está relacionado con la naturaleza de los sistemas, sean cerrados o abiertos. Los sistemas trabajan mejor cuando operan dentro de niveles de desempeño tolerables denominados estándares y contra los que se comparan los niveles de desempeño actuales. La información proporcionada al comparar los resultados de los estándares junto con el proceso de reportar las diferencias a los elementos de control recibe el nombre de retroalimentación. Ver la figura 1.1.

Los sistemas emplean un modelo de control básico que se resume en los siguientes puntos:



RETROLIMENTACIÓN DE RESULTADOS

FIGURA 1.1.

1. Un estándar para lograr un desempeño aceptable.
2. Un método para medir el desempeño actual.
3. Un medio para comparar el desempeño actual contra el estándar.
4. Un método de retroalimentación.

Los sistemas que pueden ajustar sus actividades para mantener niveles aceptables continúan funcionando. Aquellos que no lo hacen tarde o temprano dejan de trabajar.

El concepto de *interacción* con el medio ambiente, que es lo que caracteriza a los sistemas abiertos, es esencial para el control. Recibir y evaluar la retroalimentación permite al sistema determinar que tan bien está operando.

En contraste, los sistemas cerrados sostienen su nivel de operación siempre y cuando posean información de control adecuada y no necesiten nada de su medio ambiente. Dado que esta condición no puede sostenerse por mucho tiempo, la realidad es que no existen sistemas cerrados. El concepto, sin embargo, es importante porque ilustra un objetivo del diseño de sistemas: construir sistemas que necesiten la menor intervención del medio externo para mantener un desempeño aceptable. Por consiguiente, la autorregulación y el propio ajuste son objetivos de diseño en todos los ambientes de sistemas.

Es importante destacar que los componentes que forman un sistema pueden ser a su vez sistemas más pequeños, es decir, los sistemas pueden estar formados por varios niveles de sistemas o subsistemas.

Hasta este momento se han descrito conceptos de sistemas en general, ya que como vemos, casi cualquier ámbito es susceptible de descomponerse en sistemas y subsistemas; la informática no es la excepción. Las finalidades de los sistemas de información son:

procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas.

Los sistemas de información están formados por subsistemas que incluyen hardware, software, medios de almacenamiento de datos para archivos y bases de datos. El conjunto particular de subsistemas utilizados -equipo específico, programas, archivos y procedimientos- es lo que se denomina una aplicación de sistemas de información.

Dado que los sistemas de información dan soporte a los demás sistemas de una organización, los analistas tienen primero que estudiar el sistema organizacional como un todo para entonces detallar sus sistemas de información. Los organigramas se emplean con frecuencia, para describir la forma como están relacionados los diferentes componentes de la organización, tales como divisiones, departamentos, oficinas y empleados. Aunque los organigramas indican con precisión las relaciones formales entre los diferentes componentes no dicen nada con respecto a la forma en que opera el sistema organizacional; ya que en este tipo de diagramas no es posible plasmar todos los detalles importantes. A continuación se dan varios ejemplos de detalles importantes para el análisis de sistemas:

- 1.- **Canales informales.** ¿Qué interacciones existen entre las personas y los departamentos que no aparecen en el organigrama o no están descritos en los procedimientos de operación?
- 2.- **Interdependencia.** ¿De qué otros departamentos y componentes de la organización depende un elemento en particular?
- 3.- **Personas y funciones clave.** ¿Cuáles son las personas y los elementos más importantes en el sistema para que éste tenga éxito?
- 4.- **Enlaces críticos de comunicación.** ¿Cómo es el flujo de información e instrucciones entre los distintos componentes de la organización? ¿Cómo se comunican entre sí?

La anterior no es una lista exhaustiva de preguntas pero recalca la responsabilidad de identificar las características importantes y necesarias que deben tener los nuevos sistemas.

Actividades de los analistas de sistemas. El analista especifica la forma en que va a operar el sistema y sus subsistemas, las entradas requeridas, las salidas que se deben producir y los trabajos que se efectuarán tanto por las computadoras como en forma manual. Por otro lado, también participan en el control de los sistemas de dos formas: la primera, cuando describen los elementos de control, tales como estándares y métodos para evaluar el desempeño en relación con los demás estándares para los sistemas de información que diseñan. Al mismo tiempo, los sistemas que son especificados, proporcionan información a los directivos y usuarios que permite a éstos determinar si los sistemas que administran operan correctamente. Además, un paso esencial en el diseño es incorporar mecanismos de retroalimentación ya que su inclusión permite sostener la operatividad del sistema.

¿QUÉ ES EL ANÁLISIS DE SISTEMAS?

El análisis y diseño de sistemas, tal como lo realizan los analistas de sistemas, pretende estudiar sistemáticamente la operación de ingreso de datos, el flujo de los mismos y la salida de la información; todo ello dentro del contexto de una empresa en particular. En suma, el análisis y el diseño de sistemas sirven para analizar, diseñar y fomentar mejoras en la operación de la empresa, lo cual puede realizarse mediante el uso de sistemas de información computarizados.

Si un sistema se instala sin una planeación adecuada, es muy probable que no cumpla con los objetivos para los que fue hecho, y después de que el usuario sólo satisfaga a medias sus necesidades y batalle con él por algún tiempo, poco a poco lo irá dejando en desuso, para quedar finalmente en el olvido. El análisis y el diseño de sistemas permiten estructurar el costoso esfuerzo de la implantación de los sistemas de información, que de otra manera ocurriría de forma azarosa. El diseño y el análisis de sistemas se conforman por una serie de procesos, que al ejecutarse sistemáticamente mejoran la operación de un negocio, mediante el uso de los sistemas de información computarizados. Una buena parte del análisis y del diseño de sistemas involucra el trabajo en colaboración con los usuarios, actuales o eventuales, de estos sistemas de información.

Dentro de las organizaciones, el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con los métodos y procedimientos más adecuados.

Para dar una definición de *análisis de sistemas* podríamos mencionar que es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

Por su parte, una definición de *diseño de sistemas* podría ser: el proceso de planificar, reemplazar o complementar un sistema organizacional existente por otro mejor, previamente dilucidado en un análisis.

Así pues, **el análisis especifica qué es lo que el sistema debe hacer mientras que el diseño establece cómo alcanzar los objetivos.**

Existe una metodología, mezcla de experiencia y técnica, que se denomina "**Ciclo de vida de los sistemas**" (SDLC: de sus siglas en inglés **System Development Life Cycle**). Existe cierto grado de desacuerdo en cuanto al número exacto de etapas que la conforman. En el presente trabajo sin embargo, presentaremos el SDLC que ha demostrado ser el que más se apega a un estándar.

El análisis de sistemas corresponde a las tres primeras etapas del SDLC, **la primer etapa** es donde el analista se involucra en la identificación de los problemas, de las oportunidades y de los objetivos, esta fase es crucial para el éxito del resto del proyecto, pues nadie está

dispuesto a desperdiciar su tiempo dedicándolo al problema equivocado. En esta etapa se requiere que el analista observe de forma objetiva lo que ocurre en la empresa. Luego, en conjunto con los otros miembros de la organización, hará notar los problemas. Muchas veces esta detección de necesidades ya fue realizada previamente y por eso se llega a la conclusión de invitar al analista para hacer una investigación más a fondo y determinar las oportunidades del sistema en el ámbito examinado.

Las oportunidades son aquellas situaciones que el analista considera que pueden perfeccionarse mediante el uso de los sistemas de información computarizados. Al aprovechar las oportunidades, la empresa puede lograr una ventaja competitiva o llegar a establecer un estándar industrial.

La identificación de objetivos también es un componente importante de la primera fase. En primera instancia, el analista deberá descubrir lo que la empresa intenta realizar. Y luego, estará en posibilidad de determinar si el uso de los sistemas de información apoyará a la empresa a alcanzar sus metas al encaminarla a oportunidades específicas.

En la segunda etapa del SDLC se busca la determinación de los requerimientos de información a partir de los usuarios particularmente involucrados. Para determinar los requerimientos de información dentro de la empresa pueden utilizarse diversos instrumentos, los cuales incluyen: el muestreo, el estudio de los datos y formas usadas por la organización, la entrevista, los cuestionarios, la observación de la conducta de quien toma las decisiones, así como de su ambiente y también el desarrollo de prototipos.

En esta etapa el analista hace todo lo posible por identificar qué información requiere el usuario para desempeñar sus tareas. Puede ver cómo varios de los métodos para establecer necesidades de información lo obligan a relacionarse directamente con los usuarios. Esta etapa sirve para elaborar la imagen que el analista tiene de la organización y de sus objetivos. En ocasiones se llegan a concluir sólo las dos primeras etapas del SDLC.

La tercera etapa que ejecuta el analista de sistemas consiste en determinar las necesidades propias del sistema. Una vez más existen herramientas y técnicas especiales que facilitan al analista la realización de las determinaciones requeridas. Estas incluyen el uso de diagramas de flujo de datos que cuentan con una técnica estructurada para representar en forma gráfica la entrada de datos de la empresa, los procesos y la salida de información. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos que contiene todos los elementos que utiliza el sistema, así como sus especificaciones y el espacio de impresión que requieren.

Durante esta fase el analista de sistemas también estudia las decisiones estructuradas por realizar, que son decisiones donde las condiciones, alternativas, acciones y reglas de acción podrían determinarse. Existen tres métodos para el análisis de decisiones estructuradas: el lenguaje estructurado (en nuestro caso el español), las tablas de decisiones y los árboles de decisiones.

No todas las decisiones en las empresas son estructuradas; no obstante, es importante que el analista de sistemas las comprenda. Las decisiones semiestructuradas (decisiones que se toman bajo riesgo), con frecuencia se apoyan en los sistemas de toma de decisiones. Cuando analiza las decisiones semiestructuradas, el analista los examina de acuerdo al grado de complejidad del problema y con el número de criterios considerados al llevar a cabo las decisiones.

El análisis de decisiones de criterio múltiple, que son (aquellas decisiones donde los factores tienen que equilibrarse) también es parte de esta etapa. Se disponen de muchas técnicas para el análisis de decisiones de criterio múltiple; incluyendo, entre otras, el proceso de intercambio y la aplicación de métodos de ponderado.

A esta altura del ciclo de vida del sistema, el analista prepara una propuesta del sistema que resume todo lo que ha encontrado, presenta un análisis costo-beneficio de las alternativas y plantea las recomendaciones, si es que existen, y la manera en que se plantea una recomendación depende de las características individuales de cada analista aunadas a su formación profesional.

TIPOS DE ANÁLISIS DE INFORMACIÓN

Los sistemas de información se desarrollan con diferentes propósitos, los cuales dependen de las necesidades de la empresa. Por ejemplo, tenemos los sistemas de procesamiento de transacciones (**TPS, Transaction Processing System**), los sistemas de información para la administración (**MIS, Management Information Systems**), y los sistemas de apoyo para la toma de decisiones (**DSS, Decision Support Systems**), que son diferentes tipos de sistemas de información computarizados que analizan y diseñan mediante la aplicación de los conceptos y las técnicas del diseño y análisis de sistemas.

Los TPS tienen como finalidad mejorar las actividades rutinarias de una empresa y de las que depende toda organización, las transacciones más comunes incluyen: facturación, entrega de mercancía, nóminas y procesos financieros. Los tipos de transacciones cambian en cada una de las diferentes organizaciones. Sin embargo, la mayor parte de las compañías procesan dichas transacciones como parte de sus actividades cotidianas. Las empresas con mayor éxito llevan a cabo este trabajo de forma ordenada y eficiente.

Los TPS incluyen entre otras las siguientes actividades:

- Clasificación
- Cálculos
- Ordenamiento
- Almacenamiento y recuperación
- Generación de resúmenes

Todas estas actividades forman parte del nivel operacional de cualquier organización. El estudio de un grupo de organizaciones de éxito también muestra la existencia de características similares, entre ellas:

1. Gran volumen de transacciones.
2. Gran similitud entre las transacciones.
3. Los procedimientos para el procesamiento de transacciones están bien comprendidos y se pueden describir con detalle.
4. Existen muy pocas excepciones a los procedimientos normales.

Estas características permiten establecer rutinas para el manejo de transacciones. Las rutinas describen qué buscar en cada transacción, los pasos y procedimientos a seguir, y lo que debe hacerse en caso de que se presente una excepción. Los procedimientos para el procesamiento de transacciones se denominan "procedimientos de operación estándar".

Los **sistemas de información para la administración (MIS)**, por su parte, no substituyen a los sistemas de procesamiento de transacciones, más bien todos se apoyan en éstos. Los sistemas de información para la administración se sustentan en la relación que surge entre las personas y las computadoras, requieren para su operación de: las personas, el software y el hardware; soportan un amplio espectro de tareas de las organizaciones, incluyendo el análisis y la toma de decisiones.

Los usuarios de MIS utilizan una base de datos compartida para tener acceso a la información. Dicha base de datos almacena tanto datos como modelos que le ayudan al usuario en la interpretación y el uso de la información que eventualmente se utiliza en la toma de decisiones. Este tipo de sistemas puede llegar a unificar ciertas funciones informáticas de la empresa, aunque el sistema como tal no exista dentro del negocio como una estructura individual.

Los sistemas de apoyo para la toma de decisiones (DSS) conforman el tercer tipo de sistemas de información computarizada, toman fundamentos de los sistemas descritos anteriormente, -sobre todo en el sentido de que ambos dependen de una base de datos como fuente de información-, pero se distinguen de los MIS al hacer énfasis en el soporte en cada una de las etapas de la toma de decisiones. Sin embargo, la decisión en sí depende de la persona responsable de la misma. Estos sistemas se diseñan con orientación a la persona o grupos de personas que los utilizarán, y no como los tipos anteriores que se diseñan para desahogar un proceso tedioso o para la toma de decisiones estructuradas, es decir aquella decisión donde existen procedimientos claros para tomarla y donde es posible identificar con anticipación, todos los factores que deben considerarse para la decisión.

HERRAMIENTAS PARA EL ANÁLISIS

En términos generales, una herramienta es cualquier dispositivo que, cuando se emplea en forma adecuada, mejora el desempeño de una tarea, tal como el desarrollo de sistemas basados en computadora.

En general las herramientas de software se agrupan en las siguientes categorías: análisis, diseño y desarrollo.

HERRAMIENTAS DE ANÁLISIS

Estas herramientas ayudan a los especialistas en sistemas a documentar un sistema existente, ya sea éste manual o automatizado, y a determinar los requerimientos de una nueva aplicación. Estas herramientas incluyen:

Herramientas para la recolección de datos. Capturan detalles que describen sistemas y procedimientos en su uso. Documentan procesos y actividades de decisión. Se utilizan para apoyar la tarea de identificar requerimientos.

Herramientas para diagramación. Crean representaciones gráficas de sistemas y actividades. Apoyan el dibujo y la revisión de diagramas de flujo de datos e iconos asociados con el análisis estructurado. Así mismo incluyen representación para programas de diagramas de flujo.

Herramientas para el diccionario. Registran y mantienen descripciones de los elementos del sistema, tales como grupos de datos, procesos y almacenamiento de datos. Con frecuencia proporcionan capacidad de examinar las descripciones del sistema para decidir si son incompletas o inconsistentes.

Las herramientas con mayor utilidad, en cualquier categoría, están siendo ya automatizadas tanto para mejorar la eficiencia del analista como para permitir obtener del esfuerzo de análisis, resultados más completos y exactos.

De manera general, el uso de las herramientas anteriormente mencionadas tiende a facilitar la elaboración de la documentación y la posterior toma de decisiones. En este último sentido contamos con tres herramientas básicas: árboles de decisión, tablas de decisión y lenguaje estructurado.

Árboles de decisión. Los árboles de decisión son representaciones gráficas y secuencias de las variables de decisión que indican qué condiciones considerar en primera instancia, cuáles en segunda y así sucesivamente. La raíz del árbol de decisión es el punto donde se inicia el análisis de una situación específica; las ramas indican la secuencia de decisiones que conducen a la acción apropiada que es necesario emprender.

Tablas de decisión. Las tablas de decisión relacionan condiciones y acciones por medio de reglas de decisión. Una regla de decisión establece las condiciones que deben de satisfacerse para emprender un grupo particular de acciones. La regla de decisión incorpora todas las condiciones que al mismo tiempo deben de ser verdaderas y no sólo una condición.

Lenguaje estructurado. El lenguaje estructurado se emplea para declarar reglas de decisión. Los tres tipos de declaraciones son las siguientes: estructuras de secuencia, estructuras de decisión, y estructuras de interacción. Estas declaraciones muestran acciones incondicionales, acciones repetitivas y acciones que ocurren sólo cuando se presentan ciertas condiciones.

El lenguaje estructurado ofrece un camino conciso para resumir un procedimiento donde se deben tomar decisiones y emprender acciones. Por otra parte, los resultados pueden ser revisados con facilidad por otras personas para detectar y corregir tanto errores como equivocaciones. Los errores que permanezcan en el análisis después de realizar la investigación de sistemas, aparecerán durante el diseño e implantación, donde será mucho más costoso corregirlos.

Como última fase del análisis se prepara la documentación que contiene los resultados de esta etapa: la clasificación e interpretación de los hechos; el diagnóstico de la problemática, los requerimientos y las propuestas para mejorar el sistema.

La documentación del análisis sirve como base para la siguiente fase del ciclo de vida del sistema: el diseño.

DISEÑO DE LOS SISTEMAS DE INFORMACIÓN

El diseño de sistemas se considera como la base de las etapas posteriores. Si no se considerara la etapa de diseño se construiría un sistema inestable, un sistema que tendría problemas a la hora de hacer pequeños cambios, un sistema que sería difícil de probar y cuyas medidas de calidad serían difíciles de evaluar. Nótese en la figura 1.2. que la etapa de diseño es el soporte de las siguientes.

Al diseño lo podemos definir de muchas maneras por la gran cantidad de actividades que lleva a cabo y a los diversos enfoques que se le dan. No obstante, podemos decir que el diseño es el proceso de aplicar distintas técnicas y principios con el propósito de planificar, reemplazar o complementar un sistema ya existente con los suficientes detalles como para permitir su realización física.

El diseño de sistemas de información es la primera etapa de tres actividades dentro de la implantación de un sistema: diseño, desarrollo y prueba. En la fase de diseño se consideran

los reportes y demás salidas que el sistema debe producir, estableciendo la forma en la que éste cumplirá con los requerimientos identificados en la etapa de análisis.

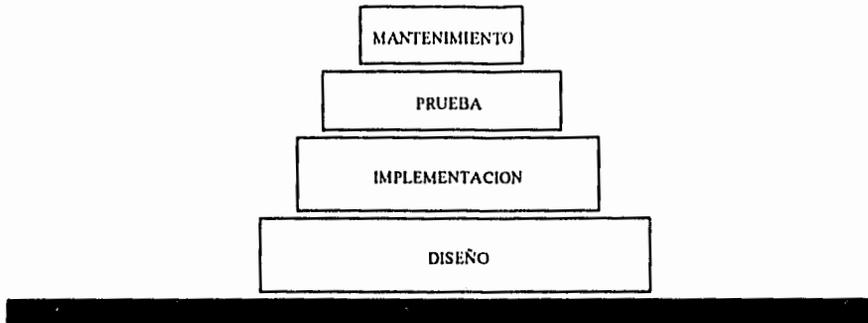


FIGURA 1.2.
IMPORTANCIA DEL DISEÑO

El diseño es técnicamente la parte central de la ingeniería de software. Durante la etapa de diseño se desarrollan, se revisan y se documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones del software cuya calidad se puede evaluar. En esta etapa podemos traducir con precisión los requisitos del cliente en un producto o sistema acabado.

La etapa de diseño comienza con una evaluación de la especificación del sistema y de la especificación de requisitos de software.

PROCESO DE DISEÑO

El proceso de diseño de sistemas comienza con una descripción del diseño arquitectónico, procedural y de datos. Esto es, se desarrolla una estructura modular, se definen las interfases y se establece la estructura de los datos. ¹El diseño de datos es la más importante de las tres actividades del proceso de diseño, ya que transforma el modelo del campo de información, creado durante el análisis, en las estructuras de datos que se van a requerir para implementar el software.

Wasserman A. en su libro "Information System Design Methodology" propone una serie de principios que se pueden usar para la especificación y diseño de datos:

1. Los principios sistemáticos de análisis aplicados a la función y el comportamiento también deben aplicarse a los datos.

¹ Es una representación de la relación lógica existente entre los elementos individuales de datos.

2. Deben identificarse todas las estructuras de datos y las operaciones que se han de realizar sobre cada una de ellas.
3. Debe establecerse y usarse un diccionario de datos para definir el diseño de los datos y del programa.
4. Se deben posponer las decisiones de datos de bajo nivel hasta más adelante en el proceso de diseño.
5. La representación de una estructura de datos sólo debe ser conocida por los módulos que hagan un uso directo de los datos contenidos en la estructura.
6. Se debe desarrollar una biblioteca de estructuras de datos útiles y de las operaciones que se les pueden aplicar.
7. El diseño de software y el lenguaje de programación deben soportar la especificación y la realización de tipos abstractos de datos.

El diseño abarca tres aspectos: el arquitectónico, el procedimental y el de datos.

El **Diseño Arquitectónico** define las relaciones entre los principales elementos estructurales del programa. Su objetivo principal es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. El diseño arquitectónico mezcla la estructura de programas y la estructura de datos definiendo las interfases que faciliten el flujo de los datos a lo largo del programa.

El **Diseño Procedimental** transforma los elementos estructurales en una descripción procedimental del software. Dicho diseño se realiza después de que se ha establecido la estructura del programa y de los datos. Se deberá especificar los detalles de los procedimientos sin ambigüedad.

El **Diseño de Datos** es la selección de las representaciones lógicas de los objetivos de datos (estructuras de datos), identificados durante la etapa de definición y especificación de requisitos (fase de análisis).

Dentro del contexto de diseño detallado muchas aplicaciones modernas requieren una actividad distinta: el diseño de la interfase. El diseño de la interfase establece la disposición y los mecanismos para la interacción hombre-máquina.

Desde el punto de vista de gestión de proyecto, al diseño de software lo podemos dividir en dos:

1.- Diseño Preliminar, enfocado en la transformación de los requisitos en los datos y la arquitectura del software.

2.- Diseño Detallado dirigido al refinamiento de la representación arquitectónica que lleva la estructura de datos detallada y a las representaciones algorítmicas del software.

EVALUACIÓN DE LA CALIDAD DE SOFTWARE

Para evaluar la calidad de una representación del diseño debemos establecer los siguientes criterios:

1.- Un diseño debe exhibir una **organización jerárquica** que haga un uso inteligente del control entre los componentes del software.

2.- Un diseño debe ser **modular**, esto es , el software debe estar dividido de forma lógica en elementos que realicen funciones y subfunciones específicas .

3.- Un diseño debe contener **representaciones distintas y separadas de los datos y de los procedimientos.**

4.- Un diseño debe llevar a **módulos que exhiban características funcionales independientes.**

5.- Un diseño debe llevar a **interfases que reduzcan la complejidad** de las conexiones entre los módulos y el entorno exterior.

6.- Un diseño debe **obtenerse mediante un método que sea reproducible** y que esté conducido por la información obtenida durante el análisis de los requisitos del software.

LA ETAPA DE DESARROLLO

La etapa de desarrollo es la que se encarga de traducir el conjunto de requisitos y especificaciones del diseño en el elemento operativo del sistema que llamamos software, es decir, su función primordial es la **codificación**.

Basándose en las especificaciones de diseño, se lleva a cabo la generación de un programa en un lenguaje de programación apropiado.

Los encargados de desarrollar el software pueden instalar software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los programadores. Por regla general los programadores (o analistas programadores) que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales.

Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

Dentro de la etapa de desarrollo se pueden utilizar herramientas automatizadas, que ayudan al analista a trasladar los diseños en aplicaciones funcionales. Existen tres categorías de herramientas para el desarrollo:

- **Herramientas para ingeniería de software.** Estas apoyan el proceso de formular diseños de software, incluyendo procedimientos y controles, así como la documentación correspondiente.
- **Generadores de código.** Producen el código fuente y las aplicaciones a partir de especificaciones funcionales bien articuladas.
- **Herramientas para pruebas.** Apoyan la fase de evaluación de un sistema o de partes del mismo contra las especificaciones. Incluyen facilidades para examinar la correcta operación del sistema así como el grado de perfección alcanzado en comparación con las expectativas.

El empleo de herramientas de programación adecuadas es un factor muy importante para el éxito del sistema, por lo que examinaremos algunas a continuación:

Lenguajes de Cuarta Generación

Los lenguajes de cuarta generación fueron creados para ayudar a satisfacer la necesidad de desarrollar software con mayor eficiencia. Estos lenguajes incluyen un amplio espectro de lenguajes de computadora que hacen hincapié sobre lo que debe hacerse más que sobre cómo realizar la tarea. Las especificaciones de los programas se desarrollan con un nivel mucho mayor que el encontrado en los lenguajes de tercera generación.

Los lenguajes de cuarta generación se clasifican en tres categorías: lenguajes no orientados hacia procedimientos, lenguajes de consulta y recuperación, y lenguajes generadores de reportes.

Lenguajes no orientados hacia los procedimientos

El lenguaje con el que trabajan los analistas y usuarios finales no está orientado hacia los procedimientos. Un solo mandato (por ejemplo SORT, SELECT, LOCATE) lleva a cabo una función completa. Un mandato de este tipo puede reemplazar el equivalente de más de cien instrucciones de un lenguaje de tercera generación.

Los lenguajes de este tipo por lo general forman parte de los lenguajes de consulta y recuperación, así como de los generadores de reportes.

Lenguajes de consulta y recuperación

Los lenguajes de consulta y recuperación (o simplemente lenguajes de consulta) facilitan la recuperación de datos almacenados sin necesidad de escribir muchas instrucciones orientadas hacia procedimientos, o especificar el formato de los datos. Estos lenguajes permiten a los usuarios formular preguntas -consultas- en formatos tabulares o parecidos al inglés.

Algunos lenguajes en esta categoría permiten a los usuarios dar entrada a datos y actualizar archivos o bases de datos.

Generadores de reportes

Los generadores de reportes permiten a los usuarios obtener con facilidad datos de archivos o bases de datos. Se puede obtener el contenido parcial o total de los registros. Estos dan a los usuarios mayor control sobre la apariencia y contenido de la salida. Los resultados se pueden presentar en un formato de reporte que se establece en forma automática por software, o el usuario también puede proporcionar las especificaciones que instruyan al sistema para preparar títulos específicos, descripciones de página y encabezados de columnas.

Es común que los usuarios deseen resúmenes de los totales con respecto a las categorías de actividades en la forma de totales y subtotales, para esto, los generadores de reportes requieren únicamente que los usuarios especifiquen el campo sobre el que debe calcularse el subtotal; el software lleva a cabo los procedimientos de manera automática. También se pueden manejar de esta manera operaciones aritméticas y lógicas.

Generadores de aplicaciones

Los lenguajes de cuarta generación están orientados hacia la producción de salidas, ellos preparan y muestran reportes impresos. En contraste, los generadores de aplicaciones son programas de software que permiten la especificación de toda una aplicación en un nivel muy alto. Estos proporcionan las condiciones para desarrollar aplicaciones que acepten datos, efectúen cálculos, sigan complicadas rutinas de procesamiento lógico y produzcan reportes y salidas. El generador de aplicaciones produce el código fuente. Algunos producen programas completos. Otros, denominados generadores de programas, preparan parte del código del programa como módulos individuales, y permiten al usuario enlazar

otros módulos con los producidos por el generador. Estos generadores de programas son una forma especial, un subconjunto de los generadores de aplicaciones.

Generadores de pantallas

Desde el punto de vista del desarrollo, la creación de un formato para pantalla es una de las áreas que más consume tiempo y donde existe mayor propensión a cometer errores. Esto es debido a que las pantallas son la forma en que los usuarios interactúan con una aplicación. Este proceso incluye las tareas de acomodar encabezados, instrucciones y campos de datos sobre la pantalla, así como establecer los procedimientos para aceptar los datos que entran al sistema por medio del teclado, al mismo tiempo que muestran sobre la pantalla los datos proporcionados.

Un generador de pantalla es una herramienta interactiva para "dibujar" pantallas y efectuar la validación automática de la entrada y procesamiento. Es posible seleccionar con respuestas sencillas preferencias sobre el presentar con mayor brillantez la información más importante, el utilizar determinados colores o hacer uso del video inverso.

Los generadores de pantalla también permiten que los usuarios preparen automáticamente componentes que sean de ayuda en la interacción usuario-máquina, incluyendo la localización de campos para entrada de datos, campos para presentar datos, encabezados de columna, etiquetas y mensajes.

Sistemas de diccionario de datos

Los sistemas de diccionarios de datos guardan definiciones y descripciones de los datos utilizados en los sistemas de información de la organización. Apoyarse en estas definiciones es benéfico para el desarrollo de sistemas, ya que se ahorra tiempo, pues no se necesitan determinar las descripciones apropiadas para los datos, en lugar de esto se emplean las que se encuentran en el diccionario.

El diccionario de datos contiene descripciones de todos los datos utilizados en el sistema, las características lógicas de los sitios donde se almacenan los datos del mismo, incluyendo nombre, descripción, alias, contenidos y organización. También identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.

Algunos sistemas de diccionario de datos ofrecen características similares a las que se encuentran en los generadores de aplicaciones.

Bibliotecas de código reutilizable

Un conjunto de módulos individuales de código que fueron desarrollados para emplearse en otros sistemas, pero que son reutilizables, forma una biblioteca que es una herramienta muy valiosa para el desarrollo de sistemas. Si es necesario se pueden recuperar y modificar varios módulos de código reutilizable. Al insertar estos módulos en el sistema se crean sistemas que funcionen con rapidez y con un costo relativamente bajo.

Existe otro tipo de herramientas que nos facilitan la generación de código, éstas son las herramientas asistidas por computadora o herramientas automatizadas.

La generación de código para computadoras es una tarea que realizan mejor las computadoras que las propias personas, ya que las reglas de generación se pueden aplicar en forma consistente y exacta.

Herramientas de tipo back-end

Las herramientas de tipo back-end tienen como finalidad ayudar al analista a formular la lógica del programa, los algoritmos de procesamiento y la descripción física de datos, también ayudan a la interacción con los dispositivos para entrada y salida. Estas actividades convierten los diseños lógicos del software en un código de programación que es el que finalmente da existencia a la aplicación. Dado que su empleo está destinado al desarrollo de software, este tipo de herramientas también se conoce como herramientas para programación asistida por computadora.

Herramientas integrales

Las herramientas integrales proporcionan un ambiente que automatiza tareas clave a lo largo de todo el proceso de desarrollo. Estas herramientas abarcan todo el ciclo de vida de la aplicación, no sólo el proceso de desarrollo. Presentan facilidades para manejar aspectos de análisis y desarrollo, también facilitan el diseño, administración y mantenimiento del código. Asimismo, brindan un ambiente eficiente para crear, almacenar, manipular, administrar y documentar sistemas.

Herramientas asistidas por computadora para la ingeniería de sistemas (CASE).

El objetivo de las herramientas CASE es automatizar los aspectos clave de todo el proceso de desarrollo, desde el principio hasta el fin. Las herramientas de tipo CASE incluyen cinco componentes: herramientas para diagramación, un dispositivo de información, generadores de interfases, generadores de código y herramientas de administración. Por ahora sólo hablaremos de los componentes que actúan en forma más directa sobre la etapa de desarrollo o codificación.

Generador de interfases

Las interfases con el sistema son los medios que permiten a los usuarios interactuar con una aplicación, ya sea para dar entrada a información y datos o para recibir información. Los generadores de interfases ofrecen la capacidad para preparar imitaciones y prototipos para las interfases con los usuarios. Por lo general, soportan la rápida creación de menús de demostración para el sistema, de pantallas de presentación y del formato de los informes.

Generadores de código

Los generadores de código automatizan la preparación de software. Éstos incorporan métodos que permiten convertir las especificaciones del sistema en código ejecutable.

La generación de código aún no ha sido perfeccionada. Los mejores generadores de código producen aproximadamente el 75% del código fuente de una aplicación. El resto debe ser escrito por los programadores. La codificación manual, que es el nombre que recibe este proceso, sigue siendo necesaria.

Dado que las herramientas CASE son de propósito general, es decir no están limitadas a ciertas áreas específicas de aplicación, resulta que el desafío de automatizar el proceso de generación de software es sustancial.

Los mayores beneficios se obtienen cuando los generadores de código se encuentran integrados con un depósito central de información, que no es otra cosa más que el diccionario de datos. Esta combinación alcanza el objetivo de crear un código que pueda volverse a emplear. Cuando las especificaciones cambian, se puede volver a generar el código al alimentar los detalles del diccionario de datos a través del generador de código. El contenido del diccionario puede emplearse de nuevo para preparar el código ejecutable.

Se debe tener mucho cuidado al seleccionar las herramientas para el desarrollo, ya que tienen un gran poder al aumentar la productividad disminuyendo la cantidad de tiempo necesario para documentar, analizar y desarrollar sistemas de información. Si no se usan adecuadamente puede ser fatal en todo el ciclo de vida de la aplicación.

Una vez que se ha codificado el sistema previamente diseñado, se procede a efectuar una las fases del ciclo de vida del sistema que consumen más tiempo y esfuerzo: las pruebas.

PRUEBAS Y LIBERACIÓN

La fase de pruebas, si la definimos de la manera más simple, consiste en emplear de manera experimental el sistema, para asegurarse que las especificaciones, el diseño y la codificación no tenga fallas, es decir, para descubrir errores y comprobar que funciona de acuerdo con las especificaciones obtenidas en etapas previas; y que trabaja como los usuarios esperan que lo haga. Se recomienda que sean varios usuarios e incluso personas ajenas al grupo que escribió los programas originales, los que utilicen el sistema, de tal manera que la organización descubra cualquier falla, que las pruebas sean completas e imparciales y que el software sea confiable, antes de que la organización lo implante y dependa de él.

Diversos autores distinguían antiguamente en el proceso de prueba 5 etapas:

1. Prueba de Funciones
2. Prueba de Módulos
3. Prueba de Subsistemas
4. Prueba del Sistema (de Integración)

5. Prueba de Aceptación

Sin embargo, las técnicas y estrategias de prueba no estaban hasta entonces plenamente definidas, pues la Ingeniería de Software a veces era desplazada por las formas antiguas, empíricas y de sentido común. Sin embargo, y gracias al desarrollo que ha tomado la Ingeniería de Software, algunas gentes como Roger S. Pressman ahora dan énfasis a las Técnicas de Diseño de Casos de Prueba y luego utilizan las Estrategias de Prueba, que integran las técnicas en una serie de pasos bien planificados que dan como resultado una correcta construcción del software.

Técnicas de Diseño de Casos de Prueba

Tienen como objetivo derivar un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Y para llevar a cabo este objetivo se utilizan dos categorías diferentes de técnicas:

- a) Prueba de la Caja Blanca y,
- b) Prueba de la Caja Negra.

Pruebas de la Caja Blanca

Las pruebas de la caja blanca se centran en la estructura de control del programa, Hetzel las define como "prueba a pequeña escala", porque casi siempre van dirigidas a pequeños componentes de programas, es decir, módulos o grupos de módulos.

La Prueba del **Camino Básico**, propuesta inicialmente por Tom McCabe, es una técnica de la caja blanca que hace uso de representaciones gráficas de las estructuras de control lógico dentro del programa (p. Ej. If - then- else, do- while, until, case), definiendo luego matrices que representan las conexiones entre puntos de control, para derivar el conjunto de caminos linealmente independientes que aseguren la cobertura.

La **Prueba de Condiciones** y de Flujo de Datos es una prueba de caja blanca, enfocada a las estructuras de control que ejercita las condiciones lógicas y relacionales contenidas en módulos de programa.

La **Prueba de Bucles** analiza la diversidad de ellos en diversos grados de complejidad (simples, anidados, concatenados, no estructurados), puesto que los bucles son considerados como la piedra angular de la inmensa mayoría de los algoritmos implementados en software.

Pruebas de la Caja Negra

Las pruebas de la caja negra son diseñadas para validar los requisitos funcionales, es decir, los datos fundamentalmente, sin fijarse en el funcionamiento interno de un programa. Por ello se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba. El método de caja negra divide el campo de entrada de

un programa en clases de datos, para después derivar casos de prueba que ejerciten determinadas funciones del software.

El **Análisis de Valores Límite** prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables, por ejemplo, si una condición de entrada especifica un "rango" delimitado por los valores a y b, se deben diseñar casos de prueba para los valores a y b y para los valores justo por debajo y justo por encima de a y b, respectivamente.

Las **Técnicas de Representaciones Gráficas de Causa-Efecto** permiten al encargado de la prueba, validar complejos conjuntos de acciones y condiciones.

Finalmente, es necesario comentar las observaciones que algunos han hecho respecto a las pruebas, y es que "la prueba nunca termina, simplemente se transfiere del encargado del desarrollo al cliente. Cada vez que el cliente usa el programa, lleva a cabo una prueba". Aplicando el diseño de casos de prueba, el ingeniero de software puede conseguir una prueba más completa y descubrir y corregir así el mayor número de errores antes de que comiencen las "pruebas del cliente".

Estrategias de Prueba

El objetivo de la prueba de software, como ya dijimos, es descubrir errores. Para conseguir este objetivo se planifica y se ejecuta una serie de pasos de prueba de unidad, de integración, de validación y del sistema.

Las **Pruebas de Unidad y de Integración** se centran en la verificación funcional de cada módulo (prueba de unidad) y en la incorporación de los módulos en una estructura del programa (integración ascendente, descendente).

La **Prueba de Validación** demuestra el seguimiento de los requisitos del software, es decir, una vez que ha terminado la prueba de integración, el software está completamente ensamblado como un paquete. Se han encontrado y corregido los errores de interfase y puede comenzar la validación, que se dice se logra cuando el software funciona de acuerdo con las expectativas razonables del cliente, para ello los criterios a tomarse en cuenta se centran en la validez de requisitos (p. Ej.: portabilidad, compatibilidad, recuperación de errores, facilidad de mantenimiento), repaso de la configuración del software, también llamado "auditoría" del software.

La **Prueba del Sistema** es una estrategia que valida el software una vez que se ha incorporado en un sistema mayor, como son las *pruebas de recuperación* (del estado del sistema, de los datos, del arranque, etc.), *pruebas de seguridad* (para evitar penetraciones al sistema impropias o ilegales), *pruebas de resistencia* o también llamada de "sensibilidad", que son diseñadas para enfrentar a los programas a situaciones anormales, como por ejemplo, incrementar las frecuencias de datos de entrada en un orden de magnitud con el fin de comprobar cómo responden las funciones de entrada, o ejecutar casos de prueba que produzcan excesivas búsquedas de datos residentes en disco; la *prueba de*

rendimiento es otra estrategia de prueba del sistema diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

Depuración

La depuración se puede definir como el proceso de identificar las áreas del programa que causan errores y modificarlas para corregirlos. A diferencia de las estrategias de prueba, propiamente dichas (una actividad sistemática y planificada), la **depuración** se puede considerar un arte. A partir de una indicación sintomática de un problema, la actividad de depuración debe rastrear la causa del error. Estos errores se pueden originar en cualquiera de las fases del ciclo de vida del sistema, como lo indica la figura 1.3.

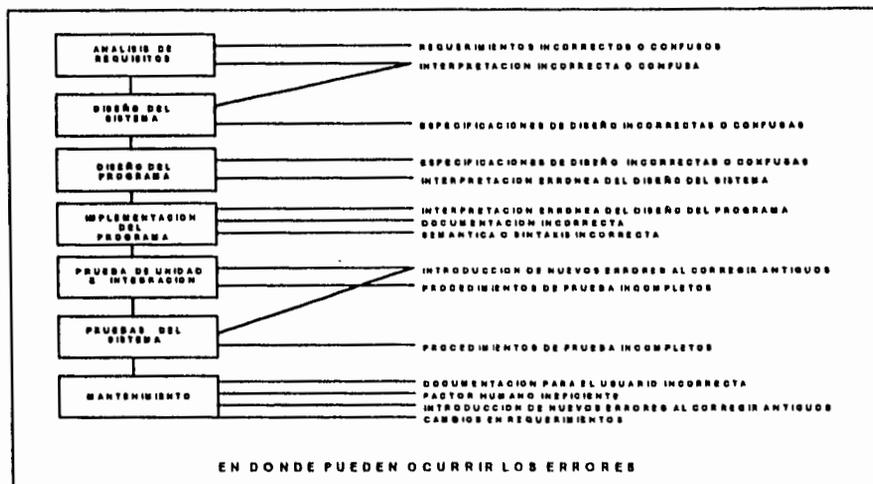


FIGURA 1.3.
POSIBLES ORÍGENES DE LOS ERRORES

La depuración implica 2 etapas:

1. Se deben localizar aquellas partes incorrectas del código del programa.
2. El programa se debe modificar de manera que cumpla con sus requisitos. Algunos instrumentos de depuración pueden ser:

- Los **programas de vaciado de símbolos**, que relacionan los nombres de los objetos utilizados en su programa con sus valores.
- Paquetes para el **Rastreo** de Programas.

- **Analizadores Estáticos de Programas**, que no necesitan la ejecución del programa, sino que cuando se lee el texto de un programa, rastrean el texto y buscan anomalías que pudieran provocar errores.
- **Analizadores Dinámicos de Programas**. Dentro de los recursos disponibles de la depuración, el más valioso es el apoyo de otros miembros de la plantilla de desarrollo del software, porque son los que están mejor relacionados con el sistema en esta etapa.

El requisito de que el software sea cada vez de mejor calidad demanda un planteamiento más sistemático de la prueba.

El **proceso de liberación** se refiere al proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y contruir todos los archivos de datos necesarios para utilizarla y empezar a trabajar, y dejar el campo abierto a los ajustes o mantenimiento.

MANTENIMIENTO

El mantenimiento, la última fase del proceso de ingeniería de software, consume la mayor parte del presupuesto destinado al software de computadora. Esto es entendible, pues el mantenimiento es considerado como todo aquello que se hace para cambiar el sistema, después de que está en operación.

Algunos piensan que el mantenimiento del software es parecido al mantenimiento del hardware, o sea, la reparación o prevención de partes dañadas o que funcionan impropriadamente. Sin embargo, el mantenimiento del software no se puede ver en la misma forma.

Con excepción de los casos más simples, la mayoría de los sistemas son evolutivos, es decir, una o más de las características que definen el sistema generalmente cambian durante la vida del mismo.

A medida que se desarrollan más programas, surge la siguiente tendencia: **la cantidad de esfuerzo y de recursos dedicados al mantenimiento crece**. Al final, algunas organizaciones de software pueden llegar a encontrar la barrera del mantenimiento, no siendo capaces de iniciar nuevos proyectos porque tienen todos sus recursos dedicados al mantenimiento de programas antiguos.

El mantenimiento se enfoca fundamentalmente en cuatro problemas:

1. Mantenimiento del *control sobre las funciones* del sistema día a día.
2. Mantenimiento del *control sobre las modificaciones* hechas al sistema.
3. *Perfeccionamiento de las funciones* existentes del sistema.

4. Prevención de la degradación del rendimiento del sistema a niveles inaceptables.

Estos cuatro problemas han definido cuatro tipos de mantenimiento: correctivo, adaptativo, perfectivo y preventivo, en la gráfica de la figura 1.4. se ilustra el porcentaje de esfuerzo requerido por cada tipo, sobre el esfuerzo total de mantenimiento.

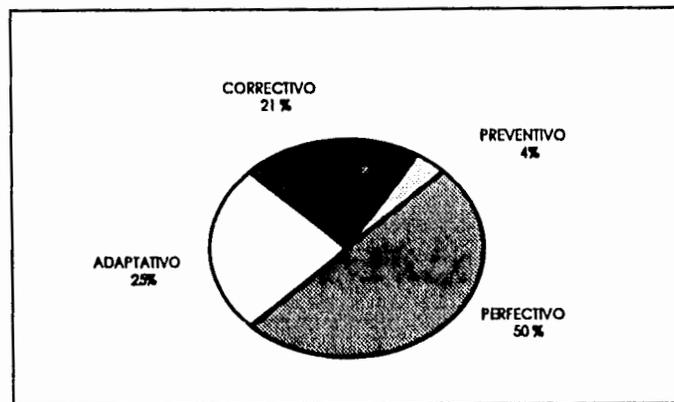


FIGURA 1.4.
DISTRIBUCIÓN DEL ESFUERZO EN MANTENIMIENTO

El Mantenimiento Correctivo

Actúa para corregir errores que no han sido descubiertos antes de poner en uso el software. O, visto de otra forma, según se detectan los errores, se van haciendo las reparaciones emergentes. Se puede implementar un amplio rango de cambios para corregir problemas más generales en procedimientos de diseño de código.

El Mantenimiento Adaptativo

Se aplica cuando los cambios del entorno externo precipitan las modificaciones del software. En algunas ocasiones, un cambio introducido en alguna parte del sistema hace necesario los cambios en otras partes de mismo. La implementación de estos cambios secundarios es lo que se define como mantenimiento adaptativo.

Los cambios hechos para agregar parámetros extras son también cambios adaptativos. Estos no corrigen errores, simplemente permiten al sistema adaptarse a un cambio en cualquier parte. Por ejemplo, suponiendo que un compilador se amplía por la adición de un "debugger", los menús o las definiciones de las teclas de función serán alteradas para que los usuarios del compilador tengan la opción de utilizar el debugger cuando compile un programa. Es decir, este menú y tecla modifican el sistema para acomodar la herramienta del debugg.

El Mantenimiento Perfectivo

Incorpora las mejoras solicitadas por la comunidad de usuarios. Las funciones del sistema se basan en un ambiente del mundo real en el cual operan. Los cambios en el ambiente se reflejarán en cambios en el sistema. Tales ampliaciones son lo que se conoce como mantenimiento perfectivo. Por ejemplo, un sistema operativo podrá requerir mantenimiento perfectivo al hacerse necesarios los cambios cuando se introduce un nuevo tipo de dispositivo de almacenamiento.

El Mantenimiento Preventivo

Mejora la futura facilidad de mantenimiento y la fiabilidad como base para las mejoras posteriores. Este tipo de mantenimiento es muy similar al que se realiza en el Hardware, al intentar prevenir un mal funcionamiento. Por ejemplo, la adición de un "reconocedor" o "checador" de tipo en la entrada de datos de un sistema. Este reconocedor permitirá a los módulos de entrada de datos detectar errores antes de pasar a otros módulos, en donde pueden afectar o "corromper" una base de datos.

Es importante mencionar que las herramientas de **ingeniería inversa** y de **reingeniería** llegarán a ser una parte importante del proceso de mantenimiento a finales de los 90. La **ingeniería inversa** extrae información del diseño del código fuente de un programa sin disponer de ninguna otra documentación. La **reingeniería** combina las características de extracción de información de análisis y diseño de la ingeniería inversa con una capacidad de reestructuración de los datos, la arquitectura y la lógica del programa, y de algún modo, una mayor facilidad de mantenimiento en el futuro.

COSTO DEL MANTENIMIENTO

El costo del mantenimiento de sistemas de software ha crecido rápidamente durante los últimos 20 años. Durante los setentas el mantenimiento constituía entre el 30 y 40% del presupuesto de una organización de sistemas de información. Actualmente, muchas compañías gastan cerca del 90% del presupuesto en el mantenimiento. Los costos de mantenimiento en dinero son los que más nos interesan. Sin embargo, otros costos, menos tangibles, pueden, en última instancia, ser la causa de muchas preocupaciones.

Un costo intangible del mantenimiento del software se encuentra en una oportunidad de desarrollo que se ha de posponer o que se pierde, debido a que los recursos disponibles deben estar dedicados a las tareas de mantenimiento. Otros costos intangibles son:

- Insatisfacción del cliente cuando una petición de reparación o de modificación aparentemente legítima no se puede atender en un tiempo razonable.
- Disminución de la calidad global del software debido a los errores latentes que introducen los cambios en el software mantenido.
- Trastornos en otros esfuerzos de desarrollo al tener que "poner" a trabajar la plantilla en tareas de mantenimiento.

El costo final del mantenimiento de software es una drástica reducción de la productividad (medida en líneas de código por persona entre mes o en puntos de función por persona entre mes), cuando se inicia el mantenimiento de viejos programas.

En la figura 1.5., se muestran los factores técnicos y no técnicos que afectan los costos de la etapa de mantenimiento de sistemas.

Factores que afectan los Costos de Mantenimiento	
No Técnicos	Técnicos
Aplicación en que se apoya	Interdependencia de los módulos
Estabilidad del personal	Lenguaje de Programación
El Tiempo de vida del programa (depende de su aplicación)	Estilo de Programación
La dependencia del programa de su ambiente externo	Comprobación y Prueba del programa
Estabilidad del Hardware (No Cambiarlo) su ambiente externo	Calidad y cantidad de la documentación del programa.

FIGURA 1.5.
FACTORES QUE AFECTAN EL COSTO DEL MANTENIMIENTO.

Por último, en la Fig. 1.6. se muestran en porcentajes de ocurrencia, algunas de las dificultades más comunes para llevar a cabo el mantenimiento de sistemas de información.

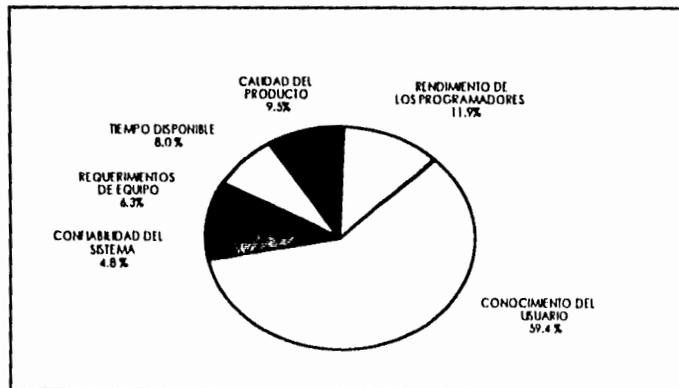


FIGURA 1.6.
DIFICULTADES MÁS COMUNES PARA LLEVAR A CABO EL MANTENIMIENTO

CONCLUSIONES

De este capítulo podemos concluir que para poder llevar a cabo la implementación de los sistemas de información primero se deberá realizar un análisis del problema, de las oportunidades y de los objetivos; identificar que información requiere el usuario para el desempeño de sus tareas así como determinar las necesidades propias del sistema.

Además se deberá analizar de que manera ingresan los datos, el flujo de los mismos y la salida de la información.

Finalmente, se deberá diseñar ya sea reemplazando o complementando un sistema ya existente, según los requerimientos de la empresa.

CAPÍTULO 2

PLANEACIÓN

Todo trabajo o proyecto requiere de una serie de pasos para alcanzar exitosamente los objetivos fijados, esto es, llevar a cabo un plan de actividades a realizar. El desarrollo de un sistema de información no es la excepción, aún más, debido a que la economía de toda empresa en la actualidad está basada en la información, quien la manipule de manera rápida, precisa y eficiente, será quien pueda desarrollarse y crecer. Las empresas invierten millones de pesos en la captura, almacenamiento y preservación de este recurso; esto hace que el proyecto de construir un sistema de información deba planearse cuidadosamente.

El presente capítulo proporciona una definición de lo que es la planeación, describe brevemente los principales métodos de planeación de sistemas de información: Planeación de Sistemas Empresariales (BSP), Planeación Estratégica de Arquitectura de Computadoras y Factores Críticos de Éxito. Asimismo, se trata de explicar la función de la planeación en este proyecto, estableciendo sus alcances y limitaciones. Por último, se presenta la planeación de un proyecto de desarrollo de un sistema de información, basado en el ciclo de vida de desarrollo de sistemas.

2.1. DEFINICIÓN DE PLANEACIÓN

La planeación consiste en fijar el curso concreto de las actividades que han de seguirse en el desarrollo de sistemas de información, estableciendo la secuencia de operaciones para realizarlas y las determinaciones de tiempo y recursos necesarios para llevarlas a cabo. Siempre habrá algo que no podrá planearse a detalle, por eso todo plan debe considerar tiempos de holgura y ser flexible para adaptarse a los cambios que surjan en éste, en razón de la parte imprevisible, o por las circunstancias que hayan variado conforme al avance del proyecto.

La planeación de un proyecto de *software* es una actividad cuyo objetivo es estimar los **recursos, costos y tiempos** que se verán involucrados durante el desarrollo del sistema de información, por lo que el planificador de un proyecto de *software* tiene que considerar, antes de que comience el proyecto, su duración, el esfuerzo requerido y el personal que lo llevará a cabo. Además, debe contemplar los recursos necesarios de *hardware* y *software*.

La planeación es un esfuerzo intelectual, requiere determinación consciente de los cursos de acción y que las decisiones que se tomen se basen en propósito, conocimiento y estimaciones determinadas.

2.2 MÉTODOS DE PLANEACIÓN

Las tres metodologías más utilizadas para la planeación de sistemas de información son:

- ⇨ El Método de Planeación de Sistemas Empresariales (BSP) de IBM,
- ⇨ El Método de Planificación Estratégica de Arquitectura de Computadoras de Nolan, Norton & Co., y
- ⇨ El Método de Factores Críticos de Éxito.

El método **BSP de IBM** es uno de los más utilizados, se concentra en la identificación de los datos necesarios para poner en marcha a una organización.

El método de **Planificación Estratégica** enlaza la capacidad actual de la organización con sus necesidades futuras. Este método recalca la necesidad de desarrollar una fuerte infraestructura técnica, que sirva como plataforma para el desarrollo de aplicaciones.

El método de **Factores Críticos de Éxito** busca identificar las áreas clave de la organización y asegurar su incorporación a los sistemas de información.

Cualquiera de éstos métodos contempla en mayor o menor medida los siguientes aspectos:

- ⇨ **Estimación**
Es el pronóstico de tiempos y recursos necesarios para concluir un proyecto. Toda estimación conlleva un grado de incertidumbre. Se llevan a cabo estimaciones pesimistas y optimistas, elaborándose un promedio que determina la duración de cada actividad del proyecto.

2.1. DEFINICIÓN DE PLANEACIÓN

La planeación consiste en fijar el curso concreto de las actividades que han de seguirse en el desarrollo de sistemas de información, estableciendo la secuencia de operaciones para realizarlas y las determinaciones de tiempo y recursos necesarios para llevarlas a cabo. Siempre habrá algo que no podrá planearse a detalle, por eso todo plan debe considerar tiempos de holgura y ser flexible para adaptarse a los cambios que surjan en éste, en razón de la parte imprevisible, o por las circunstancias que hayan variado conforme al avance del proyecto.

La planeación de un proyecto de *software* es una actividad cuyo objetivo es estimar los recursos, costos y tiempos que se verán involucrados durante el desarrollo del sistema de información, por lo que el planificador de un proyecto de *software* tiene que considerar, antes de que comience el proyecto, su duración, el esfuerzo requerido y el personal que lo llevará a cabo. Además, debe contemplar los recursos necesarios de *hardware* y *software*.

La planeación es un esfuerzo intelectual, requiere determinación consciente de los cursos de acción y que las decisiones que se tomen se basen en propósito, conocimiento y estimaciones determinadas.

2.2 MÉTODOS DE PLANEACIÓN

Las tres metodologías más utilizadas para la planeación de sistemas de información son:

- ⇒ El Método de Planeación de Sistemas Empresariales (BSP) de IBM,
- ⇒ El Método de Planificación Estratégica de Arquitectura de Computadoras de Nolan, Norton & Co., y
- ⇒ El Método de Factores Críticos de Éxito.

El método **BSP de IBM** es uno de los más utilizados, se concentra en la identificación de los datos necesarios para poner en marcha a una organización.

El método de **Planificación Estratégica** enlaza la capacidad actual de la organización con sus necesidades futuras. Este método recalca la necesidad de desarrollar una fuerte infraestructura técnica, que sirva como plataforma para el desarrollo de aplicaciones.

El método de **Factores Críticos de Éxito** busca identificar las áreas clave de la organización y asegurar su incorporación a los sistemas de información.

Cualquiera de éstos métodos contempla en mayor o menor medida los siguientes aspectos:

- ⇒ **Estimación**
Es el pronóstico de tiempos y recursos necesarios para concluir un proyecto. Toda estimación conlleva un grado de incertidumbre. Se llevan a cabo estimaciones pesimistas y optimistas, elaborándose un promedio que determina la duración de cada actividad del proyecto.

⇨ **Análisis de Riesgos**

Partiendo de que un riesgo es una posible contingencia que afectará el desarrollo de actividades de un proyecto, el análisis de riesgos contempla la identificación, proyección y evaluación de éstos, define su probabilidad de aparición y proyecta su impacto si ocurrieran. El riesgo se mide por el grado de incertidumbre de las estimaciones establecidas de recursos, costos y duración.

⇨ **Seguimiento y Control**

Se refiere a la determinación de las actividades que se pueden realizar en forma simultánea, así como de aquellas que necesitan la finalización de una actividad previa para poder desarrollarse. Por otro lado, el control significa retroalimentación para comparar lo realizado con lo planeado y, de esta forma, poder acelerar o reprogramar las actividades para que concluyan a tiempo.

Opciones para la realización de estimaciones:

Existen diferentes métodos para llevar a cabo las estimaciones de un proyecto de *software*:

- ⇨ **Técnicas de descomposición** para generar estimaciones de costo y esfuerzo del proyecto.
- ⇨ **Modelos empíricos de estimación de recursos**
- ⇨ **Herramientas automatizadas.**

Técnicas de descomposición. Se descompone el problema en un conjunto de pequeños problemas, esperando que sean más manejables.

En este grupo están las técnicas de estimación de líneas de código (LDC) y los puntos de función¹ (PF). Estas técnicas difieren en el nivel de detalle que requiere la descomposición. Entre sus características comunes están:

1. Se debe comenzar con una definición restringida del ámbito de *software*.
2. A partir de lo anterior, se intenta descomponer el *software* en pequeñas subfunciones que puedan ser estimadas individualmente
3. Se evalúa la variable de estimación (LDC ó PF) para cada subfunción.
4. Se comparan las estimaciones obtenidas con estándares o métricas de productividad y se deriva el costo y el esfuerzo para la subfunción.
5. Combinando las estimaciones de las subfunciones se produce la estimación total para el proyecto entero.

Modelos empíricos de estimación de recursos. Un modelo de estimación para el desarrollo de *software* utiliza fórmulas derivadas empíricamente para predecir los recursos que se requieren en la planeación del proyecto. Los datos empíricos en los que se basan la mayoría de estos modelos se obtienen de una muestra de proyectos limitada, por lo que

¹ Puntos de función (PF). Son indicadores numéricos de la funcionalidad o utilidad del *software*, como número de entradas y salidas, número de llamadas del usuario, número de archivos y de interfaces externas.

un mismo modelo de estimación no siempre es apropiado para todas las clases de *software* o para todos los entornos de desarrollo.

Existen cuatro clases de estos modelos:

1. **Modelos univariantes estáticos.** Toman la forma $\text{recurso} = c1 \times (\text{característica estimada})^c2$. Las constantes $c1$ y $c2$ se derivan de los datos recopilados de anteriores proyectos. La característica estimada pudiera ser: cantidad de líneas de código fuente, el esfuerzo u otra característica de *software*.
2. **Modelos multivariantes estáticos.** Un modelo de este tipo es $\text{recurso} = c1e1 + c2e2 + c3e3 + \dots$, también usa datos históricos; $c1$, $c2$ y $c3$ son constantes obtenidas empíricamente para la característica i ésima, e_i es la característica i ésima del *software*.
3. **Modelos multivariantes dinámicos.** Proyecta los requisitos de recursos como una función del tiempo. Los recursos se definen en una serie de pasos consecutivos, se asigna cierto porcentaje de esfuerzo o de otro recurso a cada etapa del proceso de ingeniería de *software*.
4. **Modelos teóricos.** Examinan el *software* desde un punto de vista microscópico, o sea, las características del código fuente (por ejemplo número de operadores y operandos).

Herramientas automatizadas. Son programas que utilizan las técnicas de descomposición y los modelos empíricos de estimación de *software*. Permiten estimar costos y esfuerzos y llevar a cabo análisis del tipo "¿qué pasa si?" con variables importantes del proyecto como la fecha de entrega o la selección de personal. Algunas de estas herramientas proporcionan también calendarización de desarrollo y riesgos asociados. Como ejemplos de estos programas están: BYL (Before You Leap), WICOMO y Decplan.

Factores que afectan la precisión de las estimaciones

Por la naturaleza de la estimación, existen algunos factores que pueden afectar la precisión y confiabilidad de los resultados:

- **La complejidad del proyecto.** Se ve afectada por la familiaridad del grupo de trabajo con anteriores proyectos. Así, una determinada aplicación de tiempo real parecerá muy complicada a un staff especializado en aplicaciones no interactivas, mientras que parecerá muy sencillo para aquellos con experiencia en controles de procesos de alta velocidad.
- **Tamaño del proyecto.** A medida que aumenta el tamaño, la interdependencia entre los distintos elementos del *software* crece rápidamente. La descomposición del problema se hace más difícil, dado que los elementos descompuestos pueden ser todavía muy grandes y por lo tanto, la estimación se torna más complicada.
- **Estructuración del proyecto.** Se refiere a la facilidad de subdividir las funciones del sistema y a la naturaleza jerárquica de la información que debe ser procesada. A medida

que aumenta el grado de estructuración, la posibilidad de estimar con precisión mejora y el riesgo disminuye.

Qué es lo que se estima

Los recursos que se consideran para llevar a cabo el desarrollo de *software* son:

- ◊ **Recursos humanos.** El personal que desarrollará el sistema de información.
- ◊ **Recursos de *Hardware*.** Plataforma y equipo que nos permite generar los programas de aplicación.
- ◊ **Recursos de *Software*.** El *software* que se utilizará para el desarrollo del nuevo sistema. Hoy en día, el *software* es el elemento más caro de la mayoría de los sistemas de información. Un error en la estimación del costo puede ser lo que marque la diferencia entre los beneficios y las pérdidas.

Toda vez que el *software* interactúa con otros elementos del sistema de información, se debe considerar la naturaleza y la complejidad de cada interfaz para determinar cualquier efecto sobre los recursos, costos y calendario de desarrollo. (Las interfaces abarcan: *hardware* que ejecutará el *software* en perspectiva, *software* ya existente que debe enlazarse con el nuevo, usuarios y procesos que preceden o suceden al *software*).

Análisis de Riesgos

Al planear un proyecto es necesario que se consideren los posibles riesgos que pueden hacer que éste fracase o afecten de tal manera que se modifiquen los plazos, los costos y la valoración de recursos que se había estimado en un principio.

El análisis de riesgos consta de cuatro actividades:

1. Identificación del riesgo
2. Proyección del riesgo
3. Evaluación del riesgo
4. Gestión y Supervisión del riesgo

Identificación del riesgo

Consiste en enumerar los riesgos concretos de un proyecto en sus diferentes categorías. Se puede tratar de riesgos del proyecto, riesgos técnicos y riesgos del negocio.

Los **riesgos del proyecto** identifican probables problemas presupuestarios, de calendarización, de personal, de recursos, del cliente y de requisitos. Así como su impacto sobre el proyecto de *software*.

Los **riesgos técnicos** se refieren a problemas potenciales de diseño, implantación, interfaz, verificación y mantenimiento.

Los **riesgos de negocio** son los que con mayor probabilidad afectarán al proyecto, y dentro de éstos encontramos a los siguientes:

- a) Construcción de un sistema que en realidad nadie quiere;
- b) Construcción de un sistema que no se ajusta a la estrategia global de la empresa;
- c) Pérdida del líder del proyecto por cambios de enfoque o de personal, y
- d) Pérdidas presupuestarias.

Proyección del riesgo

También denominada estimación del riesgo, en esta actividad se consideran: la **probabilidad de que el riesgo sea real** y las **consecuencias** de los problemas asociados con el mismo, suponiendo que aparece. Las cuatro actividades que realiza el administrador del proyecto son:

- ◊ Establecimiento de una escala que refleje la probabilidad observada en un riesgo. La **escala puede definirse** en términos lógicos (si o no), cualitativos (muy probable, moderado, bastante improbable) o cuantitativos (probabilidades numéricas).
- ◊ Definición de las consecuencias del riesgo.
- ◊ Estimación del impacto del riesgo en el proyecto y en el sistema.

Los factores que afectan el impacto de un riesgo son: la naturaleza o **problemas potenciales** que se pueden producir; el alcance que provoca la severidad y la duración que determina el momento en que se sentirá el impacto y su duración.

Evaluación del riesgo

Durante esta actividad se evalúa la exactitud de las estimaciones que se han realizado durante la proyección del riesgo. Se intenta dar prioridad a los riesgos en base a su probabilidad de ocurrencia y se trata de planear la forma de controlarlos y/o prevenirlos.

En esta actividad se deben definir **niveles de referencia** o límites para el riesgo. Para la mayoría de los proyectos de *software* se utilizan niveles de referencia en cuanto a: el costo, el calendario y el rendimiento. Si una combinación de riesgos hace que se exceda uno o más de estos límites, se concluye que surgirá un problema que terminará por interrumpir el proyecto.

Los pasos que se siguen para la evaluación de los riesgos en un proyecto de *software* son:

1. Definir los niveles de referencia, o puntos de comparación del riesgo para la proyección.
2. Se acostumbra representar los riesgos en forma de terna (r, l, x) , donde r es el riesgo, l la probabilidad de que ocurra y x el impacto del riesgo. Se debe intentar desarrollar la relación entre cada terna (r, l, x) y cada uno de los niveles de referencia.
3. Determinar las condiciones que provocarían la interrupción de un proyecto.
4. Intentar predecir cómo afectarán al nivel de referencia las combinaciones de riesgos.

Gestión y supervisión del riesgo

Se parte de la revisión de las terna de riesgos y posteriormente se dan los pasos para tratar de impedir o suavizar los impactos del su posible ocurrencia. En el caso de proyectos grandes, donde como promedio aparecen de 30 a 40 riesgos, la serie de pasos para controlarlos representa un proyecto en sí mismo.

Se elabora un documento denominado Plan de Gestión y Supervisión de Riesgos, donde se concreta el análisis de riesgos y se incluye en el Plan global del proyecto.

Una vez desarrollado el Plan de Gestión y Supervisión de Riesgos, se comienza la supervisión del riesgo. La supervisión de riesgo es una actividad de seguimiento del proyecto; sus tres objetivos básicos son: 1) Detectar la ocurrencia de un riesgo que haya sido previsto; 2) Asegurar que los pasos de control del riesgo se estén aplicando correctamente, y 3) Recopilar información que se pueda utilizar para futuros análisis de riesgos.

Seguimiento y control

Una vez elaborada la calendarización del proyecto se comienza la actividad de seguimiento y control. En ella se sigue la pista a cada actividad componente del proyecto en cuestión. Si una actividad se sale del calendario, el administrador del proyecto puede determinar el impacto del error de planeación sobre los hitos² intermedios y sobre la fecha final de entrega. En ese caso, se reasignan recursos, se reordenan las actividades o, como última opción, se modifican los compromisos de entrega para resolver un problema no detectado oportunamente.

Esta actividad de seguimiento y control se efectúa permanentemente durante el desarrollo de un proyecto de *software* y a menudo se usan herramientas automatizadas para efectuarla con más precisión y facilidad.

2.3. ALCANCES

La planeación de todo proyecto nos permite establecer el camino a seguir para concluirlo oportuna y eficientemente, con la característica de que este camino es flexible, pues permite comparar las actividades realizadas con las planeadas y acelerar o reprogramar las actividades para que se concluyan a tiempo. Aunque esto pudiera parecer contradictorio, el proyecto debe permitir la flexibilidad y la unidad entre todas sus etapas; sin embargo, no es posible hacer a un lado los riesgos implícitos en el desarrollo de un proyecto.

² los hitos son puntos clave situados dentro del proceso de ingeniería de *software*, que proporcionan al líder del proyecto una indicación regular del proceso.

2.4. PLANEACIÓN DEL SISTEMA DE FACTURACIÓN CORPORATIVA (SIFACOR)

El objetivo de nuestro proyecto es desarrollar una aplicación que permita el registro y control de las transacciones de venta y otros movimientos contables de TELCEL, proporcionando documentos con buena presentación, además de valor fiscal si así lo requieren. Las principales consideraciones para su diseño son las siguientes:

- Facilitar el proceso de facturación a través de pantallas de captura sencillas y claras, así como el de consultas en línea; es decir, en el momento mismo de elaboración de un documento.
- Enlace de las operaciones del centro de atención con el área contable de la empresa.
- Impresión de documentos (Facturas, Notas de crédito, Notas de cargo y Contratos), con buena presentación.
- Impresión de reportes específicos.
- Opción de reimpresión de documentos.
- Enlace con el control de existencias (inventario y aviso de mínimos) del almacén.
- Prevenir la duplicidad de folios.
- Establecimiento de niveles de seguridad en la administración y manejo del sistema.

Existen además una serie de condicionantes que inciden en este diseño por formar parte de un proyecto mayor de la Dirección de Informática, se resumen en la siguiente lista.

Aprovechar la infraestructura existente.

- ☞ Herramienta de desarrollo Informix.
- ☞ Plataforma UNIX
- ☞ Equipos Bull DPX2
- ☞ Utilización de Impresoras que cuenten con programación de formatos.

Operación Multiusuario.

- ☞ Integridad del número consecutivo de folios.

Otro aspecto determinante de esta planeación es considerar que el desarrollo de este proyecto se basará en el modelo de ciclo de vida de desarrollo de sistemas, y como tal se divide en las siguientes etapas: **1) Análisis, 2) Diseño, 3) Desarrollo, 4) Implantación y Liberación, 5) Mantenimiento.** Cada una de las cuales comprende diferentes actividades para su realización:

- ☞ La etapa de **análisis** comprende: la recopilación y análisis de información, propuesta de solución y la factibilidad de dicha propuesta.

- ⇨ La etapa de **diseño** comprende: descripción del sistema, diseño estructural y detallado del sistema, consideraciones de comunicación de información, diseño de las entradas y salidas y la organización de los datos.
- ⇨ La etapa de **desarrollo** comprende la evaluación de la herramienta de programación y la codificación del sistema.
- ⇨ La etapa de **implementación y liberación** comprende: la instalación del sistema, las pruebas y evaluación, depuración y ajustes, capacitación del personal que lo utilizará y puesta en marcha.
- ⇨ La etapa de **mantenimiento** se realiza durante la vida útil del sistema.
Como la organización misma, el sistema debe ser capaz de adaptarse a los cambios requeridos en el interior de la empresa ó impuestos desde el exterior.

A continuación se presenta una relación más detallada de las actividades que contempla esta planeación.

Análisis

- Recopilación de información
 - Visitas a las áreas de operación
 - Entrevistas
- Análisis de información
 - Análisis del flujo de la información
- Propuesta de solución
 - Preparación de la propuesta
 - Presentación de la propuesta
- Factibilidad
 - Limitaciones del proyecto
 - Análisis Costo / Beneficio

Diseño

- Descripción del sistema
 - Revisión de la propuesta
- Diseño estructural y detallado del sistema
 - Narrativo de módulos del sistema
 - Narrativo de los subsistemas
- Consideraciones de comunicación de información
 - Consideraciones de flujo de información
 - Determinación de relaciones entre entidades
- Diseño de entradas y salidas

- Diseño de pantallas de captura
- Diseño de pantallas de consulta
- Diseño de reportes
- Organización de los datos
 - Diseño de la Base de Datos
 - Diseño del Diccionario de Datos

Desarrollo

- Elección de la herramienta de programación
 - Presentación de herramientas de programación
 - Selección de la herramienta de programación
- Programación de la solución
 - Escribir programas siguiendo diseño
 - Compilar los programas
 - Corrección de errores de programación
 - Documentar Bitácora de programas
 - Pruebas de programas

Implementación y liberación

- Instalación del sistema
 - Instalación y puesta a punto del equipo donde se implantará el sistema
- Las pruebas y evaluación
 - Presentación del sistema a los usuarios
 - Pruebas con datos ficticios
 - Pruebas con datos reales
 - Corridas paralelas
 - Documentación de observaciones y fallas
- Depuración y ajustes
 - Requerimientos adicionales
 - Cambios o modificaciones
- Capacitación
 - Entrenamiento a operadores y usuarios
 - Evaluación del aprovechamiento del entrenamiento
- Puesta en marcha
 - Fijar fecha de inicio de operación del sistema
 - Preparativos de recursos para inicialización
 - Determinación de puntos periódicos de revisión

Mantenimiento

- Realizar respaldos periódicos de los archivos que genera la facturación
- Llevar a cabo modificaciones y adaptaciones que se necesiten
- Etc.

Con la anterior descripción es posible elaborar la gráfica de Gantt y la estimación de recursos y costos de proyecto como se muestra en las FIGURA 2.4.1 y 2.4.2.

Como se puede observar en la FIGURA 2.4.1. la gráfica se divide en tres zonas principales: encabezados, cuerpo, y la zona el final de la gráfica donde se registran observaciones, conclusiones, etc. La zona media o cuerpo es la parte principal ya que es aquí donde se asocian las actividades con el tiempo que tomará realizarlas. Dentro de la primer columna se enlistan las actividades necesarias para el desarrollo de la aplicación. Se resalta con mayúsculas aquellas que marcan el comienzo de alguna de las etapas que conforman el ciclo de vida del desarrollo de sistemas. La segunda columna se marca con un asterisco (*), en ella aparece la inicial de él ó los responsables de llevar a cabo la actividad correspondiente, el significado de cada inicial se menciona en los recuadros del extremo derecho inferior y en el último renglón de la gráfica, donde se tiene que: **L**, corresponde a una actividad que deberá realizar el líder de proyecto; **A**, determina actividades que deberán llevarse a cabo por alguno o varios de los analistas programadores; y **G**, que muestra aquellas actividades donde el Gerente del área observará una participación directa.

Las columnas marcadas del 1 al 31 muestran el número de semana en la que deberán realizarse las actividades del proyecto; cada columna representa una semana, así, una actividad que abarque cuatro columnas indica que su período de realización se extiende por espacio de cuatro semanas. Como se puede observar las actividades de mayor duración se encuentran en las etapas de análisis, diseño y desarrollo.

La actividad de mantenimiento se marca distinto a las demás por tratarse de una etapa posterior a la liberación del SIFACOR, y aunque se tienen contempladas actividades específicas de mantenimiento y complementación del sistema, un buen número de las actividades que aquí habrán de realizarse se consideran imponderables que se manifiestan con el usuario hasta que el sistema comienza a formar parte de su rutina diaria, entre éstos podemos contar las correcciones inevitables y los ajustes a reportes, catálogos, reestructuración de claves, etc.

Finalmente, la columna de la extrema derecha se marca con un signo de porcentaje (%), ya que aquí podrá registrarse el grado de avance de alguna actividad hasta cierta fecha específica de corte.

En la tabla de la figura 2.4.2. se muestran los costos necesarios para proveer los recursos básicos del proyecto SIFACOR, en ella sólo aparecen los costos unitarios de los recursos directamente originados por él, sin embargo, no hay que olvidar que el proyecto SIFACOR es sólo una etapa de un proyecto mayor de la Dirección de Informática en el cuál se consideraron algunos de los recursos más costosos, como es el equipo en sí y el manejador de la Base de Datos.

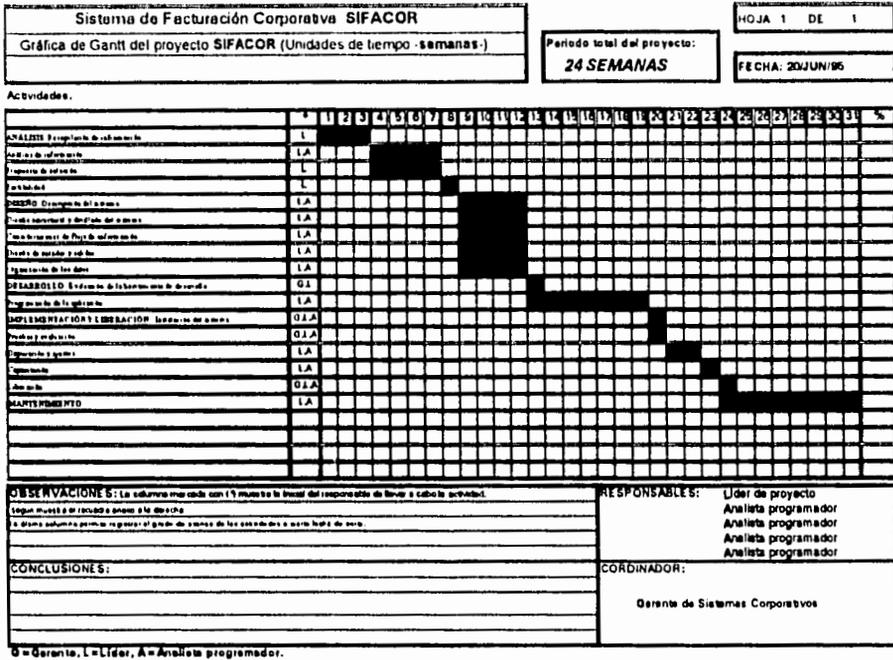


FIGURA 2.4.1.
GRÁFICA DE GANTT DEL PROYECTO SIFACOR

RECURSOS	DESCRIPCIÓN	COSTO
Humanos	1 Líder de proyecto	N\$ 5000.00 por mes
	4 Analistas Programadores	N\$ 3500.00 por mes
Materiales	Hardware	
	PC's c/tarjeta de red	N\$ 9200.00 c/u
	Impresoras	DLL 5200.00 c/u
	Software	
	Software para comunicaciones	DLL 83.00 c/u

FIGURA 2.4.2.
COSTOS DE LOS PRINCIPALES RECURSOS DEL PROYECTO

Otra consideración importante de un proyecto es tomar en cuenta que toda planeación conlleva riesgos, y todo riesgo, de ocurrir, produce un impacto. Los riesgos más fácilmente identificables en este proyecto se enumeran en una breve lista, así como sus posibles impactos, FIGURA 2.4.3.

RIESGO	IMPACTO
Rotación de personal	Retraso del proyecto, mayores costos
Modificación de requerimientos por parte del usuario	Retraso del proyecto, mayores costos
Reducción del presupuesto asignado al proyecto	Retraso ó cancelación parcial o total del proyecto

FIGURA 2.4.3.
DETERMINACIÓN DEL IMPACTO DE UN RIESGO SI ÉSTE SE CUMPLE

Como se puede observar los riesgos mencionados en la figura 2.4.3. son algunos de los riesgos inherentes que cualquier proyecto de ingeniería de *software*, grande o pequeño. El impacto casi siempre demora el proyecto y las demoras pueden significar importantes sumas de dinero, de ahí la importancia de al menos vislumbrar el panorama de las medidas a tomar en caso de presentarse alguna contingencia.

Las figuras anteriores brindan un panorama general del proyecto, darles seguimiento durante su periodo de desarrollo facilitará tomar las acciones correctivas necesarias en caso de desvíos o imprevistos.

CONCLUSIONES

La planeación es la etapa principal para el desarrollo y óptima realización de todas y cada una de las actividades que conforman un proyecto, tomando en cuenta varios aspectos como son: el tiempo, recursos humanos, equipo y costos.

Se ha visto que la planeación es un paso fundamental para la creación de un sistema de información. Aunque existen sistemas que se han logrado sin planeación, la experiencia indica que los que sí cumplen con los objetivos planteados en un principio son los que sí se planearon. La planeación coloca, por así decirlo, un marco para las acciones a seguir a todo lo largo del proyecto, permitiendo así el seguimiento y modificación pertinentes a cada actividad con el fin de alcanzar los objetivos fundamentales.

La planeación es absolutamente necesaria en todo proyecto de desarrollo de sistemas de información. Aunque toda planeación tiene riesgos inherentes, éstos pueden ser

detectados, evaluados y controlados para no volverse críticos e interrumpir parcial o totalmente el desarrollo del proyecto.

En este capítulo se estableció la importancia de la planeación en un proyecto de *software*, sin esta guía difícilmente podrían llevarse a cabo sistemas de información grandes y complejos donde son múltiples los elementos que hay que considerar.

La planeación como su nombre lo indica planea, determina o considera el mejor camino para desarrollar un proyecto, en este caso de *software*; sin embargo, no es por mucho un conjunto de reglas rígidas a manera de receta para lograr un objetivo, sino más bien, una guía para orientar al líder de un proyecto.

CAPÍTULO 3

ANÁLISIS

El presente capítulo se abocará al análisis de información. Se establecerán las necesidades y operaciones que la empresa quiere atender a través de un nuevo sistema de información. Se analizará el sistema de facturación actual REAL WORLD, mediante el establecimiento de un diagrama de contexto¹ que mostrará la relación del sistema con su entorno, y permitirá evaluar la posibilidad de modificarlo para satisfacer los requerimientos de los usuarios, que se complementarán mediante los resultados de la recopilación de información apoyada en entrevistas principalmente. Basados en el análisis de la información obtenida, se propondrán alternativas de solución; una vez seleccionada una alternativa se estudiarán con detalle los procesos y flujos de información mediante el análisis estructurado, estableciendo así las funciones que son susceptibles de automatizarse y que se incluirán en el nuevo sistema. Finalmente se hará una evaluación de la factibilidad del desarrollo de la aplicación en función de los recursos con que cuenta la organización.

¹El diagrama de contexto es una representación gráfica de las entidades que interactúan con el sistema y los flujos de información entre ellos.

3.1 RECOPIACIÓN DE INFORMACIÓN

TELCEL es una empresa mexicana que brinda servicios de telefonía celular en la mayor parte del territorio nacional. Estos servicios comprenden la venta de aparatos telefónicos y accesorios (baterías, antenas, cargadores), así como la prestación de servicios telefónicos específicos (buzones, conferencias tripartitas, rescate, auxilio vial y tasación entre otros).

En la figura 3.1. se muestra el organigrama de la empresa TELCEL, que está compuesto por las siguientes gerencias y direcciones:

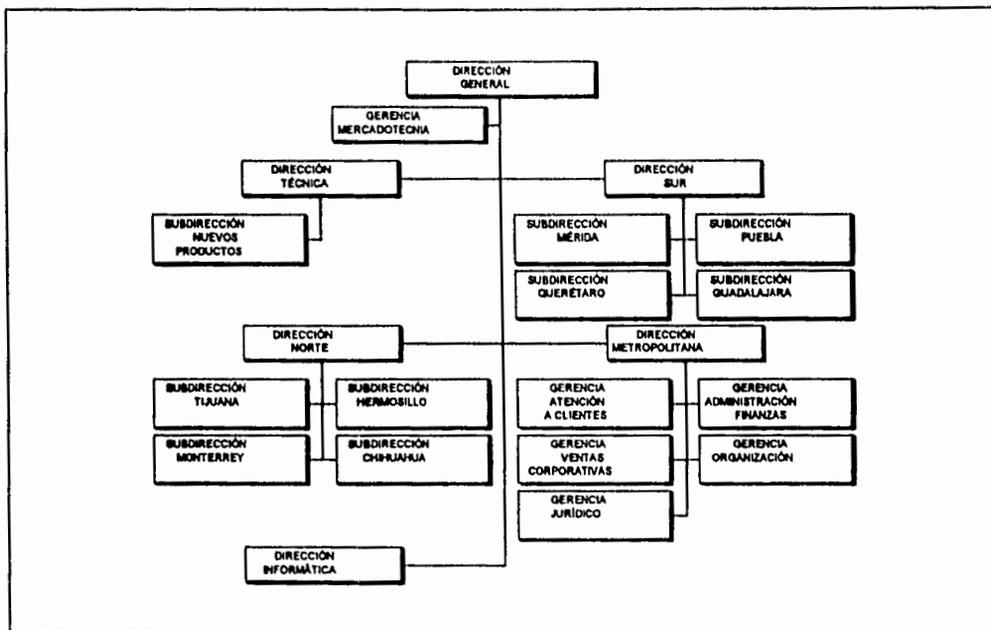


FIGURA 3.1.
ORGANIGRAMA DE TELCEL

Gerencia de Mercadotecnia. Se encarga principalmente de crear una imagen de la empresa destinada a los consumidores, a través de la publicidad de sus servicios.

Dirección Técnica. Se encarga de la operación y mantenimiento de las radiobases, antenas y equipo de comunicaciones que hacen posible el servicio de TELCEL.

Dirección de Informática. Su función es procesar todos los volúmenes de información generados por el servicio de telefonía y de la operación misma de la empresa.

Dirección Metropolitana. Tiene su símil en las direcciones Norte y Sur; la conforman las gerencias de Organización, Asuntos Jurídicos, Administración y Finanzas, Atención a Clientes y Ventas Corporativas².

Las tres últimas gerencias mencionadas son el objeto de atención, ya que son las que generan los requerimientos que originan el presente análisis. Por lo que se procederá a describir las actividades que se llevan a cabo en estas gerencias.

Gerencia de Atención a Clientes

Se encarga de poner en manos del cliente los productos y servicios que ofrece TELCEL, actualmente se cuenta con un solo centro de atención, aunque se prevé la necesidad de aumentar su número. En este centro se asesora al cliente en cuanto a los servicios que puede adquirir y la forma en que operan éstos, así como la gama de productos y accesorios que se ponen a su disposición. Una vez que el cliente pasa a ser un abonado de la empresa, Atención a Clientes se encarga de atenderle en lo sucesivo para cualquier necesidad que se le presente, por ejemplo: el cambio de aparatos, dudas o aclaraciones en el servicio, activación o inhibición de algún servicio en particular de manera temporal o indefinida, etc.

Por otro lado, una de las principales preocupaciones de esta gerencia es mejorar siempre la calidad de servicio que brindan sus asesores, por tal motivo se recaba información estadística de ventas que es el principal parámetro para establecer: los programas de capacitación de mejora en atención al cliente, de innovación de productos y servicios, etc., así como la rotación de los asesores y sus clientes.

Gerencia de Administración y Finanzas

Dentro de esta gerencia el área específica que ha generado requerimientos es la de Contabilidad, puesto que en ésta se lleva el control de los ingresos y egresos generados por el proceso de facturación de venta de los productos y servicios que brinda TELCEL.

Dentro de sus actividades primordiales se encuentra la acreditación de reembolsos y la determinación de cargos.

La acreditación de reembolsos ocurre cuando es necesario retribuir al cliente alguna cantidad por concepto de alguna devolución o la cancelación parcial o total de alguna factura, etc.

Por otro lado, la determinación de cargos ocurre cuando se detectan diferencias en alguna cuenta; es decir cuando se ha omitido, por algún motivo, el cobro de algún concepto al cliente.

Las actividades anteriormente descritas siempre tienen como resultado la afectación directa a una cuenta contable lo que hace posible su control interno.

Con el fin de apoyar a las operaciones de estas áreas de la empresa, hace aproximadamente tres años se implantó en las oficinas de Ventas y Atención a Clientes un sistema de facturación denominado REAL WORLD, que dio el soporte suficiente a las necesidades de ese momento; pero al paso del tiempo el número de clientes fue creciendo a un ritmo inesperado, lo que dio pie a mayores necesidades que el sistema fue incapaz de satisfacer. Entonces se detectaron graves fallas en la integridad de los datos y en el rendimiento del

² Esta gerencia es una división especial de la gerencia de atención a clientes, por lo que en lo sucesivo se hará mención sólo a Atención a Clientes.

sistema al manejar volúmenes cada vez mayores de información, pues a la fecha el número de abonados asciende a más de 200 mil. A raíz de esto, las entidades que hacen uso de la información proporcionada por el REAL WORLD, externaron una serie de necesidades que se listan a continuación:

ENTIDAD	NECESIDADES	OBSERVACIONES
Contabilidad General.	<ul style="list-style-type: none"> ⇨ Generar automáticamente el arrastre contable³ ⇨ Aceptar distintos tipos de pago ⇨ Tener un control estricto de los consecutivos fiscales ⇨ Tener un control de las áreas que elaboran Facturas, Notas de Crédito y Notas de Cargo. 	<ul style="list-style-type: none"> ⇨ Se tiene un catálogo de cuentas. ⇨ Actualmente se recibe efectivo, se deberá aceptar pagos con cheque, tarjeta, sin costo y crédito. ⇨ Poder consultar e imprimir las facturas y folios ⇨ Generar reportes.
Almacén	<ul style="list-style-type: none"> ⇨ Contar con un sistema de control de inventario. 	<ul style="list-style-type: none"> ⇨ Poder contar con múltiples bodegas. ⇨ Poder realizar valuación y clasificación de inventarios. ⇨ Poder considerar puntos de reorden. ⇨ Utilización del código de barras⁴.
Ventas Corporativas y Atención a Clientes	<ul style="list-style-type: none"> ⇨ Dar un mejor y más rápido servicio para el usuario. 	<ul style="list-style-type: none"> ⇨ Generación de documentos y reportes.

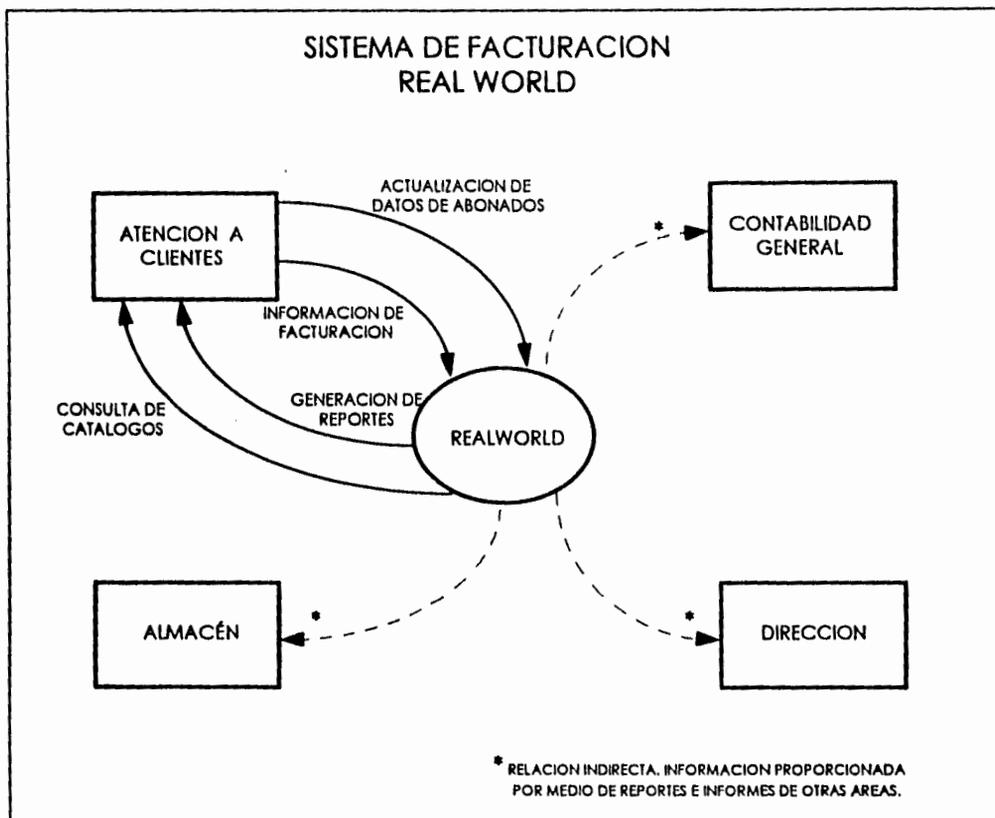
Estas necesidades obligan el llevar a cabo un análisis preliminar que permita determinar si el sistema actual se puede modificar, ó bien si será necesaria la creación de un nuevo sistema de información, para satisfacerlas eficientemente.

El programa actual de facturación se denomina REAL WORLD y es una aplicación comercial desarrollada en COBOL ESTRUCTURADO, un lenguaje convencional de alto nivel orientado hacia el procedimiento y con una amplia disponibilidad para manipulación de archivos.

En la figura 3.2. , se presenta el diagrama de contexto del REAL WORLD, donde se aprecia la interacción entre este sistema y las áreas involucradas en el proceso de facturación.

³Arrastre contable es la afectación directa a cuentas contables. Cuando se vende un producto o servicio el monto de éste afecta a varias cuentas contables.

⁴Para establecer enlace entre el inventario en bodega y el status en los productos en la base de datos introduciendo el número de serie en cada producto en una base de datos de inventario.



**FIGURA 3.2.
DIAGRAMA DE CONTEXTO DEL SISTEMA ACTUAL (REAL WORLD)**

Como se puede observar en el diagrama de contexto, la única entidad que tiene relación directa con el sistema es el área de Atención a Clientes. Esta entidad es la que proporciona información a las demás, es decir, el área de Atención a Clientes extrae la información del REAL WORLD, la formatea, generalmente a través de hojas de cálculo y otras herramientas, y así la da al resto de las áreas.

Actualmente no se cuenta con el código fuente del REAL WORLD, por lo que no es posible mejorar el sistema para satisfacer los requerimientos de los usuarios. Por esta razón se vislumbra desde el principio del análisis que no se podrá explotar esta información más allá de sus actuales alcances, esto nos lleva a considerar las limitantes del sistema actual.

LIMITANTES DEL SISTEMA ACTUAL

Como ya se mencionó, con el aumento del número de abonados, el REAL WORLD comenzó a mostrar graves fallas de integridad en los datos y en el rendimiento al manejar volúmenes de

información cada vez mayores. Por ejemplo, la elaboración e impresión de una sola nota de venta (típica) tardaba alrededor de 22 minutos. Para calcular la ineficiencia del sistema baste multiplicar este tiempo por la cantidad de clientes que acudían al centro de atención en ese mismo día. El número máximo de facturas que se podían elaborar diariamente era de 50. Esto provocaba pérdidas económicas y otras intangibles como la mala imagen ante el público.

Otras limitantes del REAL WORLD son:

1. En cuanto a la confiabilidad de la información, es de mencionar que el REAL WORLD permite la recaptura del número consecutivo de folio sin validar la existencia previa del documento. Por lo que es frecuente generar facturas con números de folio duplicados.
2. No existe liga alguna entre la facturación y la generación de notas de crédito o cargo; por lo que hay que hacer un rastreo prácticamente manual de estas notas.
3. Además es frecuente que los operarios inexpertos cometan errores sin que el sistema esté protegido contra ellos, lo que provoca graves fallas generales.

Pero dentro de las limitantes más importantes del REAL WORLD, se cuenta con un reclamo apremiante del área de contabilidad general por mantener la integridad del consecutivo fiscal (folios ininterumpidos) de facturas por venta de celulares, accesorios y servicios. Cabe decir que el REAL WORLD no está integrado a la contabilidad de la empresa, lo que significa que toda la afectación a cuentas contables por concepto de ventas de celulares, accesorios y servicios debe hacerse a mano, previa reimpresión de todas las facturas del mes para este proceso.

4. No existe ninguna opción que provea reportes de la venta del día o al menos del número de facturas generadas en una fecha en particular.
5. El programa no tiene la posibilidad de cancelar una factura equivocada.

Las observaciones mencionadas en las Necesidades y Limitantes del Sistema quedarían incompletas si no se hiciera una investigación más detallada, por lo que la recopilación adicional de información cumple un importante papel en este análisis.

RECOPIACIÓN DE INFORMACIÓN

En la fase de análisis del proceso de desarrollo del sistema, el analista de sistemas depende de técnicas específicas para recopilar información.

Los datos recolectados se analizan para determinar las especificaciones de los requerimientos, es decir, la descripción de las características del nuevo sistema que la empresa requiere.

Entre las técnicas de recopilación de información que se utilizaron están:

- ⇨ **Entrevista** con los usuarios involucrados con el sistema, es importante que el analista de sistemas se asegure de que cada persona que responda entienda que el objetivo final del trabajo es hacer que el nuevo sistema sea más útil. En este caso, se entrevistó a los responsables de las áreas involucradas por su familiaridad con los procesos efectuados en ellas.
- ⇨ La **observación directa** tanto del comportamiento del sistema como de sus usuarios, para así poder determinar lo que se está haciendo, la forma en que se hace, quién lo realiza, cuánto tiempo se requiere, dónde se hace y porqué.

Los resultados obtenidos durante la recopilación de información se mencionan a continuación, separados por áreas :

ÁREA	ENTREVISTADO
Almacén	Gerente del Área.

REQUERIMIENTOS

NUEVOS REPORTES Y MÓDULOS	ADECUACIONES Y MODIFICACIONES
<ul style="list-style-type: none"> ⇨ Reporte por rango de Facturas con totales por modelo de equipo y accesorios ⇨ Reporte de existencias (diaria por movimientos y concentrado por artículo) ⇨ Reporte de Costos por bodega ⇨ Módulo de distribución de bodega central a otras bodegas ⇨ Generación de Notas de salida de bodega ⇨ Catálogo de bodegas por región ⇨ Distribución de bodegas a tiendas ⇨ Implementación de código de barras para entrada, salidas y ventas de equipo ⇨ Históricos de costo 	

ÁREA	ENTREVISTADO
Atención a Clientes	Gerente del Área.

REQUERIMIENTOS

NUEVOS REPORTES Y MÓDULOS	ADECUACIONES Y MODIFICACIONES
<ul style="list-style-type: none"> ⇨ Reportes de ventas por vendedor y puntos de venta con detalle de venta ⇨ Reporte acumulado de productos vendidos con totales por tipo de producto (Teléfono, accesorios , servicios) ⇨ Módulo para reubicación de vendedores a otras tiendas 	<ul style="list-style-type: none"> ⇨ No permitir la modificación de precios unitarios en la facturación para los conceptos de teléfonos y accesorios ⇨ Autorización de descuentos por supervisor o jefe asesor ⇨ Bitácora de autorizaciones de descuentos y reportes ⇨ Agregar a las consultas de facturas y productos cortes por tienda

ÁREA	ENTREVISTADO
Contabilidad General	Gerente del Área

REQUERIMIENTOS

NUEVOS REPORTE Y MÓDULOS	ADECUACIONES Y MODIFICACIONES
<ul style="list-style-type: none"> ⇨ Reporte de Facturas (Cortes Diarios, Mensuales, por forma de Pago) ⇨ Generación de recibos de Pago ⇨ Agregar reimpresión de Notas de Crédito, Notas de Cargo y Facturas Roaming ⇨ Reporte de ventas por vendedor y punto de venta con detalle de ventas ⇨ Reporte Cuatrimestral de S.H.C.P. ⇨ Reporte de 50 Mejores Clientes. ⇨ Catálogo de Comisiones y Definición de Cuota Mensual. ⇨ Reporte del Cálculo de Comisiones por Vendedor. ⇨ Autorización de Comisiones por Vendedor. ⇨ Consultas y Reportes de Factura. ⇨ Consultas y Reportes de Ventas por tipo de Producto. ⇨ Consulta de Reporte de Estados de Factura (Como se afectan por otros documentos) ⇨ Reporte de Factura con detalle. ⇨ Reporte de Facturas por Cobrar (Facturas de Crédito). ⇨ Reporte de Facturas por punto de Venta agrupadas por su forma de pago 	<ul style="list-style-type: none"> ⇨ Reimpresión de Documentos (Agregar clave de seguridad) ⇨ Impresión de código de barras en facturas y contratos ⇨ Detallar procedimientos de cuenta en facturación (manejo de cargos, abonos y saldos en cuentas contables). ⇨ Generar Notas de Crédito o cargos parciales por precio ⇨ Agregar Fecha de Pago a las Facturas. ⇨ Implementación Espejo de Notas de Cargo y Notas de Crédito. Esto significa que cuando se genera uno de estos documentos automáticamente se reflejarán los conceptos del documento que van a afectar. ⇨ Si existe la posibilidad de que un cliente llegue con su propio teléfono a pedir una asignación de línea, se deberá generar un módulo de contratación (asistido de la facturación). ⇨ Si no se da el caso anterior se deberá validar en la facturación que: ⇨ El Precio Unitario se debe de poder modificar para Notas de Crédito y Notas de Cargo sólo cuando el concepto sea por un servicio . ⇨ Generar módulos de Facturación, Notas de Crédito y Cargo para ISIS⁵

⁵ ISIS es el sistema que efectúa la tasación de tiempo- aire.

La directiva de la empresa consideró apropiado ser autoimpresor de comprobantes fiscales por las ventajas que se obtienen: el hecho de no realizar el pago a terceros por la elaboración de éstos y evitar la duplicidad de sus comprobantes para el mal uso de los mismos. Así que se deben tomar en cuenta como requerimientos del nuevo sistema, los requisitos fiscales para la autoimpresión que marca el artículo 29-A del Código Fiscal de la Federación y la Resolución Miscelánea publicada el 31 de marzo de 1994⁶. Si la S.H.C.P. detecta que la empresa no cumple con alguno de los requisitos mencionados, podrá revocar la autorización otorgada sobre la autoimpresión de estos comprobantes.

Adicional a lo anterior, es necesario considerar que el sistema que se piensa desarrollar - que se denominará SIFACOR - formará parte de un proyecto integral de la dirección de la empresa, el cual consiste en la integración y estandarización a nivel nacional de los procesos corporativos con el fin de automatizar en forma completa los departamentos de la empresa utilizando la plataforma tecnológica existente.

Analizando la información anteriormente descrita se procederá a concretar las necesidades que deberá cubrir el sistema que sustituirá al REAL WORLD.

3.2 NECESIDADES DEL SISTEMA

Dentro de las necesidades del nuevo sistema tenemos:

- * Integración de la facturación de los Centros de Atención con la contabilidad de la empresa
- * Eliminación de la duplicidad de folios trabajando bajo un ambiente multiusuario
- * Mantener la integridad de los datos aún cuando el sistema sea operado por usuarios inexpertos
- * Generar la documentación que se entrega al cliente en el momento de contratación de una línea celular o al adquirir algún otro bien o servicio, en una forma rápida, confiable y de buena calidad.
- * Contemplar el uso de formatos precargados en la impresora.
- * Generación de diversos reportes para las áreas de TELCEL
- * Generación de reportes para el área de ventas: manejo de comisiones por vendedor, por volumen de ventas.
- * Autoimpresión de los comprobantes fiscales.
- * Se requiere control de las existencias de aparatos y accesorios en almacén, por modelo y por número de serie, avisando cuando se llega al punto de reorden⁷.
- * Generación de notas de crédito, notas de cargo ligadas a la facturación.
- * Llevar a cabo el arrastre contable.
- * Información real sobre la venta y renta de accesorios.
- * Establecimiento de nuevos centros de atención a clientes en diversos puntos de la República, donde se proporcionen los mismos servicios que en el centro del D.F.
- * Permitir la cancelación de facturas.
- * Incluir consultas en línea rápidas.
- * Autorización de descuentos.
- * Incluir niveles de seguridad en el sistema.

⁶El detalle de estos requisitos fiscales se presenta en un apéndice.

⁷El punto de reorden es el límite de existencias mínimas en el almacén. Cuando se alcanza se debe proveer el faltante.

Con lo anterior en mente, se procederá a presentar dos propuestas de solución.

3.3 ALTERNATIVAS DE SOLUCIÓN

Como resultado del análisis anterior, se ha determinado que el software REAL WORLD que actualmente está operando, no satisface las necesidades de los involucrados en su uso y por sus características (no existe código fuente), no es posible adecuarlo a los requerimientos ya señalados.

Así que, basados en los requerimientos recabados en las áreas de incumbencia, se establece la necesidad de poseer un sistema que permita interactuar eficientemente al personal de las áreas señaladas y además que cumpla con requisitos de desempeño y calidad⁸.

A continuación se detallan las alternativas de solución que se proponen:

1. Se plantea el desarrollo de un sistema integral de facturación cuya aplicación satisfaga la mayor parte de las necesidades registradas. Se estima que un sistema con estas características tardaría aproximadamente 16 meses desde el inicio de su diseño hasta su completa liberación.
2. Desarrollo de un sistema que satisfaga de momento las necesidades más apremiantes y liberación paulatina del resto de los módulos, dependiendo de su prioridad (Facturación, Catálogos, Consultas, Reportes y Administración del Sistema).

Se propone a la alternativa dos como la más viable, puesto que se aboca a la necesidad primordial de implantar un sistema que emita documentos de valor fiscal con calidad y oportunidad. Además, este enfoque permitirá ser más concisos a la hora de desarrollar los módulos.

Se estima que el sistema de la segunda opción se liberará en alrededor de seis meses con lo que se puede cubrir la necesidad inmediata.

3.4 ANÁLISIS ESTRUCTURADO

Una vez realizado un análisis preliminar que permite determinar si se deberá modificar el sistema actual ó bien, como es en este caso, el desarrollo de un nuevo sistema; se debe hacer un análisis más profundo de las actividades que llevan a cabo las entidades para establecer en detalle las especificaciones del nuevo sistema. Estas especificaciones deben dejar bien claro lo que debe hacer el nuevo sistema. Para realizar esto se hará uso del análisis estructurado que facilita la comprensión de los procesos por medio de diagramas de nivel que ilustran el flujo de información en los diferentes procesos.

⁸Entre los factores que determinan la calidad de un sistema de información están: la fiabilidad, eficiencia, integridad, facilidad de uso, facilidad de mantenimiento, flexibilidad, facilidad de prueba, portabilidad, reusabilidad y la facilidad de interoperación (McCall "Factors in Software Quality")

Primeramente se identificarán las actividades primordiales de cada entidad, para posteriormente descomponerlas en procesos donde se aprecien: los flujos de datos, los almacenes de datos, así como los orígenes y destinos de la información.

Actividades de la entidad Atención a Clientes:

Evento	Procesos
Ventas	Asesora a Cliente Consulta planes Efectúa contrato Elabora factura Recibe pago Entrega documentos bienes y servicios
Aclaraciones	Revisión de contrato y factura Autoriza cambio ó devolución Canaliza Cliente a Contabilidad
Evaluación de Asesores	Consulta informe de ventas por asesor Reubica a vendedores (asesores)
Obtención de estadísticas de venta	Consulta información de ventas por asesor Consulta información de ventas por tipo de producto

Actividades de la entidad Contabilidad:

Evento	Procesos
Acreditación de reembolsos	Elabora nota de crédito Afecta a cuentas contables
Determinación de Cargos	Elabora nota de cargo Afecta a cuentas contables

Actividades de la entidad Almacén:

Evento	Procesos
Clasifica y ordena productos	
Entera a Ventas de las existencias (reporte)	
Valida constantemente máximos y mínimos de existencias	Comprueba existencias Contacta a proveedor para reabastecimiento

En la figura 3.3. se pueden apreciar los cuatro procesos principales de la entidad Atención a Clientes: Ventas, Evaluación de Asesores, Aclaraciones y Obtención Estadística de Ventas, así como las entidades externas que tienen contacto con esta área. Así mismo se detallan las entradas y salidas de información, entre ellas reportes, documentos y otros datos pertinentes.

Se observa en el diagrama el camino que sigue el producto desde el almacén hasta su entrega al cliente ya que de esa manera se cierra el ciclo de ventas. Igualmente, existe un almacén de datos "archivos de ventas" que en realidad no es un archivo por sí solo sino un conjunto de ellos. Estos archivos contienen datos de: clientes, facturas, contratos, etc. En los diagramas derivados de éste se detallan estos aspectos.

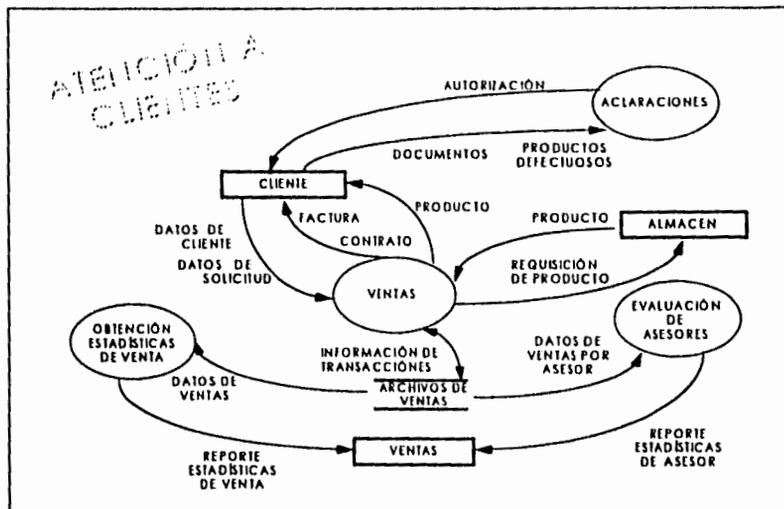


FIGURA 3.3.
DIAGRAMA DE NIVEL 0: DE ATENCIÓN A CLIENTES

En la figura 3.4. se muestran los procesos que se llevan a cabo, las entradas y salidas, así como las entidades que intervienen en la actividad ventas de la entidad Atención a Clientes. Esta actividad se descompone a su vez en seis procesos: Elabora Contrato, Alta de Cliente, Elabora Factura, Consulta Planes, Asesora Cliente y Recibe y Entrega Material.

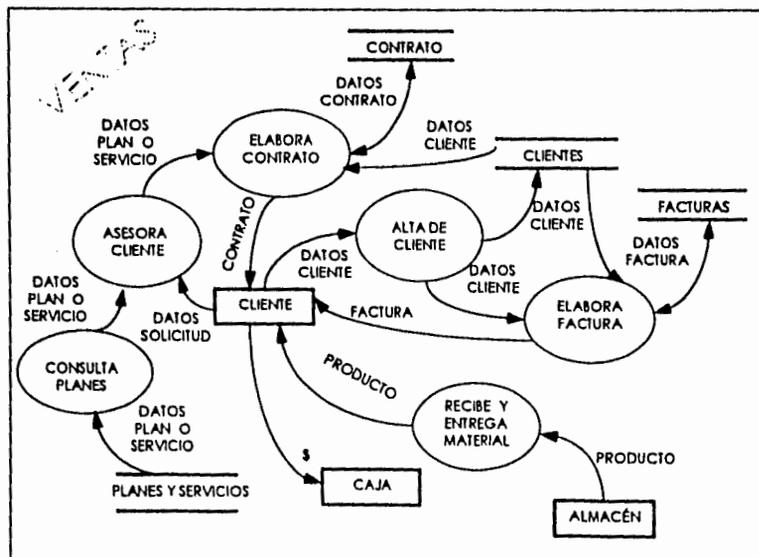


FIGURA 3.4.
DIAGRAMA NIVEL 1: VENTAS

Se puede apreciar que se especifican los almacenes de datos con más detalle que en el diagrama de nivel 0: Contratos, Clientes, Facturas y Planes y Servicios. Es de notarse que en los almacenes de datos de Clientes, Contratos y Facturas existe un flujo bidireccional de datos, mientras que en Planes y Servicios sólo se hacen consultas. Se incluye flujo de información pero también de productos y dinero, con el objeto de mostrar con claridad como se cierran los ciclos que se llevan a cabo dentro de este proceso.

Posteriormente se detallarán las burbujas de Elaboración de Contrato y de Factura, el resto de ellas no se desglosa puesto que son procesos simples que no requieren de descomposición alguna. Por ejemplo, el proceso Consulta Planes consiste solo en revisar los planes y servicios disponibles para proporcionar asesoría al cliente. El proceso Recibe y Entrega Material como su nombre lo indica, simplemente efectúa estas dos cosas.

En la figura 3.5. podemos apreciar que existen tres procesos para la elaboración de un contrato: Alta del Cliente, Llenado de Formas e Impresión de Contratos.

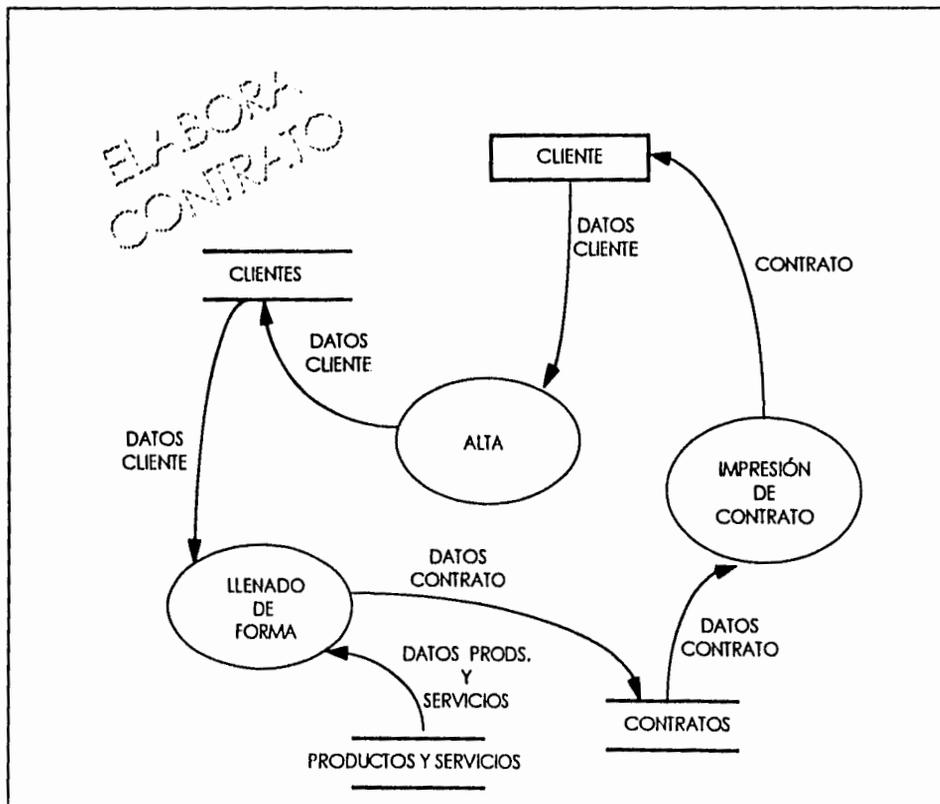


FIGURA 3.5.
DIAGRAMA DE NIVEL 2. ELABORA CONTRATO

Para el llenado de formas se hace uso de los siguientes almacenes de datos: Clientes, Productos y Servicios y Contratos. En los primeros dos se escribe información y en el último solamente se hacen consultas. El proceso Alta existe para registrar nuevos clientes en el almacén de datos Clientes, este proceso se lleva a cabo siempre y cuando el cliente no exista. Estas burbujas ya no son susceptibles de descomponerse puesto que se ha alcanzado el nivel más simple posible.

En la figura 3.6. se observa que para elaborar una factura se llevan a cabo tres procesos: Llenado de Formas, Impresión de Factura y Registro de Productos Vendidos.

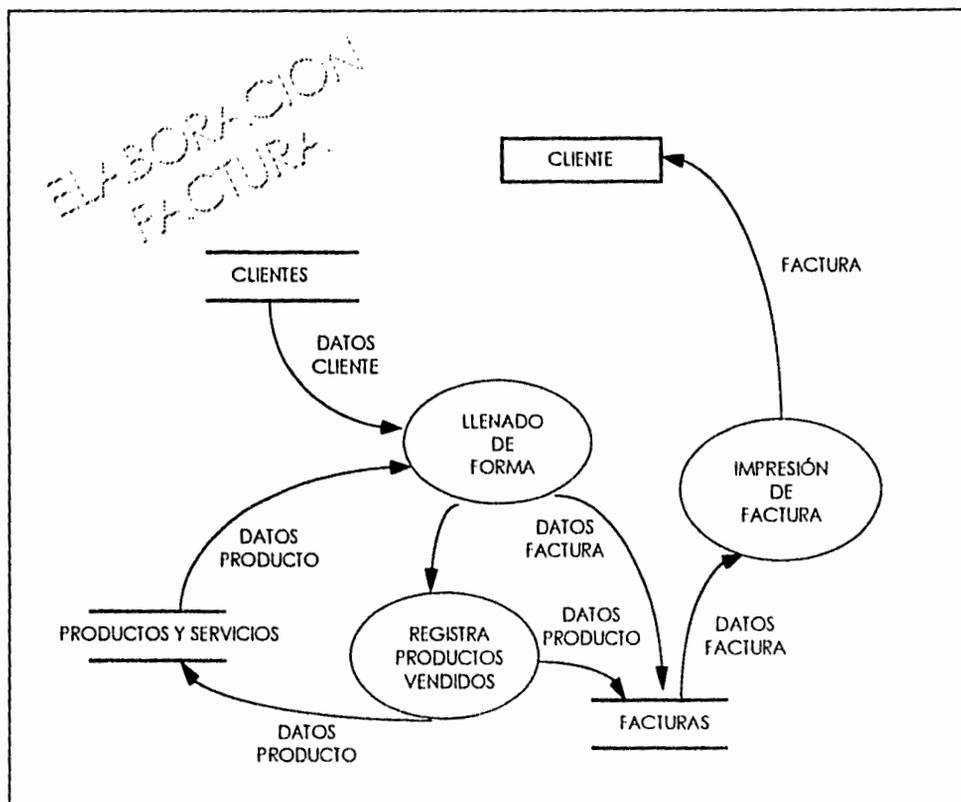


FIGURA 3.6.
DIAGRAMA DE NIVEL 2. ELABORACIÓN DE FACTURA

Como salida de este proceso está la factura que se entrega al cliente. Se hace uso de los almacenes de datos Facturas, Clientes y Productos y Servicios. En el proceso Registra Productos Vendidos se anotan los artículos o servicios que se han vendido en el almacén de datos Productos y Servicios. También a través de este proceso se registran los bienes y servicios vendidos en el almacén de datos Factura para su posterior utilización por el proceso de impresión.

En la figura 3.7. se puede apreciar que el evento Aclaraciones consiste de un proceso denominado Autoriza Aclaración, cuyas entidades externas son: Cliente y Almacén. En este proceso no se efectúa ninguna escritura en los almacenes de datos, sino que sólo se realizan consultas. Dentro del proceso Autoriza Aclaración ocurren una serie de actividades que se detallan en el siguiente diagrama de nivel 2.

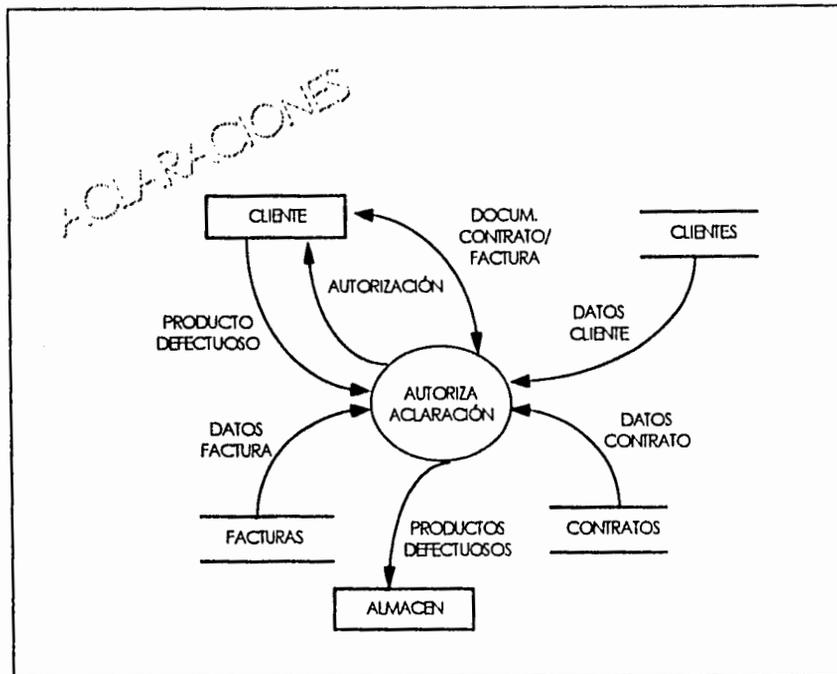


FIGURA 3.7.
DIAGRAMA NIVEL 1: ACLARACIONES

En la figura 3.8. se muestran lo que sucede cuando un cliente se presenta con una inconformidad a las oficinas de Atención a Clientes. Se lleva a cabo el evento Aclaraciones donde se realizan los siguientes procesos: Verificación de Documentación presentada por el cliente, es decir, se revisan los datos contenidos en la factura y contrato que el cliente presenta. En el caso de que un cliente presente un producto defectuoso, se verifica si en realidad el producto está defectuoso, el mismo es enviado a almacén y por último se lleva a cabo un proceso de Autorización, que consiste en dar al cliente una contraseña que contiene datos pertinentes y que deberá presentar al área de Contabilidad (Véase figura 3.12., Diagrama de flujo de datos de Acreditación de Reembolsos).

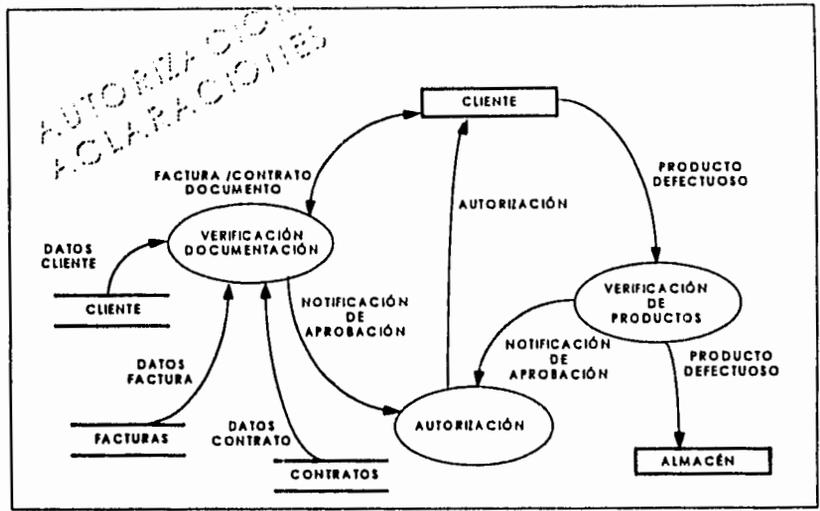


FIGURA 3.8.
DIAGRAMA DE NIVEL 2: AUTORIZACIÓN ACLARACIONES

En la figura 3.9. se puede apreciar que durante el evento Evaluación de Asesores ocurren dos procesos: Consulta de Ventas por Asesor y Reubicación de Asesores.

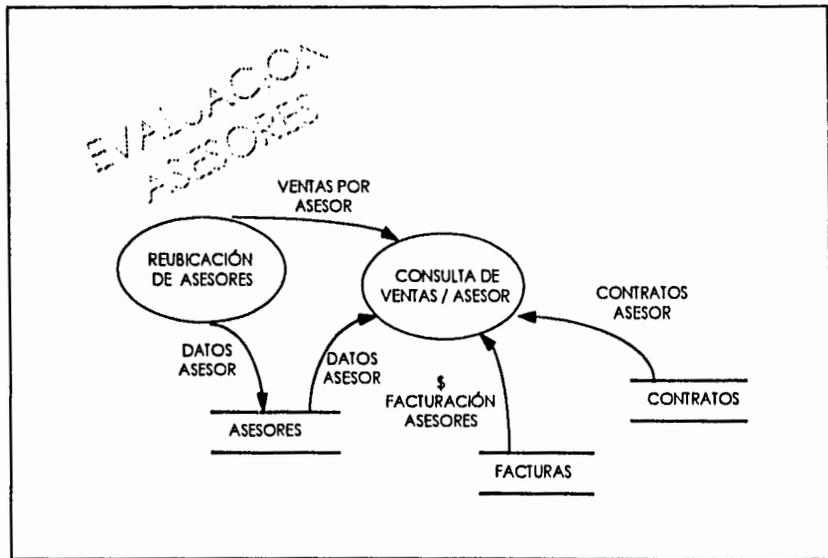


FIGURA 3.9.
DIAGRAMA DE NIVEL 1: EVALUACIÓN DE ASESORES

Durante la Consulta de Ventas por Asesor se consulta a tres almacenes de datos: Asesores, Facturas y Contratos, en estas consultas es posible determinar el desempeño de un asesor para poder así evaluar si está preparado, tiene capacidad para atender a cuentas más importantes, o si necesita períodos adicionales de preparación. Por otro lado, el proceso de Reubicación de Asesores ocurre como consecuencia del primero, ya que dada la capacidad de cada asesor así como sus circunstancias, es posible que se reasignen las cuentas que deba manejar.

En la figura 3.10. se presenta el diagrama correspondiente al proceso Obtención de Estadísticas.

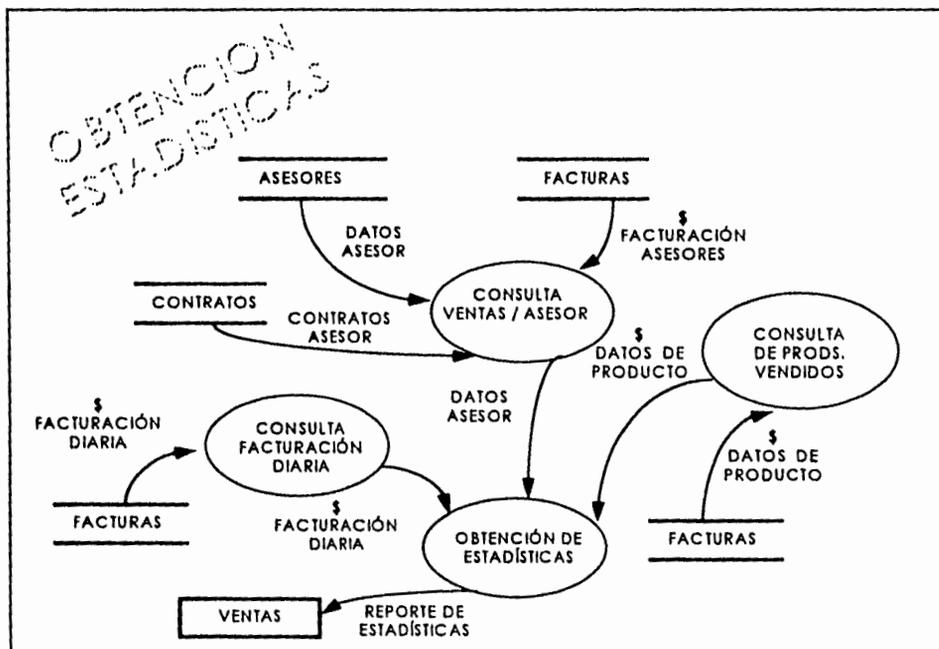


FIGURA 3.10.
DIAGRAMA DE NIVEL 1: OBTENCIÓN DE ESTADÍSTICAS

El objetivo principal de este proceso es la obtención de información estadística acerca de cuestiones como la facturación diaria, los montos de venta de cada asesor, así como de los productos vendidos, que permita la toma de decisiones. Se observa que se hace consulta a los almacenes de datos de: Ventas, Facturas, Contratos y Asesores. Se puede apreciar también que los tres procesos de consulta hacen uso del almacén de datos Facturas, sólo que toman distinta información de él: facturación diaria, datos de producto y facturación de asesor. La salida generada por el proceso Obtención de Estadísticas es un reporte que muestra cómo van las ventas de la empresa en sus diferentes aspectos, midiendo así la productividad de la misma y la eficiencia de su personal de ventas.

Con este último diagrama se termina el análisis de los eventos y flujos de información que se llevan a cabo en las oficinas de Atención a Clientes.

A continuación se analizarán los eventos correspondientes al área de Contabilidad.

En la figura 3.11. se muestra el Diagrama de Nivel 0 de la entidad Contabilidad, cuyos procesos son: Acreditación de Reembolsos y Determinación de Cargos.

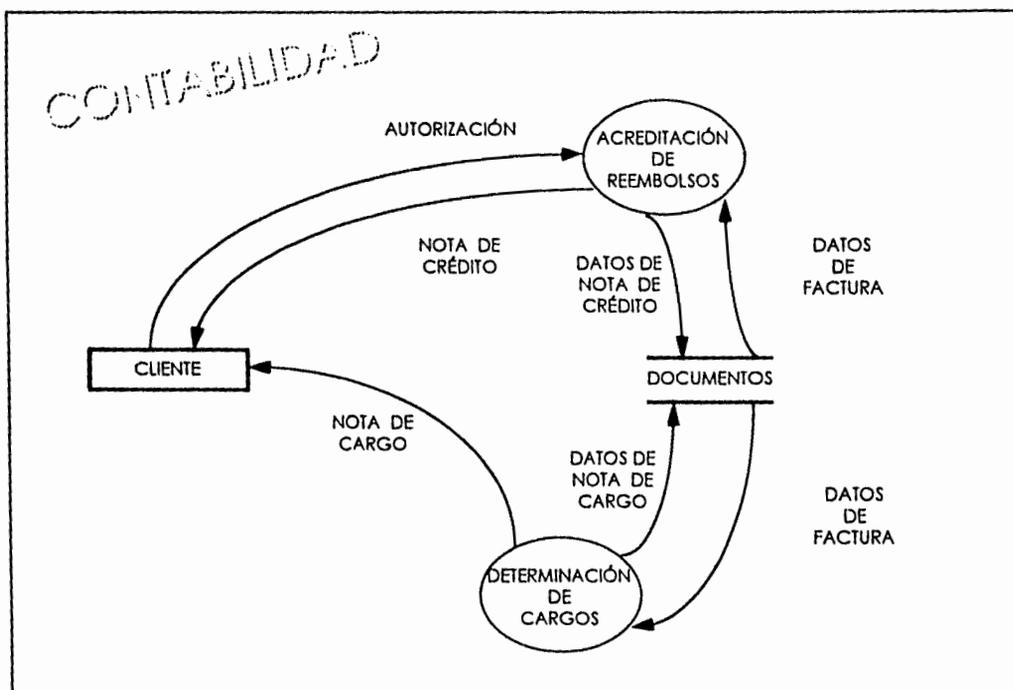


FIGURA 3.11.
DIAGRAMA DE NIVEL 0: CONTABILIDAD

Estos dos procesos hacen uso del almacén denominado Documentos (Facturas, Contratos, Notas de Cargo y Notas de Crédito) para obtener información de Facturas y determinar los montos y conceptos de las Notas de Crédito y Notas de Cargo, según sea el caso; siendo necesario almacenar los datos de estos papeles. Podemos observar que la salida de ambos procesos son documentos: Nota de Crédito en el caso de Acreditación de Reembolsos y Nota de Cargo en el caso de Determinación de Cargos, documentos que son entregados directamente al cliente para su cobro.

En la figura 3.12. se observa el diagrama de Acreditación de Reembolsos.

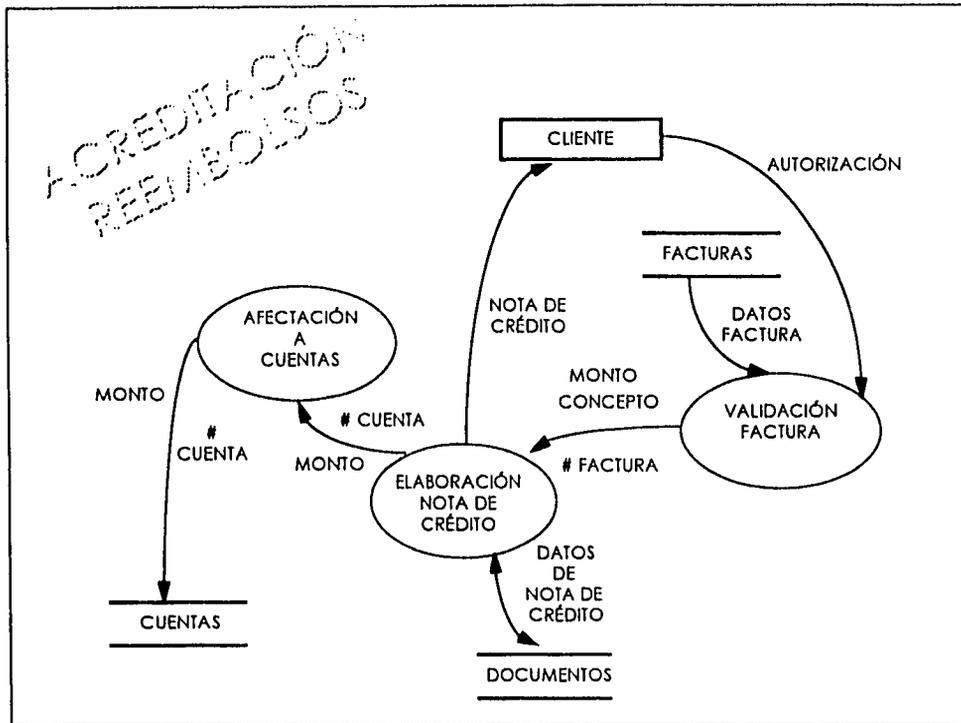


FIGURA 3.12.
DIAGRAMA DE NIVEL 1: ACREDITACIÓN DE REEMBOLSOS

En este diagrama se puede apreciar la existencia de tres procesos relacionados con Acreditación de Reembolsos: Validación de Factura, Elaboración de Nota de Crédito y Afectación de Cuentas. El proceso Elaboración de Nota de Crédito consiste de la captura e impresión de este tipo de documento. Se registra en el almacén de datos Documentos la información de la Nota de Crédito, así mismo se lee esta información para la posterior impresión del documento que es entregado al cliente. Cabe hacer notar que el proceso Afectación a Cuentas es de suma importancia ya que es ahí donde se lleva a cabo la afectación directa a las cuentas contables.

En la figura 3.13. se presenta el proceso correspondiente a Determinación de Cargos.

En este diagrama se puede apreciar la existencia de tres procesos relacionados con Determinación de Cargos: Validación de Factura, Elaboración de Nota de Cargo y Afectación de Cuentas. El proceso Elaboración de Nota de Cargo consiste de la captura e impresión de este tipo de documento. Se escribe en el almacén de datos Documentos la información de la Nota de Cargo, así mismo se lee esta información para la posterior impresión del documento que es entregado al cliente. Cabe hacer notar que el proceso Afectación a Cuentas es de

suma importancia ya que es ahí donde se lleva acabo la afectación directa a las cuentas contables.

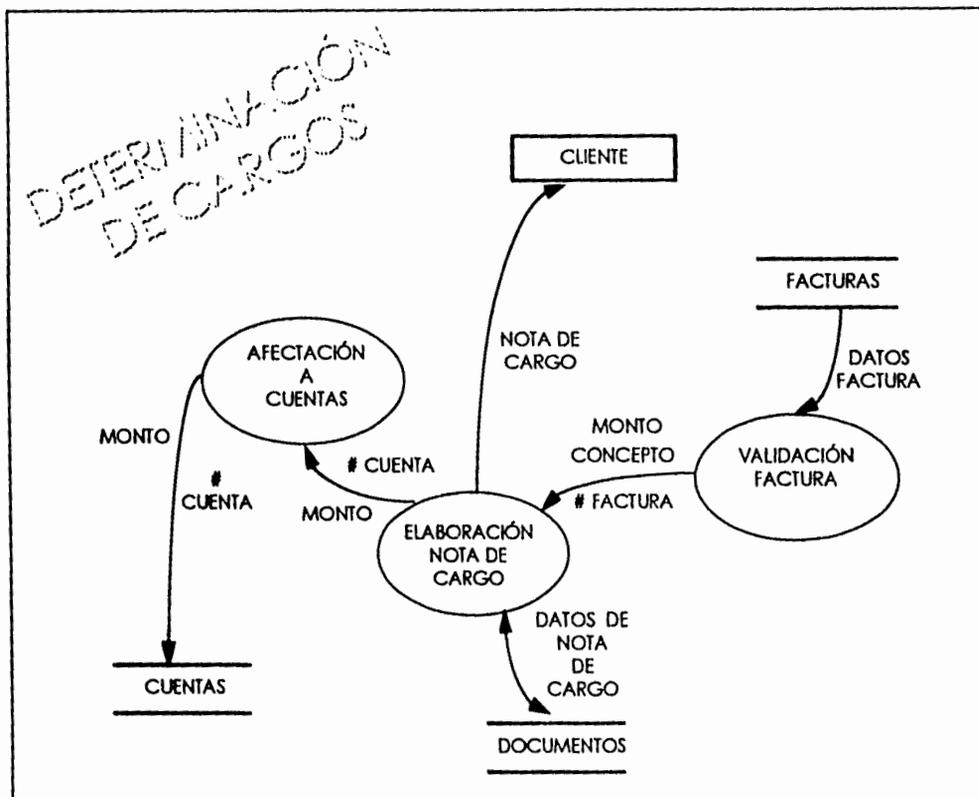


FIGURA 3.13.
DIAGRAMA DE NIVEL 1: DETERMINACIÓN DE CARGOS

Aquí concluye el análisis de los procesos que se efectúan en el área de Contabilidad. A continuación se esbozan las actividades relacionadas con la entidad Almacén.

En la figura 3.14. se presentan los procesos que se llevan a cabo en el Almacén.

En el diagrama de nivel 0: Almacén podemos apreciar que ocurren tres procesos involucrados con la venta de productos, estos son: Recepción de Productos, Clasificación y Ordenamiento, y Elaboración de Reporte de Existencias. Se observa que en el almacén inventario se guarda la información de productos que posteriormente es utilizada para generar un reporte que es enviado al área de Ventas, para enterarlo de la existencia de productos que puede vender; y que le permite al área de almacén solicitar a los Proveedores el reabastecimiento necesario de productos.

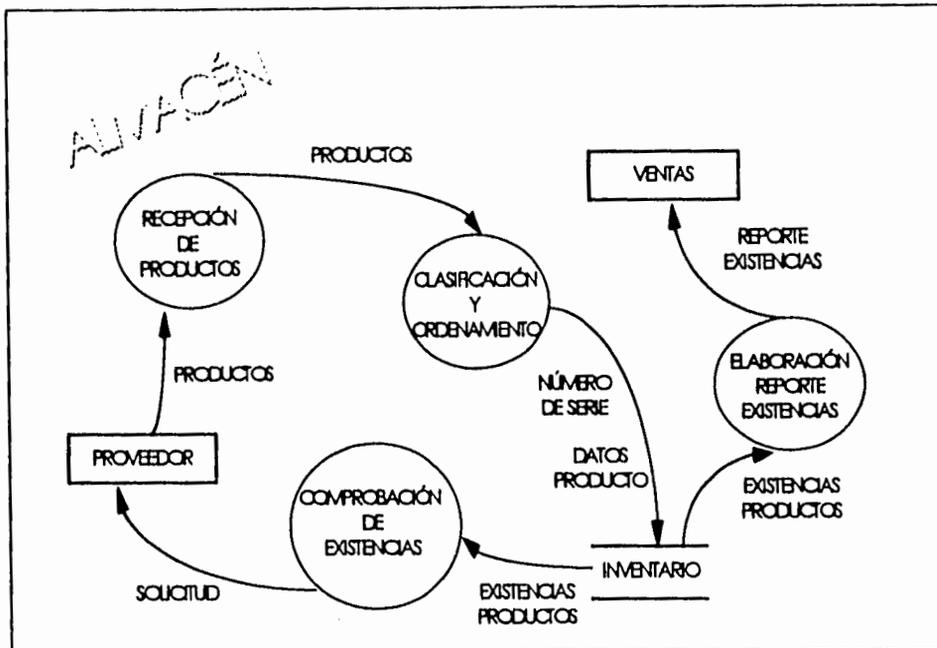


FIGURA 3.14.
DIAGRAMA DE NIVEL 0: ALMACÉN

En el diagrama 3.15. se detallan los procesos que se llevan a cabo en el almacén.

En el diagrama de nivel 1: Actividades del Almacén, se muestran cinco procesos: Recepción de Productos, Requisición a Proveedores, Comprobación de Existencias, Clasificación y Ordenamiento de Productos, y Elaboración de Reporte de Existencias. En estos procesos se observan flujos de datos hacia los almacenes de información: Inventario y Catálogo, y a las entidades: Proveedor, Ventas y al mismo Almacén. Como se observa, la entidad Almacén realiza un ciclo recibiendo los productos, ordenándolos y clasificándolos; actualizando los almacenes Catálogo e Inventario con la información de dichos productos. Posteriormente consulta estos almacenes para reportar las existencias a Ventas y poder solicitar, por medio de requisiciones, el reabastecimiento de productos a los Proveedores. Una vez que almacén recibe los productos solicitados, inicia nuevamente el ciclo.

Los diagramas de nivel hasta aquí descritos muestran como se llevan a cabo los procesos de las entidades que conforman el medio ambiente del sistema.

Como se ha podido observar, la naturaleza de los posibles usuarios del sistema es diversa, ya que se cuenta con asesores de venta por un lado, personal de contabilidad por otro y de almacén por otro.

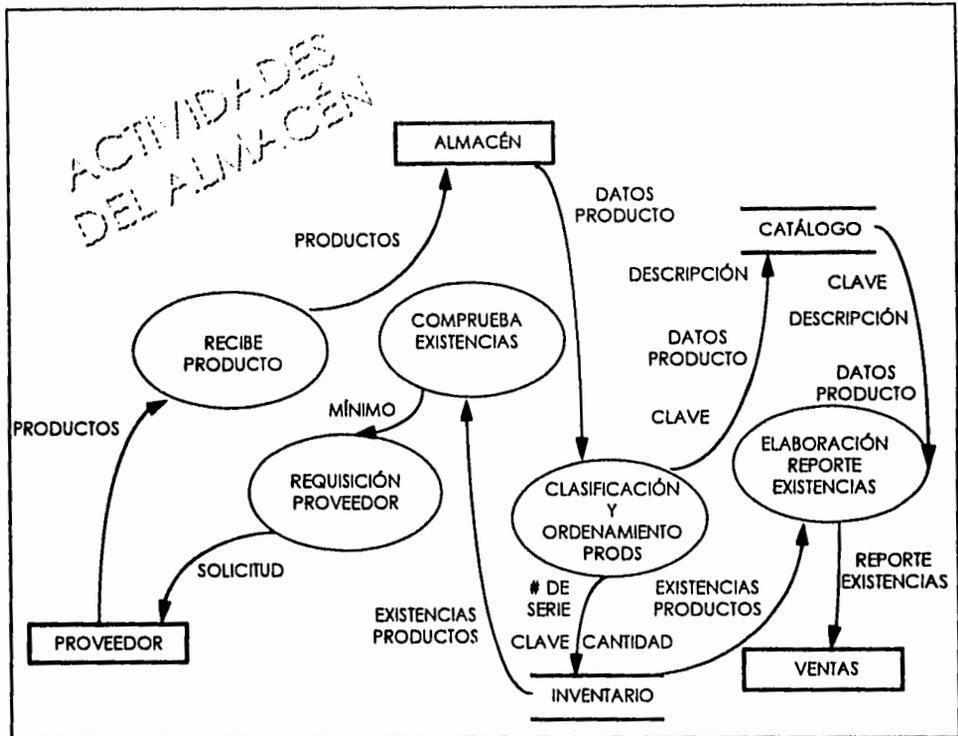


FIGURA 3.15.
DIAGRAMA DE NIVEL 1: ACTIVIDADES DEL ALMACÉN.

El hecho de que todos los usuarios pudieran tener acceso a la información que manejan unos y otros, pudiera poner en graves riesgos la integridad del sistema; sobre todo en el caso de usuarios inexpertos o, tal vez, mal intencionados. Pensando en cómo sería posible subsanar este riesgo latente, se decidió restringir el acceso de los usuarios a los programas, para que cada usuario, según su área y actividades, sólo pudiera utilizar aquellos que le correspondan.

Es por este motivo que se vislumbra la necesidad de un administrador del sistema, ajeno a las áreas usuarias, que lleve a cabo en forma eficiente el control de acceso de los usuarios. Se pensó entonces en la creación de un catálogo de usuarios donde se registre su información básica; como el nombre, área en la que trabaja --y por tanto los programas que debe usar--, así como la impresora de red más cercana a él. Un buen prospecto para llevar a cabo esta actividad es el área de **Sistemas** de la empresa, la cual está efectuando el presente análisis.

El hecho de que el área de **Sistemas** se haga responsable de la administración del sistema de información hace indispensable que se incluya esta nueva entidad al ámbito del mismo.

A continuación, en el diagrama 3.16., se detallan las actividades que esta entidad deberá llevar a cabo para ejercer la administración del sistema.

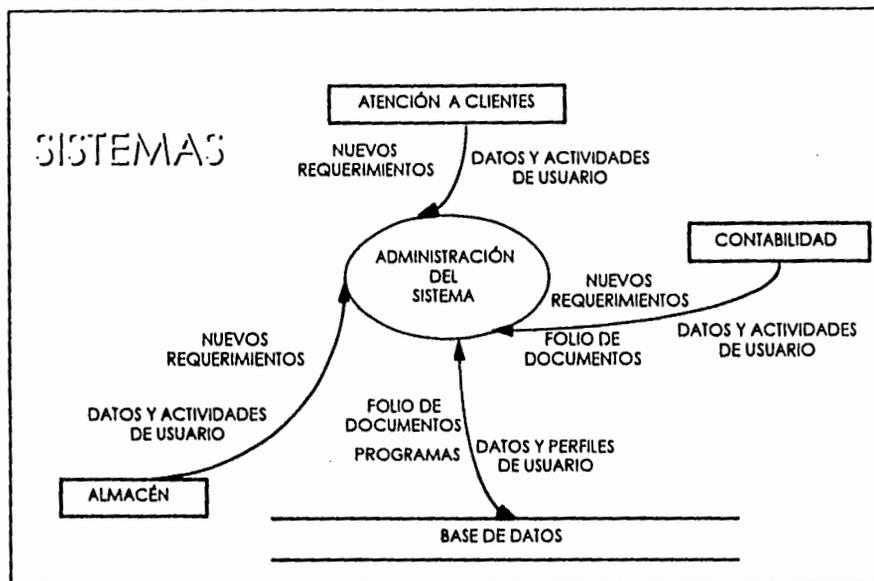


FIGURA 3.16.
DIAGRAMA DE NIVEL 0: SISTEMAS

En el diagrama de nivel 0 de Sistemas se pueden apreciar las tres entidades que harán uso del sistema y los flujos de información que intervienen durante la administración del mismo.

Como se observa, todas las entidades proporcionan información acerca del personal que utilizará el sistema y de las actividades que éstos desempeñan, para asegurar que puedan hacer uso de los programas apropiados para esas actividades, así mismo, informan de nuevos requerimientos de automatización. El almacén de información Base de Datos es en realidad un conjunto de archivos que se detallan en el diagrama de nivel 1 que se muestra a continuación, en la figura 3.17.

En esta diagrama se puede notar que el proceso Administración del Sistema está compuesto por cinco subprocesos: Asignación de Claves, Mantenimiento a Catálogo de Usuarios, Creación de Nuevos Programas, Control de Folios y Configuración de Perfiles. Este último proceso, como su nombre lo sugiere, es el que permite asignar a cada usuario del catálogo los programas e impresora que podrá utilizar; y así crear una versión personalizada de la aplicación salvaguardando la seguridad de los datos.

Los diagramas anteriores permiten determinar que existen muchos procesos que son susceptibles de automatizarse, estos procesos conformarán al nuevo sistema de información que sustituirá al sistema de facturación REAL WORLD. A partir de los diagramas ya citados también se pueden precisar las entidades involucradas en los procesos de facturación, así como los flujos de información que entran y salen al SIFACOR.

Sin embargo, aún cuando existen razones válidas para desarrollar un sistema que sustituya al actual, es necesario determinar si es factible o no.

3.5 FACTIBILIDAD DEL SISTEMA

La etapa de determinación de la factibilidad de este proyecto no cuenta con límites perfectamente establecidos, es difícil decir donde empieza y donde termina el proyecto, ya que, como se ha mencionado anteriormente, el desarrollo del SIFACOR forma parte de un proyecto integral de la dirección de la empresa. Sin embargo, mencionaremos de qué manera se satisfacen los puntos que se consideran más significativos para establecer la factibilidad de un proyecto.

Factibilidad significa -según la mayoría de los autores en la materia- que el proyecto sugerido:
1. Auxilia a la organización a lograr sus objetivos centrales. 2. Es capaz de cubrir las metas con la mayoría de los actuales recursos de la organización en las tres áreas siguientes.
Factibilidad Técnica.
Mejora del sistema actual. Tecnología disponible. Información suficiente. Posibilidad de crecimiento.
Factibilidad Económica.
Costo del estudio de sistemas. Costo del tiempo de los empleados dedicado al estudio. Costo estimado del equipo. Costo del desarrollo-adquisición del software.
Factibilidad Operativa.
¿Será utilizado cuando se instale? ¿Contará con el apoyo de usuarios durante su desarrollo? ¿Será aceptado por los usuarios?

FACTIBILIDAD TÉCNICA

Como parte de un proyecto de integración y estandarización de procesos a nivel nacional, la empresa cuenta ya con los cimientos de lo que, en corto plazo, representará la red de comunicaciones de TELCEL.

Para dar sustento a la información generada en estas oficinas, TELCEL cuenta con la Dirección de Informática que incorpora en su tecnología de operación diversas plataformas de cómputo apoyadas principalmente en equipos VAX de DIGITAL y DPX20 de BULL. La primera de ellas se orienta principalmente a la facturación de tiempo-aire (sistema ISIS) también llamada tasación, que es la que hace llegar al cliente el costo de sostener sus conversaciones telefónicas mediante su infraestructura celular. La complejidad del servicio de tasación es enorme ya que se registran al detalle los tiempos de duración de las llamadas realizadas desde un celular de la empresa, el número telefónico al cual llamó, si fueron llamadas

consideradas locales o roaming⁹, para así poder establecer el costo exacto mensual del servicio de cada cliente. En la plataforma de los equipos DPX20 se corren otras aplicaciones y será ésta la que soporte los servicios de facturación de aparatos, accesorios y servicios en una red a nivel nacional.

Existen dos equipos minis Bull DPX20 cuya plataforma es UNIX, ya comparten recursos mediante TCP-IP aquí en el Distrito Federal, y se planea que esta red crezca al ritmo de instalar y conectar un equipo por mes en cada regional.

Del mismo modo, y como parte de un acuerdo entre TELMEX y la firma de software INFORMIX, se cuenta con una licencia corporativa que permite a TELCEL el uso de esa herramienta. Dado que el producto ha demostrado buenos niveles de desempeño, con grandes volúmenes de información y versatilidad de programación al ser un lenguaje de cuarta generación, convenció a los directivos del grupo para descansar en él, el peso de las principales aplicaciones del consorcio telefónico.

Por otro lado, y conforme al interés de TELCEL por emitir sus propios comprobantes fiscales, se solicitó una demostración al principal proveedor de equipos de impresión de la empresa, la firma XEROX, que afirmaba ofrecer tecnología con esta opción.

La propuesta consistió en la presentación de una impresora láser de mediana velocidad que cuenta con un lenguaje de programación propio, que permite diseñar formatos para la posterior impresión de documentos. Esto es, que mediante dicho lenguaje es posible programar diversos formatos, como los que se podrían elaborar en una hoja de cálculo actual, tales como: tablas, formas de pedido, requisiciones, reportes, facturas y contratos; cargarlos en la memoria de la máquina para luego, desde la aplicación que genera el documento, en este caso SIFACOR, vaciar el reporte en el formato, cuadrando siempre la información perfectamente.

Además, el proveedor se comprometió a digitalizar la imagen de la cédula de identificación fiscal de la empresa, a manera de que, mediante un proceso de división de la imagen, cada pequeña sección pudiera ser declarada como un FONT (Tipo de letra que cada impresora reconoce según su fabricante), para poder también cargarlo en memoria junto con el formato, para su posterior impresión.

FACTIBILIDAD ECONÓMICA

Los análisis de costos de estas alternativas son innecesarios, porque ya estaban contemplados en el proyecto integral arriba mencionado. Se cuenta con la infraestructura de hardware y el software necesarios y el personal que desarrollará este proyecto

En cuanto al análisis costo-beneficio y dadas las dimensiones del proyecto, sus detalles rebasan los alcances de éste trabajo. Sin embargo, aún cuando la cantidad de recursos y el esfuerzo involucrados hacen evidente que se podría tratar de un proyecto caro, se puede afirmar que los beneficios tangibles e intangibles que proporcionará en el corto y mediano plazos superarán, por mucho, a la inversión inicial, resultando así el balance costo-beneficio positivo.

⁹El roaming consiste en el servicio telefónico prestado a usuarios desde un área distintinta a aquella donde se contrató el servicio (abonados foráneos), a algunos de estos usuarios se les factura en dólares.

FACTIBILIDAD OPERATIVA

La factibilidad operativa del sistema de información también presenta una gran ventaja generada por su antecesor.

Dado que REAL WORLD no ofrece integridad en la información, la Secretaría de Hacienda y Crédito Público no lo calificó satisfactoriamente para ser el software que la empresa explotará para facturar, de este modo incluso los usuarios más reacios tendrán que alinearse al uso del nuevo sistema, ya que, en contraste a lo que ocurre con otras aplicaciones, el presente cambio se impone desde el exterior de la empresa. Se prevén, sin embargo, ciertas actitudes de sabotaje, mismas que se piensa referirán un panorama de aquellas personas que encontraban "beneficios personales" en el sistema anterior, ya que sí se tienen noticias de algunos fraudes.

Como podemos darnos cuenta, la valorización de la factibilidad del sistema resulta positiva, ayudada en gran medida por el contexto de la Dirección de Informática de la empresa y el momento mismo que vive TELCEL.

CONCLUSIONES

Mediante un análisis preliminar se observó que el sistema actual deberá ser reemplazado por no satisfacer las necesidades de los interesados y por la imposibilidad de modificarlo.

A través del análisis estructurado se estudiaron las actividades y procesos operativos involucrados en la facturación de productos y servicios, lo que permite inferir las principales funciones que deberá ejercer el nuevo sistema de información.

El nuevo sistema será desarrollado en fases: la primera de ellas se liberará en seis meses y dará atención principalmente al proceso de facturación que se lleva a cabo en las oficinas de Atención a Clientes, así como a algunos procesos contables importantes. Contemplará además la elaboración de reportes y consultas específicos.

En fases posteriores, que exceden las perspectivas del presente trabajo, se dará atención a otras funciones que deberá llevar a cabo el SIFACOR como, por ejemplo, el enlace con el sistema de Caja, el establecimiento de consultas de tipo abierto para la toma de decisiones, la inclusión de consultas con HyperScript, etc.

CAPÍTULO 4

DISEÑO

El diseño de software es un proceso mediante el cual se traducen los *requisitos* detectados en el análisis en especificaciones de lo que debe hacer el sistema, para finalmente obtener una representación en software.

Antes de comenzar con el diseño del SIFACOR, vale la pena recordar su misión principal:

El SIFACOR tiene como objetivo primordial la administración de la información generada por las ventas y movimientos contables de la empresa, así como la impresión de documentos de valor fiscal con buena presentación.

Para cumplir con este objetivo, como paso fundamental en el diseño, consideraremos las especificaciones del sistema principal y sus subsistemas.

4.1. ESPECIFICACIONES DEL SISTEMA PRINCIPAL

Las especificaciones del sistema principal resultan de agrupar los procesos representados en los diagramas de nivel del capítulo anterior, según la semejanza de sus funciones, hasta obtener finalmente la estructura principal del SIFACOR, que se compone de cuatro módulos funcionales, como se muestra en la figura 4.1.

Como se puede apreciar, cada uno de los módulos tiene objetivos bien definidos que se identifican por su nombre: Consultas y Reportes, Mantenimiento de Catálogos, Facturación y Administración del Sistema.



FIGURA 4.1.
ESTRUCTURA LÓGICA DEL SIFACOR

4.2. ESPECIFICACIONES DE LOS SUBSISTEMAS

Es en este apartado del capítulo donde se describen las opciones que el sistema proporcionará al usuario

- ⇨ **MÓDULO CATÁLOGOS.** Los catálogos permitirán registrar y controlar las tablas o listas de información que son consultadas constantemente por el SIFACOR.
- ⇨ **MÓDULO FACTURACIÓN.** Será la parte central del SIFACOR, permitirá registrar y controlar todo lo referente a las transacciones de ventas de la empresa, es decir, que desde esta opción se generarán los principales documentos de SIFACOR.
- ⇨ **MÓDULO CONSULTAS Y REPORTE.** Su finalidad será proporcionar información de forma fácil y eficiente, a través de pantalla o papel, sobre los diferentes aspectos del proceso de ventas.
- ⇨ **MÓDULO DE ADMINISTRACIÓN DEL SISTEMA.** Su objetivo será facilitar la generación y modificación de configuraciones al sistema o perfiles, permitiendo armar las vistas del SIFACOR conforme a las necesidades de cada usuario, sin descuidar la integridad y seguridad de la información.

MÓDULO DE CATÁLOGOS

Este módulo está concebido para que el usuario consulte fácilmente la información que más comúnmente requerirá para generar sus documentos; ya sean facturas, notas de crédito, notas de cargo o contratos.

La administración de estos catálogos permitirá el debido control de acceso para consulta y mantenimiento de los datos ahí contenidos, de tal manera que sea posible interactuar correctamente con los demás componentes del SIFACOR.

Esta función administrativa corresponderá en algunos casos al usuario directo y en otros al administrador del sistema.

A continuación, se describe el contenido de los catálogos que sustentan al SIFACOR.

DESCRIPCIÓN DE LOS CATÁLOGOS

Regiones

Debido a que TELCEL presta sus servicios a nivel nacional, se adoptó la convención de dividir al territorio nacional en nueve regiones de servicio. Por lo tanto, es necesario contar con una tabla que contenga la clasificación de cada una de dichas regiones.

Tiendas

A los diferentes Centros de Servicio que se encuentran en cada región, se les identifica como tiendas, por esto es necesario contar con los datos (dirección, teléfono, etc.) de dichas tiendas, así como de los servicios que prestan.

Proveedores

Este catálogo contendrá los datos actualizados de los proveedores de TELCEL. Esta información permitirá el enlace oportuno con los proveedores para contar en los almacenes con los insumos necesarios para la venta.

Clientes

Este catálogo contendrá la información correspondiente a cada uno de los clientes existentes en el sistema; se podrá utilizar para consulta interactuando con otros componentes del SIFACOR, como el módulo de reportes y el de facturación en sus correspondientes notas de crédito, de cargo, contratos, etc.

Cuentas

Este catálogo mostrará la relación entre cuenta y producto, así como el monto acumulado de cada cuenta dentro del sistema, y se podrá alterar llamando directamente al catálogo o bien al llevar a cabo cualquiera de las siguientes actividades: transacciones de venta, la asignación de crédito al cliente o al efectuar la carga de inventarios del almacén.

Productos y servicios

Contendrá la clasificación de los productos existentes en el almacén (teléfonos y accesorios) y los servicios que proporciona TELCEL. También contará con un control que determina el mínimo de existencias que debe tener el almacén y alertará a los usuarios cuando se llegue a

este límite. Así mismo, inhibirá la captura de ventas cuando ya no existan los productos solicitados en el almacén.

Es importante mencionar que la alteración de este catálogo se puede hacer directamente, o bien, al hacer alguna venta, emitir notas de crédito ó cargo, o a través de la carga de inventarios.

Planes tarifarios

Debido a la gran variedad de productos y servicios que ofrece TELCEL, y como parte de sus campañas de venta, se ha creado lo que se denomina planes tarifarios que consisten en la agrupación de algunos servicios y la venta de éstos como un solo paquete por una tarifa específica. Estos planes tarifarios se deberán poder consultar en el momento en que se efectúa una venta.

MÓDULO DE FACTURACIÓN

Es la parte medular del SIFACOR, su principal función será permitir el registro y control de las transacciones de venta de productos y servicios al usuario, proporcionando la facilidad de capturar eficazmente la información necesaria para elaborar un documento de venta en presencia del cliente, por lo tanto deberá tratarse de un proceso ágil y oportuno.

Este módulo también será explotado por entidades de control interno de la empresa, como por ejemplo el área de Contabilidad, donde se generarán notas de crédito o cargo.

Este módulo o subsistema será muy completo, por lo que se describirán las funciones u opciones que cumplirá.

DESCRIPCIÓN DE LAS OPCIONES DE FACTURACIÓN

Facturación

Permitirá la captura de los datos que conforman una factura a través de la pantalla. En los datos se reflejan los generales del cliente, su condición fiscal, la forma en que se hará el pago - efectivo, cheque o tarjeta -, el listado de conceptos que se facturan y, obviamente, los montos de los mismos, el subtotal, el I.V.A. y el total.

Notas de Crédito

Una nota de crédito puede contener toda la información de una factura, aunque requiere validar información adicional; como el caso del folio de la factura de referencia y que el concepto que se quiera cancelar realmente forma parte de esa factura.

Notas de Cargo

Una nota de cargo funciona de manera similar a una nota de crédito. Así como una nota de crédito puede cancelar total o parcialmente a una factura, una nota de cargo puede cancelar a una nota de crédito, total o parcialmente. Una nota de crédito tiene el mismo formato que tiene una factura.

Facturación Roaming

Dentro de la gama de documentos que este sistema debe generar, se contempla la posibilidad de facturar en dólares; dado que la empresa presta servicios a un buen número de clientes foráneos que prefieren hacer sus pagos en esa moneda.

Reimpresión

Proveerá al usuario la posibilidad de obtener copias de un documento generado previamente. A veces se requiere reponer la factura a solicitud de un cliente - por ejemplo, en caso de extravío-, o bien, se puede reimprimir para dar seguimiento a un proceso de control interno.

Reimpresión Múltiple

A diferencia del módulo anterior, no genera uno a uno los documentos. Algunas áreas de Contabilidad, por ejemplo, Egresos, elaboran un gran número de reimpresiones. Contemplando esta necesidad se desarrollará la opción que permita la reimpresión de documentos por lote.

Contratos Celulares

Uno de los objetivos que dieron pie a la creación del SIFACOR fue el de poder dar a los documentos de contratación de servicios celulares una presentación apegada a los estándares establecidos. Así, esta opción permite la captura de los datos que se requieren en la elaboración de los contratos para su posterior impresión o reimpresión.

MÓDULO DE CONSULTAS Y REPORTES

La información de los documentos generados por este módulo fungirá como el principal parámetro, para conocer el desempeño de las áreas más importantes involucradas por el sistema, además de proveer de forma clara y organizada la información que se ha integrado al sistema en las diversas etapas de su operación. Sin embargo, y a pesar de su indudable importancia, este módulo como tal puede tratarse de un mero concepto ya que será posible que en la configuración de algún usuario no aparezca como un módulo, sino que sus opciones se encuentren diseminadas como opciones adicionales dentro de otros módulos. Las opciones a continuación descritas se desarrollarán para atender las necesidades más apremiantes de las áreas.

Consulta de facturación diaria

Este programa se concibe como una opción que permita al administrador de cada tienda conocer los montos de venta de sus asesores, cuestión que servirá para determinar la capacitación, rotación y motivación pertinente de este tipo de personal que resulta fundamental por representar la imagen de la empresa ante el cliente. Por otro lado, los montos de venta se deberán detallar en esta opción al grado de poder identificarlos por factura ó por concepto, es decir, que se pueda determinar el monto de venta de un artículo o un servicio en particular.

La "Consulta de facturación diaria" de SIFACOR atenderá los requerimientos recabados con las áreas de Ventas Corporativas, Atención a Clientes y Contabilidad.

Reporte de inventarios

El "*Reporte de inventarios*" pretende sobre todo cubrir las necesidades expuestas en las entrevistas con el área de almacén, que mostraron la necesidad de poder monitorear constantemente los puntos de reorden¹ de los diversos productos, para poder determinar el mejor momento de una nueva carga. Además de los puntos de reorden se determinó junto con el usuario la utilidad de conocer la cantidad existente en el inventario del resto de los productos aún sin encontrarse en existencias mínimas, con el fin de observar también el comportamiento de los artículos de mayor demanda.

Consulta de productos

Consulta de productos se diseña para atender las solicitudes de las áreas como: Almacén, Ventas corporativas y Contabilidad. Provee la facilidad de conocer la mayoría de la información que se asocia a un producto ó a varios según su tipo. Una vez que se seleccione el producto de interés el usuario podrá generar los siguientes reportes: del proveedor, reportes con los datos generales del producto y de las facturas que lo incluyen. Este reporte se diseña principalmente como un reporte complementario al anterior, que permite, además, determinar en que facturas aparece como concepto. Con esta facilidad se pretende rastrear incluso la preferencia del cliente hacia tal o cuál producto.

Reporte acumulado de ventas por asesor

La presente opción se desarrollará sobre todo atendiendo los requerimientos del área de Ventas corporativas y Atención a Clientes, y funcionará como un reporte que agrupe más información en relación al de "*facturación diaria*", ya que se incluirán además fechas y periodos de venta.

MÓDULO DE ADMINISTRACIÓN DEL SISTEMA

Buscando la forma más adecuada para que SIFACOR satisfaga los requerimientos de los usuarios, sin sacrificar la integridad de la información, se determinó la necesidad de controlar el acceso a los diversos programas que contendrá la aplicación. La fórmula más sencilla para lograr este control es proveer a cada usuario única y exclusivamente el acceso a aquellos programas que necesite sin que pueda alcanzar funciones más allá de las que específicamente le corresponde realizar, y que de la misma manera solo él pueda tener acceso a los programas que se le asignen. Para lograrlo se propuso la creación de "*perfiles de usuario*", que son en realidad versiones personalizadas de SIFACOR para cada usuario, según sus necesidades, es decir, si un usuario, por ejemplo de almacén, no necesita acceder programas más allá del catálogo de productos y servicios y algunos reportes, no tiene caso tentar su curiosidad con un programa que inicialice los folios de la facturación o con reportes que arrojarán datos irrelevantes para él. Así el perfil en SIFACOR de este usuario de almacén será distinto al de un usuario de atención a clientes, y a su vez distinto al de un usuario de contabilidad; incluso usuarios de la misma área, como la contable, pueden tener perfiles distintos según las actividades que realicen, y cada uno contar con un password ó palabra clave que asegure que sólo él puede hacer uso de su perfil.

Dadas las consideraciones anteriores, se determinó la importancia de desarrollar una serie de programas que estructurarán los perfiles de usuario, conteniendo la información de los usuarios del sistema, el área a la que pertenecen, etc. Estos programas conformarán el

¹ Avisos cuando un artículo llega a su menor existencia permitida

denominado "*Módulo de administración del sistema*" y corresponderá al área de sistemas su operación y mantenimiento.

A continuación se describen las opciones que contendrá el módulo de Administración del Sistema que fungirá como un control de SIFACOR.

Catálogo de usuarios del sistema

En este catálogo se almacenará la información más relevante de los usuarios del sistema, además del nombre que recibe en la red la impresora que corresponde a su área. De esta forma se canalizará correctamente el envío de los documentos, tanto en los centros de atención como para los usuarios internos de TELCEL. Desde esta opción también se podrá dar mantenimiento a los passwords de usuario según se necesite.

Catálogo de programas

Desde esta opción se dará mantenimiento a la aplicación en cuanto a los programas que la conforman y, por ende, facilitará el alta de programas que se vayan incluyendo durante la etapa de mantenimiento del SIFACOR. Es desde este catálogo donde se tomará la información que permitirá la creación de los perfiles. Los programas se registrarán tanto por nombre físico como por título o nombre lógico y un número que facilite su identificación.

Perfiles de usuario

Como ya se explicó el *perfil de usuario* es una versión personalizada de SIFACOR para cada usuario según su actividad y requerimientos, desde aquí se asociará a un usuario con los programas a los que debe tener acceso a manera de conformar su perfil en la aplicación. Los programas deben poder integrarse en orden de menú y submenú; por tanto el programa debe poder identificar si el menú será "padre" de otros menús ó si contendrá programas ejecutables directamente para poder agruparlos según lo que se considere conveniente dentro de cada menú.

Controlador de folios

SIFACOR surgirá como un nuevo sistema que substituirá a otro que se encuentra actualmente en operación, tanto aquí en el área metropolitana como a futuro en las regionales; cuando SIFACOR se libere deberá ser capaz de retomar la numeración de folios de los documentos que actualmente genera RW. "*Controlador de folios*" permitirá inicializar la numeración dando de alta aquel folio con el cuál deba comenzar la numeración de los nuevos documentos que genere.

4.3. DISEÑO DE LA BASE DE DATOS

Conforme a las especificaciones de diseño ya determinadas en el análisis se prevé la necesidad de contar con información disponible para todos los usuarios de la aplicación; esta información debe ser íntegra y consistente, para que su uso sea confiable, además de que su actualización debe ser sencilla. Un sistema de almacenamiento con estas características recibe el nombre de Base de Datos (BD). Algunas definiciones mencionan que una BD es una colección de datos significativos, interrelacionados y almacenados, que no contiene redundancias innecesarias. Su finalidad es servir a una aplicación o más, de la mejor manera posible. Los datos se almacenan de modo que resulten independientes de los programas que los usan, y se emplean métodos bien determinados para incluir datos nuevos y para modificar

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

o extraer los datos ya existentes. La interrelación de la información de la BD se obtiene de las relaciones que se establezcan entre los datos y no de su lugar de almacenamiento físico.

Base de datos relacional

Una base de datos relacional consiste en una colección de tablas. Cada tabla es una estructura en forma de arreglo rectangular de filas y columnas que mantiene datos de una entidad. Las filas de la tabla representan los registros y las columnas representan los atributos de la entidad.

El diseño de las tablas deberá seguir las siguientes reglas:

1. Cada tabla deberá tener un nombre que será único dentro de la base de datos
2. Una tabla está compuesta de una o más columnas.
3. Cada columna deberá tener un nombre que será único dentro de la tabla. El nombre de la columna puede existir en otra tabla.
4. Una tabla puede contener cero o más filas de datos.
5. La posición física de filas y columnas dentro de una tabla no tiene significación.

Llaves primarias y foráneas

La llave primaria de una tabla es una columna o grupo de columnas cuyo valor único identifica cada fila. Cada tabla debe tener una llave primaria; esta llave no debe ser un valor nulo, blanco o cero. La llave primaria posibilita localizar una fila en una tabla.

Una llave foránea es una columna o grupo de columnas en una tabla que contienen valores de llave primaria en otra tabla.

Las asociaciones entre filas de una y otra tabla son realizadas por medio de la llave foránea. Las dos tablas tendrán una columna o grupo de columnas en común para poder hacer la relación o las asociaciones.

En el diseño de una BD se deben de seguir los siguientes pasos:

1. Identificar entidades.
Definir tablas con sus llaves primarias.
2. Definir relaciones.
Identificar las relaciones entre las tablas.
3. Listar los atributos.
Adicionar o modificar las características de cada tabla

Las principales ventajas que ofrece un sistema de BD son las siguientes:

- **Permite la concurrencia en la información.**
Varios programas de usuario pueden acceder sobre un dato en particular, sin que éste deba almacenarse en más de un lugar físicamente.
- **Evita la inconsistencia.**
En un buen diseño de una BD, de un dato en particular, no aparece más que en las ocasiones estrictamente necesarias, como es el caso de las llamadas llaves foráneas ó aquellos casos en los que la normalización² de tablas así lo requiera.
- **Controla la integridad.**
Cuando una BD incluye información utilizada por muchos usuarios, es importante que no sea fácil destruir los datos almacenados ni sus relaciones. Ocasionalmente se producirán fallos de hardware y distintos tipos de accidentes. El almacenamiento de los datos y los procedimientos de actualización e inserción deben asegurar que el sistema pueda recuperarse de estas contingencias sin daño para los datos.
- **Pueden aplicarse restricciones de seguridad.**
Una base de datos facilita la restricción de acceso a la información salvaguardándola de malos usos, destrucción o alteración ya sea accidental o intencional por parte de personas no autorizadas.

Dentro de sus desventajas podemos mencionar:

La seguridad y la integridad pueden comprometerse si no se establece un buen control, ya que todos los datos se almacenan en un solo lugar y en consecuencia son más vulnerables a accidentes.

El costo de mantenimiento de una BD siempre será mayor comparado con el de un sistema de archivos convencional, ya que los controles anteriormente descritos, además de respaldos, mantenimiento, etc. siempre asocian un desembolso mayor para las organizaciones.

Un factor destacado que incrementa la dificultad de diseño de una BD es la variedad de datos que reclaman ser organizados, ya que comúnmente los usuarios tienen exigencias diversas. Esta variedad de datos requiere ser almacenada de manera que se les pueda utilizar independientemente desde una gran variedad de aplicaciones, para que asimismo puedan ser actualizados fácil y rápidamente por los mismos usuarios. Por tal motivo, los siguientes son aspectos importantes a considerar para el diseño de una BD:

- Los datos deben ser independientes de los programas que los utilizan, de tal modo que se puedan enriquecer o reestructurar sin que resulte necesario modificar los programas existentes.
- Debe ser posible explorar la BD y obtener información de ella sin necesidad de recurrir a la tarea de escribir programas en los lenguajes convencionales.

² Normalización es un proceso de depuración y simplificación de las primeras estructuras propuestas para las tablas de una BD, ya que comúnmente en esa primera intención los diseños resultan bastante redundantes.

DIAGRAMAS ENTIDAD-RELACIÓN

Una vez descritas las especificaciones principales del sistema se continuará con el proceso de diseño en el que se analizarán las entidades, la relación y los atributos que intervienen en la generación de los documentos que el SIFACOR debe emitir. Para lo cual se emplearán los "Diagramas Entidad -Relación"; donde se describen las relaciones existentes entre las entidades involucradas en el proceso de interés, así como los atributos inherentes a cada entidad.

Pero ¿Qué es lo que se debe entender por Entidades, Relaciones y Atributos?:

ENTIDADES: Son cualquier objeto o evento, acerca del cual, se recolectan datos. Una entidad puede ser una persona, un lugar o un objeto.

RELACIONES: Las Relaciones son asociaciones surgidas entre entidades. Dentro de los diferentes tipos se tienen:

RELACIÓN 1:1 (UNO A UNO)

RELACIÓN 1:M (UNO A MUCHOS)

RELACIÓN M:N (MUCHOS A MUCHOS)

ATRIBUTO: Son las características o valores que tienen las entidades.

El conjunto de estos tres factores conforman gráficamente el **DIAGRAMA DE ENTIDAD-RELACIÓN**.

DIAGRAMA DE ENTIDAD-RELACIÓN (E-R). Su principal propósito es representar los objetos que de alguna forma se involucran en el proceso de interés y las relaciones que surgen entre ellos. La notación de diagrama E-R es relativamente sencilla: Los objetos de datos se representan como rectángulos etiquetados, las relaciones se representan mediante rombos y los atributos mediante óvalos.

Las conexiones entre los objetos de datos y las relaciones se establecen mediante variadas líneas de conexión.

A continuación se muestra el **Diagrama de Entidad Relación general para el sistema SIFACOR**, así como, su desglose por relación, en donde se describen los atributos pertenecientes a cada Entidad.

En la figura 4.2. se muestra el diagrama entidad-relación de los objetos que intervienen en los procesos relacionados con la facturación de artículos y servicios telefónicos. Así, por ejemplo, el objeto documentos está relacionado por un lado con cliente a través de la relación *requiere* y, por otro, con el objeto de productos y servicios a través de la relación *se asocia*. Esto indica que el objeto documentos en la base de datos que se diseñará, deberá tener un campo que permita relacionarlo con las tablas que contendrán información del cliente y de productos y servicios respectivos.

Otra observación pertinente es que la relación entre los objetos se muestra con su cardinalidad, es decir, los límites de los valores que son permitidos por esa relación.

De esta manera, un cliente puede requerir de uno o más documentos, y cada uno de éstos a su vez, puede estar asociado a uno o varios productos y servicios.

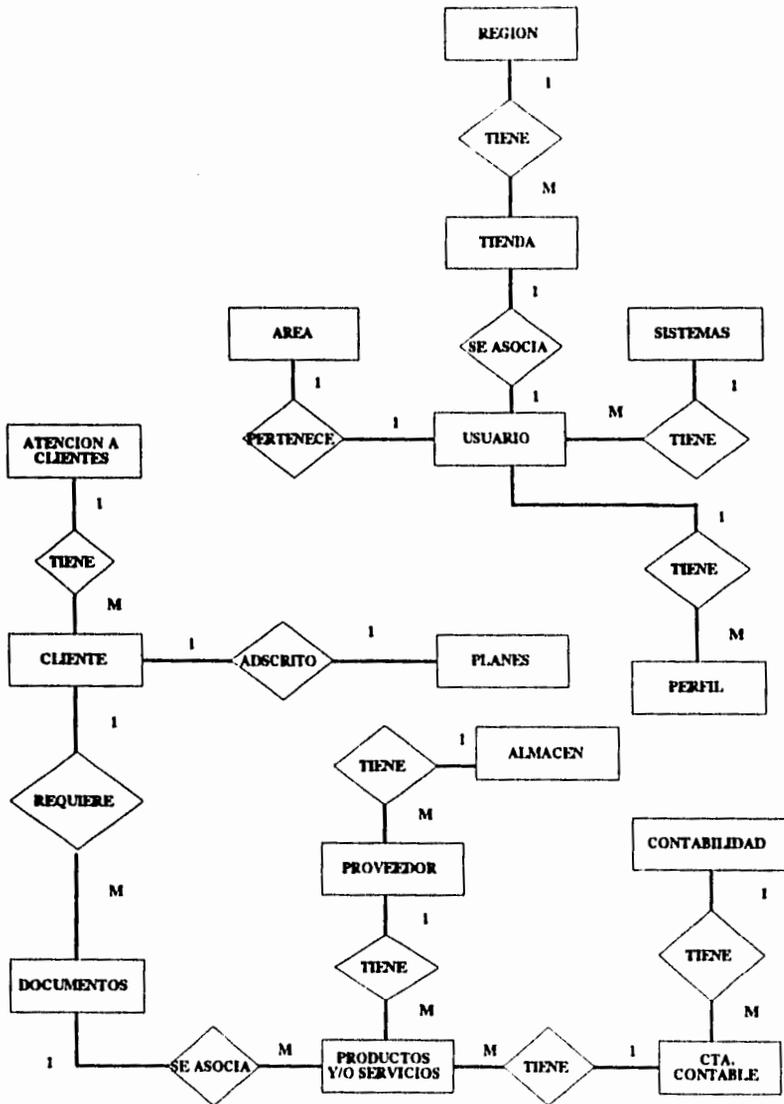


FIGURA 4.2.
DIAGRAMA DE ENTIDAD RELACIÓN GENERAL

A continuación se detalla el resto de las relaciones expresadas en este diagrama.

En la figura 4.3. se ilustra la relación entre los objetos cliente y documentos, la relación está dada porque un cliente puede requerir uno o más documentos, entendiéndose por éstos últimos a los contratos, facturas, notas de crédito y notas de cargo. En este diagrama E-R, también se muestran los atributos de estos objetos.

Los atributos de un objeto en particular deben ser capaces de identificarlo completamente. Tal es el caso del objeto cliente, donde observamos que con los atributos nombre, domicilio y teléfono queda completamente reconocido.

DIAGRAMA DE ENTIDAD RELACION CLIENTE
- DOCUMENTO

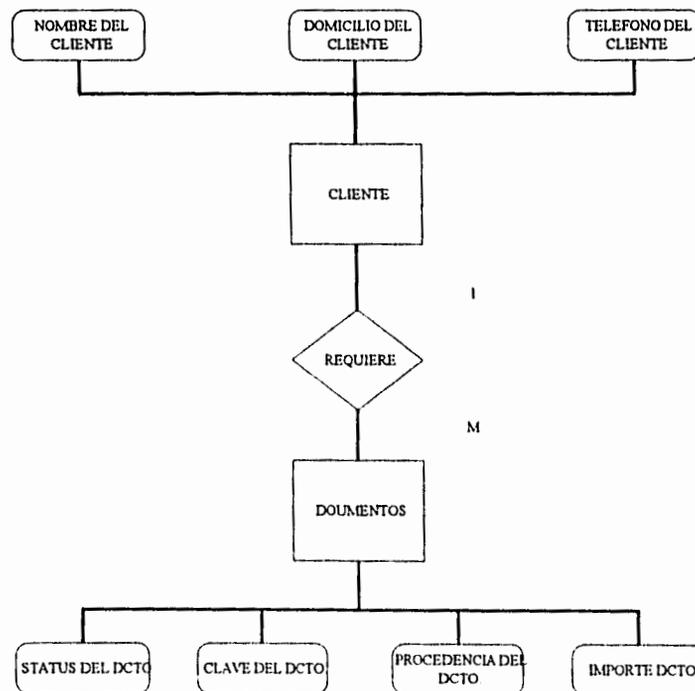


FIGURA 4.3
DIAGRAMA E-R CLIENTE-DOCUMENTOS

En la figura 4.4. se aprecia el diagrama E-R de los objetos cliente y planes, mismos que están asociados a través de la relación *adscrito a*, que significa que cuando un cliente desea adquirir un bien o servicio de TELCEL, se la adscribe forzosamente a un plan de venta. La relación es uno a uno; un cliente sólo puede estar adscrito a un único plan. Además se puede observar que el objeto planes tiene como atributos: clave y descripción.

DIAGRAMA DE ENTIDAD RELACION
CLIENTE-PLANES

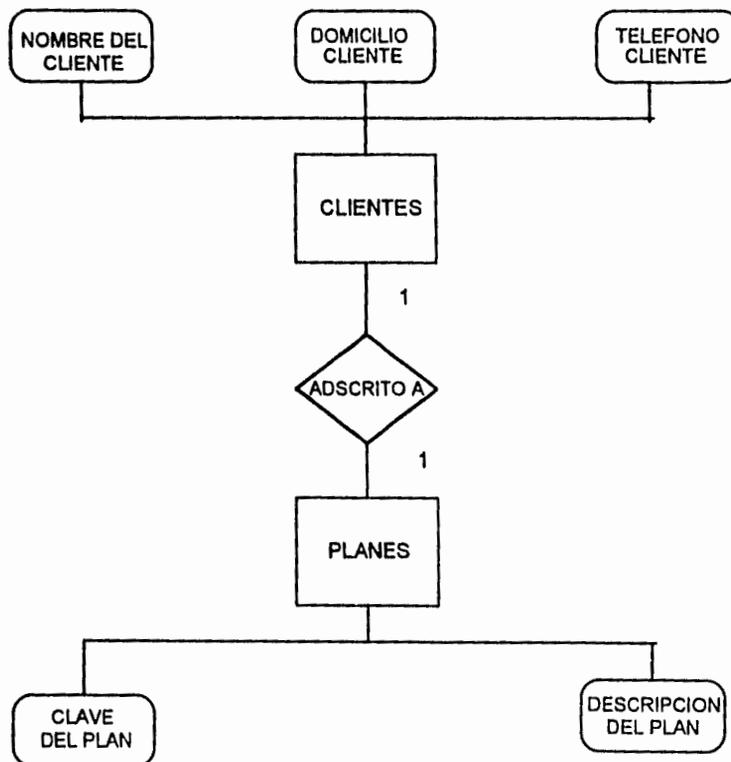


FIGURA 4.4.
DIAGRAMA E-R CLIENTE-PLANES

En la figura 4.5. se muestra la relación entre documento y productos y servicios. Aquí se puede apreciar que uno o más productos y servicios están asociados a un documento. Por ejemplo, en una factura se pueden registrar uno ó más servicios y

productos (teléfono, buzones, accesorios, servicios especiales, etc.) que un cliente desee contratar y adquirir.

La figura también muestra los atributos del objeto documentos: estatus, clave, procedencia e importe; y los atributos del objeto producto: clave, descripción y tipo.

DIAGRAMA DE ENTIDAD RELACION
DOCUMENTO - PRODUCTOS Y SERVICIOS

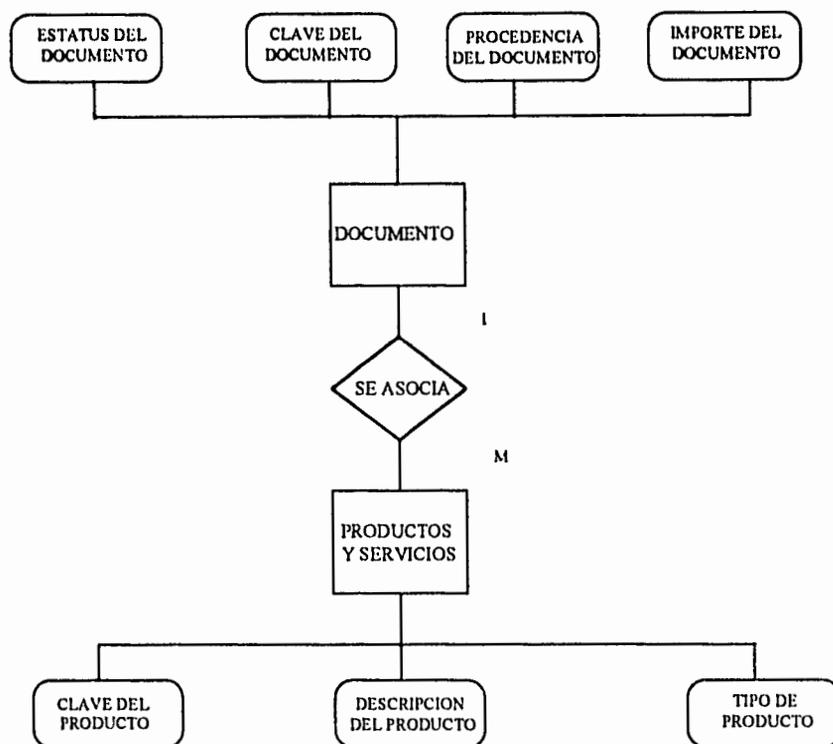


FIGURA 4.5.
DIAGRAMA E-R DOCUMENTO-PRODUCTO Y SERVICIOS

En la figura 4.6. se muestra el diagrama E-R de los objetos usuario y tienda. Como se puede observar cada usuario está asignado a una sola tienda. Un usuario es un asesor de ventas de TELCEL, por eso trabaja o está adscrito a una sola tienda. Los atributos del usuario son: clave, password, nombre y fecha de ingreso. Los atributos de una tienda son: nombre y domicilio.

DIAGRAMA DE ENTIDAD RELACION
USUARIO-TIENDA

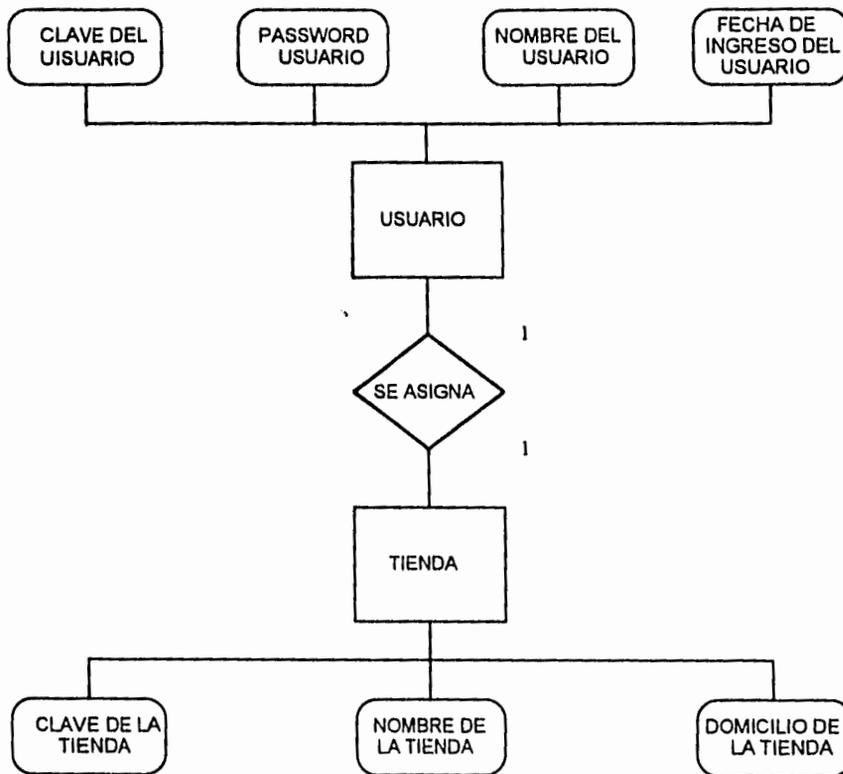


FIGURA 4.6.
DIAGRAMA E-R USUARIO-TIENDA

En la figura 4.7. se observa como se relacionan los objetos usuario y perfil. El perfil es la configuración que tendrá la vista del programa para un usuario en particular. Un usuario solo podrá tener un perfil. Vale la pena hacer mención que esta relación se llevará a cabo a través de la administración del sistema.

DIAGRAMA DE ENTIDAD RELACION
USUARIO-PERFIL.

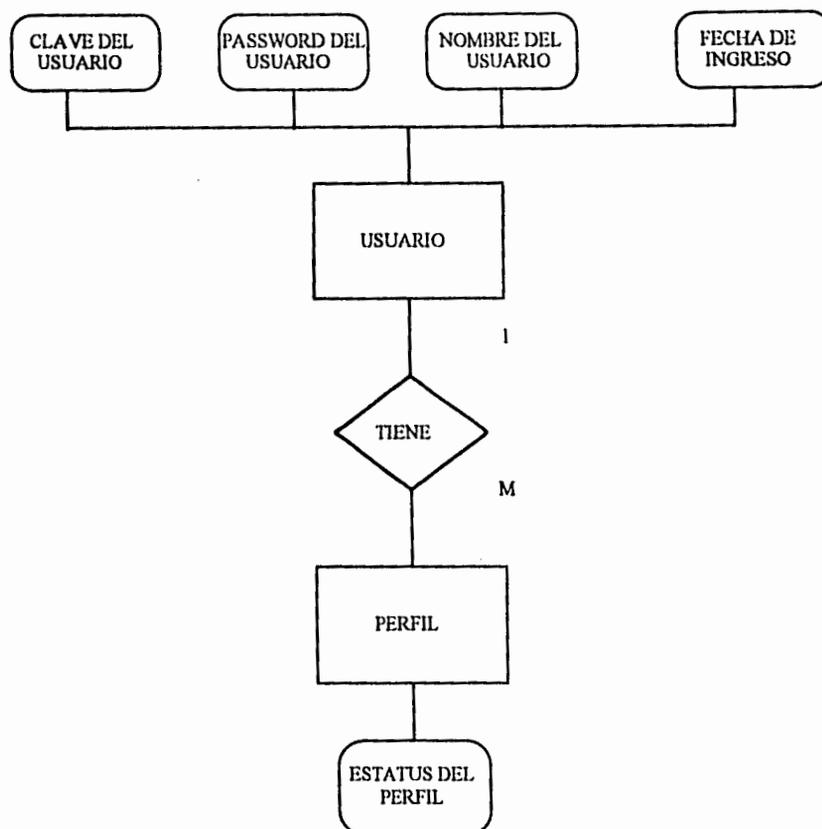


FIGURA 4.7.
DIAGRAMA E-R USUARIO-PERFIL

En el diagrama E-R de la figura 4.8. se muestra la relación entre los objetos usuario y área; el área es la actividad a la que se dedica el asesor o usuario, pudiendo ser: contable, ventas, sistemas ó almacén. Esta relación es útil para poder determinar el perfil más adecuado para cada usuario en particular. Un usuario solo puede pertenecer a un área a la vez. Los atributos del área son clave y nombre.

DIAGRAMA DE ENTIDAD RELACION
USUARIO-AREA

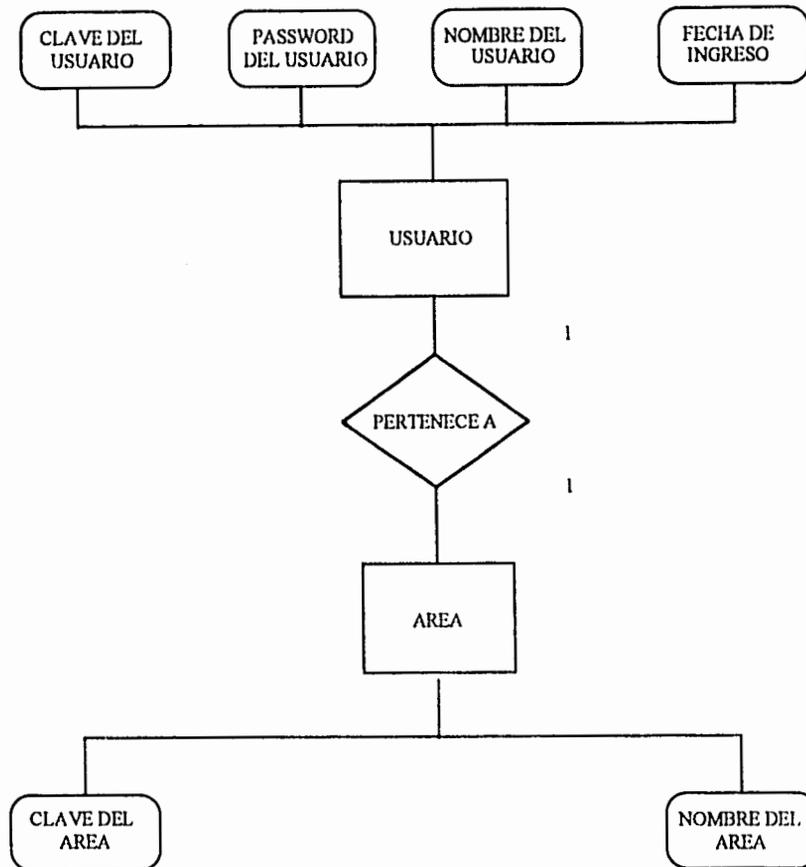


FIGURA 4.8.
DIAGRAMA E-R USUARIO-ÁREA

En la figura 4.9. se muestra el diagrama E-R de los objetos región y tienda. Además se observa que la relación está dada por que una tienda pertenece a una sola región; pero una región puede tener una o más tiendas. Las regiones son las nueve áreas en que TELCEL ha dividido el territorio nacional para prestar sus servicios con una adecuada cobertura en todo el país. Se puede apreciar que los atributos de una región son: clave y descripción.

DIAGRAMA DE ENTIDAD RELACION
REGION - TIENDA.

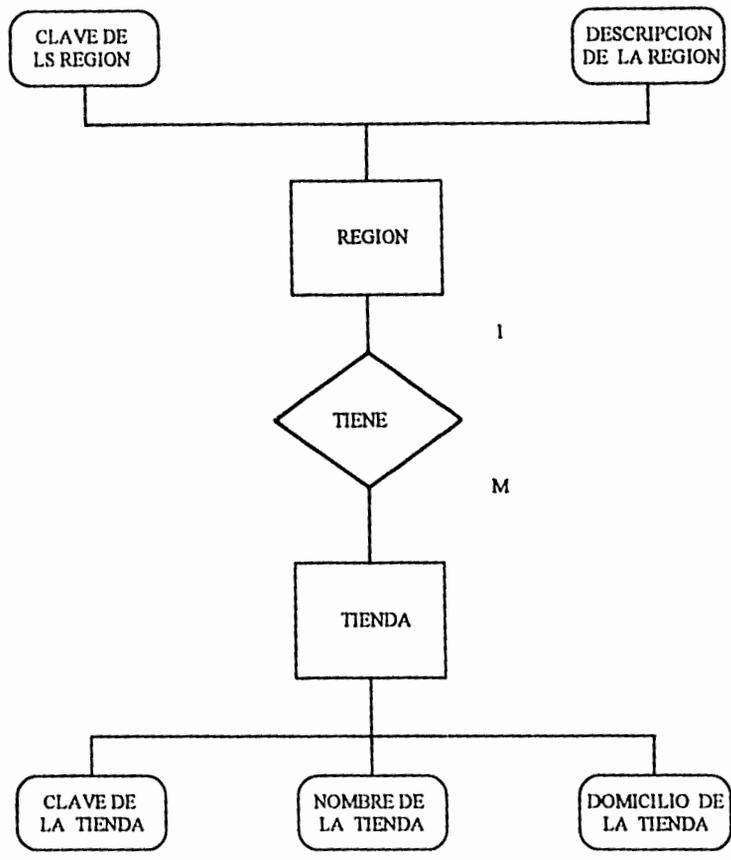


FIGURA 4.9.
DIAGRAMA E-R REGIÓN-TIENDA

En la figura 4.10. se muestra el diagrama E-R de las entidades proveedores y productos y servicios. Estos dos objetos se corresponden a través de la relación *proporciona*, es decir, un proveedor proporciona uno o más productos y servicios a TELCEL. A su vez, un determinado producto y servicio es suministrado por un solo proveedor. Los atributos del objeto proveedor son: nombre, clave y domicilio.

DIAGRAMA DE ENTIDAD RELACION
PROVEEDOR- PRODUCTOS Y SERVICIOS

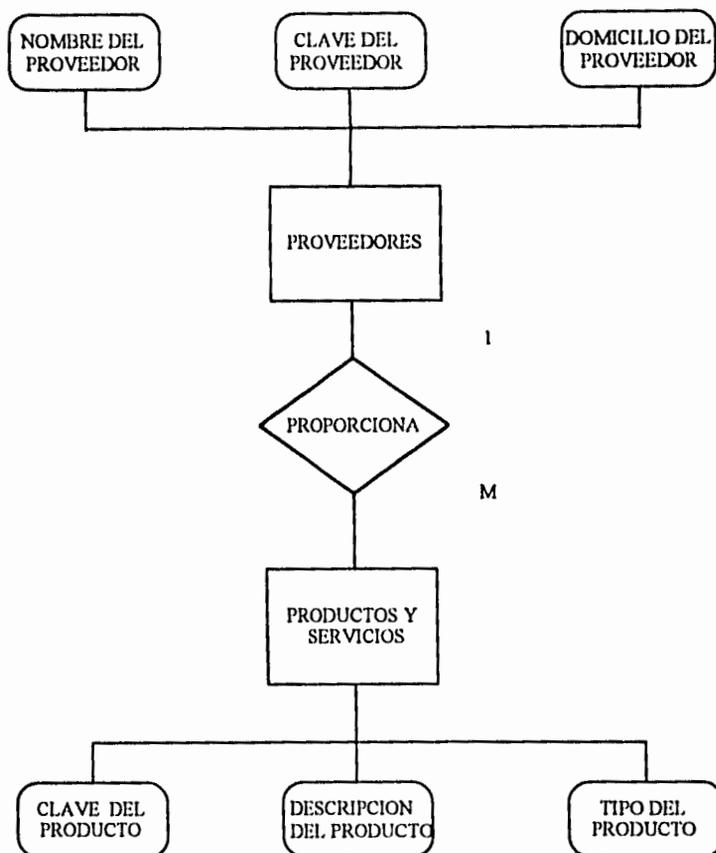


FIGURA 4.10.
DIAGRAMA E-R PROVEEDOR-PRODUCTOS Y SERVICIOS

En la figura 4.11. se aprecia la relación que existe entre los objetos cuenta contable y productos y servicios. Se indica que una cuenta contable tendrá asociada por lo menos un producto y servicio. Cuando se efectúe un movimiento contable por la venta de algún producto o servicio se afectará la cuenta contable relacionada. Los atributos de ésta son: nombre, clave, monto.

DIAGRAMA DE ENTIDAD RELACION
CUENTA CONTABLE-PRODUCTOS Y SERVICIOS

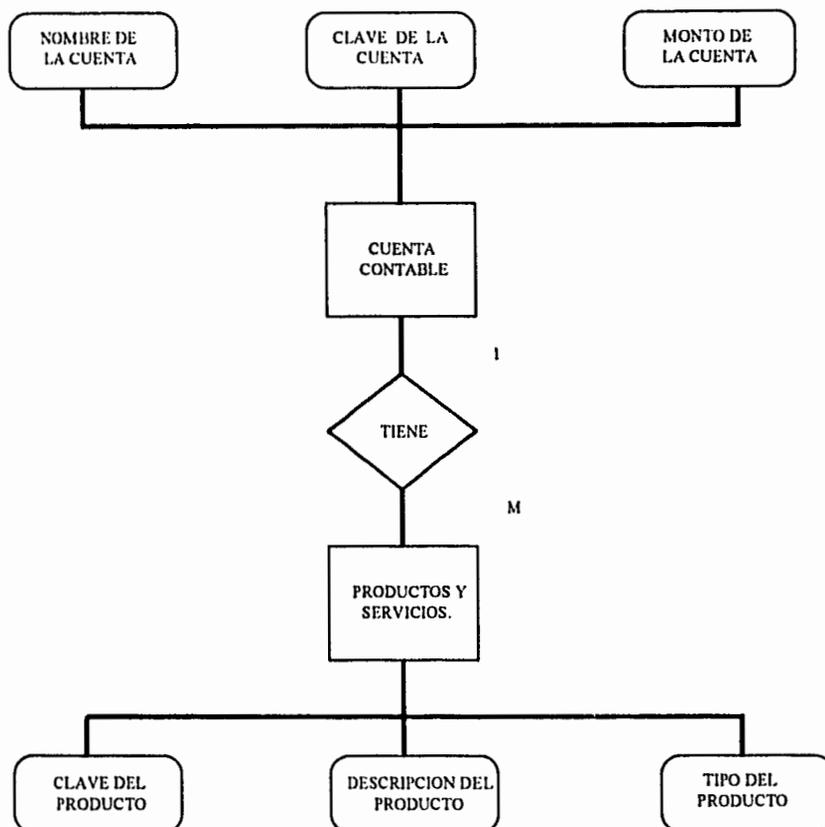
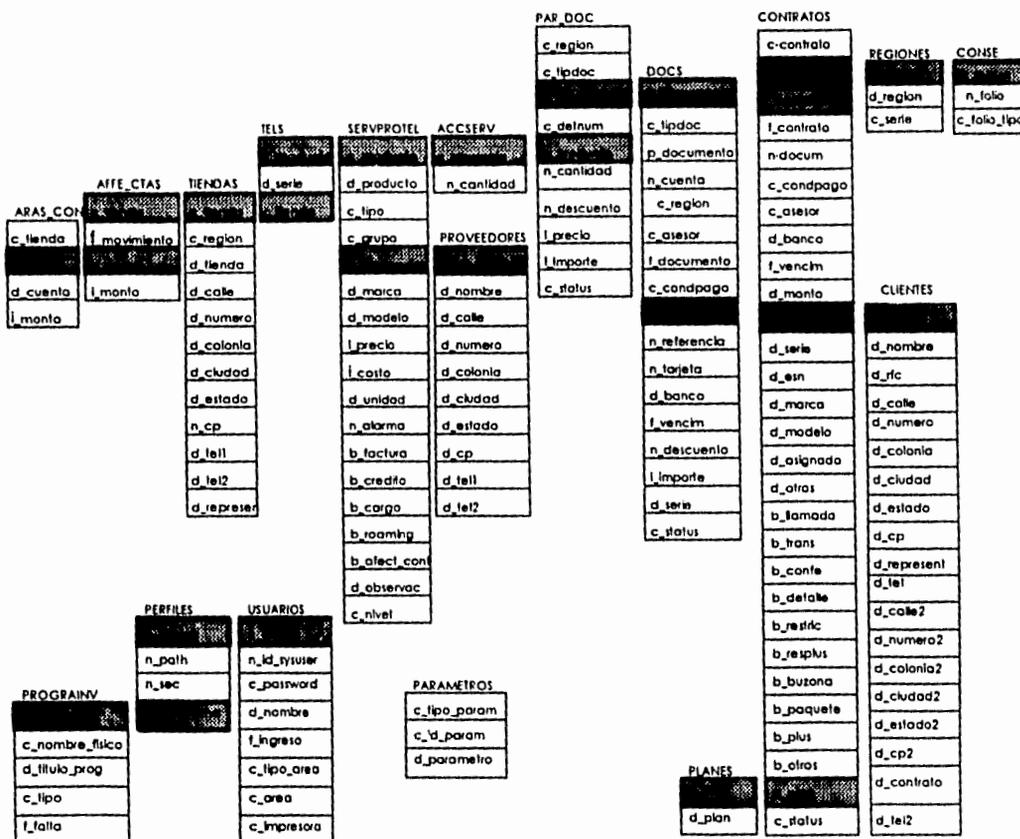


FIGURA 4.11.
DIAGRAMA E-R CUENTA CONTABLE-PRODUCTOS Y SERVICIOS

DIAGRAMA DE RELACIÓN DE TABLAS

El diagrama de relación de tablas es muy útil para apreciar la manera en que se pueden relacionar éstas, pues ellas conforman la base de datos con que trabajará el sistema.

Como se puede observar se encuentran sombreados los campos que servirán para relacionar todas las tablas de la base de datos; aunque esta no es la única forma de relacionarlas, pues algunas de ellas tienen otros campos en común por medio de los que se logrará otra relación; tal es el caso de la tabla DOCS con las tablas PAR_DOC y CONTRATOS que se pueden relacionar por el campo c_región común a las tres.



Además, se puede observar que se formaron tres grupos de tablas: el primero, que agrupa a la mayoría, está formado por las tablas que mantendrán toda la información relacionada con las operaciones de venta de la empresa (Contratos, Facturas, Notas de Crédito y Notas de Cargo, entre otros). El segundo grupo lo forman las tablas que servirán para mantener información sobre los usuarios y los programas que usará cada uno; es decir serán utilizadas para construir y guardar información de los perfiles de usuario. El tercer, y último, grupo lo forma sólo una tabla donde se tendrán parámetros de la empresa como: nombre del sistema, nombre de la empresa, etc.

DICCIONARIO DE DATOS

El diccionario de Datos es un listado organizado de todos los elementos que representan un dato que es pertinente al sistema. Este contendrá definiciones precisas acerca de los datos tales como: nombre del elemento, la descripción textual, el alias, los elementos relacionados de los datos, el rango, la longitud, el tipo, codificación y la información necesaria para su edición.

Una forma de iniciar el diccionario de datos es incluir a todos los elementos del diagrama de flujo de datos.

El diccionario de datos es útil en las fases de análisis, de diseño y en la documentación del sistema.

DICCIONARIO DE DATOS

NOMBRE DEL DATO	DESCRIPCIÓN
ARRASTRE CONTABLE	Información de la afectación a cuentas contables.
AUTORIZACIÓN	Visto bueno de atención a clientes para la elaboración de notas de crédito que se presentan en el área de contabilidad.
CANTIDAD	Cantidad de artículos de una determinada clase.
CLAVE	Clave de un nuevo artículo que se dará de alta en el catálogo.
CLAVE DE USUARIO	Password de seguridad asignado al usuario.
CLAVE DEL TELÉFONO O ACCESORIO	Número de Serie del Teléfono, Cantidad en existencia.
CONCEPTO	Descripción del bien o servicio incluido en una factura.
CONTRATO	Documento que relaciona al cliente con la empresa al adquirir un bien o servicio en términos de compromiso económico.
CONTRATOS ASESOR	Número de Contratos elaborados por asesor.
DATOS ASESOR	Nombre, clave, tienda del asesor, datos de relación de una factura con el asesor.
DATOS CONTRATO	Datos generales de contrato.
DATOS DE SOLICITUD	Descripción del bien y/o servicio solicitado.
DATOS DE ABONADOS	Datos generales de los clientes a abonados.
DATOS DE ACTIVIDADES DE USUARIO	Claves de usuario, área, tienda donde labora, puesto. ALIAS: Datos de usuario.
DATOS DE NOTAS DE CARGO	Datos necesarios para la elaboración de una nota de cargo ó factura relacionada, descripción, monto, clave del cliente.
DATOS DE NOTAS DE CRÉDITO	Datos necesarios para la elaboración de una nota de crédito: factura relacionada, descripción, monto, clave del cliente.
DATOS DE PRODUCTO EN INVENTARIO	Información de los productos en almacén.
DATOS DE PROGRAMA	Nombre del programa, tipo de programa, descripción, fecha de alta.
DATOS DE VENTA POR ASESOR	Información de movimientos comerciales realizados por los asesores: fecha de venta, monto, número de artículos, cliente. ALIAS: Ventas por asesor.

DATOS DE VENTAS	Información de transacciones de Venta: montos, fecha, Núm. de artículos vendidos, etc.
DATOS DEL CATALOGO DE USUARIOS	Información relativa a los usuarios del sistema SIFACOR: clave, nombre, actividad, etc.
DATOS DEL CLIENTE	Nombre, dirección, RFC, teléfono, clave del cliente.
DATOS FACTURA	Información requerida para elaboración de factura como: nombre del cliente, bienes adquiridos, monto, plan tarifario, descripción, etc.
DATOS PLAN O SERVICIO	Se refiere a los planes tarifarios de TELCEL, ofertas de servicio específicos conformados por distintos productos o servicios.
DATOS PRODUCTOS Y SERVICIOS	Nombre, descripción, clave y precio de un bien o servicio. ALIAS: Datos Producto, Datos de Producto.
DESCRIPCIÓN	Descripción de un artículo que se dará de alta en el catálogo de productos.
DOCUMENTOS	Factura, contrato.
EXISTENCIAS PRODUCTOS	Número de artículos en existencia en almacén de un determinado bien.
FACTURA	Documento con datos relativos a la transacción comercial efectuada.
FACTURACIÓN ASESORES	Montos de Venta conseguidos por un asesor.
FACTURACIÓN DIARIA	Monto de lo facturado diariamente en cada tienda.
FOLIO DE DOCUMENTOS	Rango de folios asignados por tienda.
INFORMACIÓN DE CATALOGOS	Información consultada por asesores de los catálogos del sistema: productos, contratos, facturas, clientes.
INFORMACIÓN DE FACTURACIÓN	Toda la información relacionada con los procesos de venta alimentada en SIFACOR.
INFORMACIÓN DE PROGRAMAS Y USUARIOS	Información inherente a los Programas (nombre, tipo, fecha de alta, número, etc.), y Usuarios (clave, número de identificación del usuario, password, nombre, tipo de área a la que pertenece, clave de la tienda a la que pertenece).
INFORMACIÓN DE TRANSACCIONES	Toda la información relacionada con la ventas y procesos involucrados con ellas.
MÍNIMO	Bandera que indica que se ha alcanzado el punto de reorden de un determinado producto.
MONTO	Importe por el que fue elaborada una factura.
MONTO DE CRÉDITO Y CARGO	Cantidades por concepto de crédito y/o cargos al cliente.
NOTA DE CARGO	Documento con el importe y descripción del cargo.
NOTA DE CRÉDITO	Documento con el importe y descripción del crédito.
NOTIFICACIÓN DE APROBACIÓN	Notificación de que los documentos o productos han sido verificados para la posterior autorización de un cambio o reembolso.
NUEVOS REQUERIMIENTOS	Peticiones del usuario para modificar al sistema de información.
NUM. DE CUENTA	Cuenta contable donde se llevará a cabo la afectación directa.
NUM. DE FACTURA	Folio de la factura.
NUM. DE SERIE	Número de serie del aparato telefónico (se captura a través del código de barras).
NUMERO DE DOCUMENTOS	Número de folio donde debe comenzar la numeración subsecuente.
PERFILES DE USUARIOS	Información para configurar las vistas por cada usuario: clave de usuario, actividad.

PRODUCTO	Bienes o servicios que adquiere el cliente.
PROGRAMAS	Es el código de los programas que se integran al sistema. ALIAS: Código Programa.
REPORTE EXISTENCIAS	Contiene la cantidad de productos de los artículos almacenados.
REPORTES Y DOCUMENTOS	Reportes y documentos que SIFACOR proporciona a entidades externas.
REQUISICIÓN DE PRODUCTO	Petición al almacén por un producto.
REPORTE ESTADÍSTICAS DE VENTA	ALIAS: Reporte de estadísticas.
SOLICITUD	Requisición de bienes al proveedor.

DICCIONARIO DE ALMACENES DE DATOS

NOMBRE DEL ALMACÉN	DESCRIPCIÓN
ASESORES	<p>No se define un sólo almacén de datos, sino dos:</p> <p>1) USUARIOS. Catálogo de usuarios que tienen acceso al sistema. Cada usuario pertenece a un área, ya sea ventas, facturación y administración (V,F,A respectivamente). Llave primaria = número de identificación del usuario. Llave secundaria = clave del usuario + número de identificación del usuario.</p> <p>2) DOCUMENTOS . Este archivo registra la información en general de un documento, ya sea factura, nota de crédito, nota de cargo o facturas roaming (F, C, A, R respectivamente), así como el importe total generado. El pago se puede hacer en efectivo, con cheque, con tarjeta, a crédito o sin costo (E, C, T, R, S respectivamente).</p> <p>Llave primaria = clave de la región + tipo de documento + clave de documento. Llave secundaria=clave de la región. Llave terciaria=clave del cliente.</p>
CATALOGO	<p>Es el almacén de teléfonos y accesorios en existencia en inventario. En el caso de teléfonos la cantidad en existencia está dada por el número de registros de archivo, ya que se registra el número de serie de cada teléfono, en qué tienda están dados de alta; al asignar el teléfono desaparece este registro.</p> <p>Llave primaria = clave del teléfono + número de serie del teléfono. En el caso de accesorios, contiene las existencias. Llave primaria = clave del accesorio.</p>
CATALOGO DE USUARIOS	<p>Catálogo de usuarios que tienen acceso al sistema. Cada usuario pertenece a un área, ya sea ventas, facturación y administración (V, F, A respectivamente).</p>

	<p>Llave primaria = número de identificación del usuario. Llave secundaria = clave del usuario + número de identificación del usuario.</p>
CLIENTES	<p>Catálogo que registra a todos los clientes a quienes se les ha generado algún tipo de documento. Llave primaria = clave del cliente.</p>
CONTRATOS	<p>Contiene información correspondiente a contratos (que se generan independiente de la factura).</p> <p>Llave primaria = clave de la región dueña del contrato + clave del contrato. Llave secundaria = clave del cliente que contrató la línea. Llave terciaria = número de serie del teléfono del cliente.</p>
CUENTAS	<p>Catálogo de las cuentas contables que son afectadas por las ventas aclaraciones de productos y servicios.</p> <p>Llave primaria = clave del grupo contable.</p>
FACTURAS	<p>Este archivo registra la información en general de un documento, ya sea factura, nota de crédito, nota de cargo o facturas roaming (F, C, A, R respectivamente), así como el importe total generado. El pago se puede hacer en efectivo, con cheque, con tarjeta, a crédito o sin costo (E, C, T, R, S respectivamente).</p> <p>Llave primaria = clave de la región + tipo de documento + clave de documento. Llave secundaria = clave de la región. Llave terciaria = clave del cliente.</p>
FOLIOS	<p>Catálogo de folios. Registra el número con el que se inicializa cada tipo de folio.</p> <p>Llave primaria = clave de la región a la que pertenece el folio.</p>
PLANES	<p>El archivo de planes contiene los tipos de planes tarifarios que se manejan en los contratos. Pueden ser a 30, 60 y 90 días.</p> <p>Llave primaria = clave del plan.</p>
PRODUCTOS Y SERVICIOS	<p>Catálogo que registra todos los productos y servicios de TELCEL. Puede haber teléfonos, accesorios y servicios (T, A, S respectivamente). Los servicios no tienen un número de existencia en inventario. Cada producto/servicio puede afectar al contrato, ya sea con una asignación ó una cancelación (A, C respectivamente).</p> <p>Llave primaria = clave del producto. Llave secundaria = clave del grupo contable al que pertenece. Llave terciaria = clave del proveedor del producto.</p>
PROGRAMAS	<p>Catálogo de todos los programas del sistema. Los programas pueden ser de tipo menú o ejecutables (M ó E respectivamente). Llave primaria = número de programa.</p>

DEFINICIÓN DE TABLAS

Los diagramas de entidad-relación anteriormente explicados en el capítulo, describen gráficamente a todos aquellos elementos que de una u otra forma participan principalmente en el proceso de emisión de *facturas*, *notas de crédito*, *notas de cargo* y *contratos*, además de todas las actividades que se llevan a cabo antes y después de generar estos documentos.

A partir de estas entidades y de las relaciones que surgen entre ellas, es posible determinar qué tipo de información habrá de guardarse en la BD de SIFACOR. El diseño de esta BD corresponde al de una de tipo relacional. Aunque no necesariamente una entidad corresponde a una tabla, es muy común que así sea. Sin embargo, es posible que se pueda determinar una tabla a partir de las dos entidades, y de la relación misma que existe entre ellas dentro de un diagrama. Los atributos representados en un diagrama de esta índole generalmente corresponden a las cualidades o características generales de algún elemento o entidad que participa en la operación del sistema, y casi siempre resulta necesario descomponer los atributos en elementos más pequeños que permitan información más detallada acerca de sí mismos. Con esta descomposición comúnmente se obtiene el suficiente grado de detalle de la información que habrá de administrarse en la BD. Siguiendo este procedimiento es como se logra determinar la estructura de las tablas que conforman la BD de la aplicación, Base de datos que llevará por nombre "*telaccser*" ya que corresponde a la facturación de *teléfonos*, *accesorios* y *servicios*.

A continuación se presentan las tablas con el nombre, estructura y breve descripción de los campos que las conforman:

ARRAS_CON			
Campo:	Tipo:	Nulos:	Descripción:
c_tienda	smallint	no	Clave de la tienda
c_grupo	char(5)	no	Clave del Grupo Contable
d_cuenta	varchar(14)	no	Nombre de la Cuenta
i_monto	decimal(12,2)	no	Monto acumulado de la Cuenta

REGIONES			
Campo:	Tipo:	Nulos:	Descripción:
c_region	smallint	no	Clave de la Región
d_region	varchar(30)	no	Descripción de la Región
c_serie	char(1)	no	Serie de la Región para folios de documentos

PAR_DOC			
Campo:	Tipo:	Nulos:	Descripción:
c_region	smallint	no	Clave de la Región de la Venta
c_tipodoc	char(1)	no	Tipo de Documento (F, C, A, R)
c_documento	integer	no	Clave del Documento
c_detnum	smallint	no	Clave del detalle del Documento
c_producto	char(6)	no	Clave del Producto

n_cantidad	smallint	no	Cantidad
n_descuento	decimal(4,2)	no	Descuento al Producto
i_precio	decimal(8,2)	no	Precio Unitario
i_importe	decimal(8,2)	no	Importe Total del Producto
c_status	char(1)	yes	

ACCSERV			
Campo:	Tipo:	Nulos:	Descripción:
c_producto	char(6)	no	Clave del Accesorio
n_cantidad	smallint	no	Cantidad en existencia

DOCS			
Campo:	Tipo:	Nulos:	Descripción:
c_region	smallint	no	Clave de la Región de la Venta
c_tipodoc	char(1)	no	Tipo de Documento (F, C, A, R)
p_documento	char(1)	si	Procedencia del Documento (Isis ó Facturación)
n_cuenta	char(14)	si	Clave de la cuenta Isis
c_documento	integer	no	Clave del Documento
c_asesor	integer	no	Clave del Asesor que realizó la venta
c_cliente	smallint	no	Clave del Cliente
f_documento	date	no	Fecha del Documento
n_referencia	integer	si	Número de Referencia
c_condpago	char(1)	no	Condición de Pago (E, C, T, R, S)
n_tarjeta	varchar(16)	si	Número de Tarjeta
d_banco	varchar(30)	si	Nombre del Banco
f_vencim	date	si	Fecha de Vencimiento de la Tarjeta
n_descuento	decimal(4,2)	no	Descuento Total del Documento
i_importe	decimal(8,2)	no	Importe Total del Documento
d_serie	char(20)	si	Número de Serie del Teléfono
c_status	char(1)	no	Status del Documento

SERVPROTEL			
Campo:	Tipo:	Nulos:	Descripción:
c_producto	char(6)	no	Clave del Producto
d_producto	varchar(30)	no	Descripción del Producto
c_tipo	char(1)	no	Tipo (T, A, S)
c_grupo	char(5)	no	Clave del grupo contable al que pertenecen
c_proveedor	smallint	si	Clave del proveedor del producto
d_marca	varchar(15)	si	Marca del producto
d_modelo	varchar(15)	si	Modelo del producto
i_precio	decimal(8,2)	si	Precio Unitario
i_costo	decimal(8,2)	si	Costo Unitario
d_unidad	char(5)	si	Unidad de medición
n_alarma	smallint	no	Punto de Reorden
b_factura	char(1)	no	Aplicable a Facturas (S/N)

b_crédito	char(1)	no	Aplicable a Notas de Crédito (S/N)
b_cargo	char(1)	no	Aplicable a Notas de Cargo (S/N)
b_roaming	char(1)	no	Aplicable a Roaming (S/N)
b_afect_cont	char(1)	no	Aplicable a Contrato (A, C)
d_observac	varchar(50)	si	Observaciones del producto

PERFILES			
Campo:	Tipo:	Nulos:	Descripción:
c_sysuser	char(5)	no	Clave del usuario (login con el que entra)
n_path	integer	no	Path donde se encuentra el programa habilitado
n_sec	smallint	no	Número secuencial
n_programa	smallint	no	Número de programa habilitado para ese usuario

TELS			
Campo:	Tipo:	Nulos:	Descripción:
c_producto	char(6)	no	Clave del Teléfono
d_serie	char(20)	no	Número de Serie del Teléfono
c_tienda	smallint	no	Clave de la Tienda en donde se encuentra el Teléfono

AFFECTAS			
Campo:	Tipo:	Nulos:	Descripción:
c_tienda	smallint	no	Clave de la Tienda donde se realiza el movimiento
f_movimiento	date	no	Fecha donde se lleva a cabo el movimiento
c_grupo	char(5)	no	Clave del grupo contable
i_monto	decimal(12,2)	no	Monto de la cuenta

PLANES			
Campo:	Tipo:	Nulos:	Descripción:
c_plan	smallint	no	Clave del plan
d_plan	varchar(30,10)	no	Descripción del plan

CONSE			
Campo:	Tipo:	Nulos:	Descripción:
c_region	smallint	no	Clave de la región a la que pertenece el folio
c_folio_tipo	char(1)	no	Tipo de folio
n_folio	integer	no	Número de folio

TIENDAS			
Campo:	Tipo:	Nulos:	Descripción:
c_tienda	smallint	no	Clave de la tienda
c_region	smallint	no	Clave de la región a la que pertenece el folio
d_tienda	varchar(30)	no	Nombre de la tienda
d_calle	char(30)	si	Calle donde se encuentra
d_numero	varchar(10)	si	Número
d_colonia	varchar(20)		Colonia
d_ciudad	varchar(10)		Ciudad
d_estado	char(4)		Estado
n_cp	integer		Código Postal
d_tel1	char(12)		Teléfono 1
d_tel2	char(12)		Teléfono 2
d_represen	varchar(30)		Nombre del representante

PROVEEDORES			
Campo:	Tipo:	Nulos:	Descripción:
c_proveedor	smallint	no	Clave del proveedor
d_nombre	varchar(30)	no	Nombre del proveedor
d_calle	char(30)	si	Calle del proveedor
d_numero	varchar(10)	si	Número
d_colonia	varchar(20)	si	Colonia
d_ciudad	varchar(10)	si	Ciudad
d_estado	char(4)	si	Estado
d_cp	integer	si	Código Postal
d_tel1	char(12)	si	Teléfono 1
d_tel2	char(12)	si	Teléfono 2

USUARIOS			
Campo:	Tipo:	Nulos:	Descripción:
c_sysuser	char(5)	no	Clave del usuario (login con el que entra)
n_id_sysuser	serial	no	Número de identificación del usuario
c_password	char(5)	no	Password del usuario
d_nombre	char(30)	no	Nombre del usuario
f_ingreso	date	no	Fecha de ingreso del usuario
c_tipo_area	char(1)	no	Tipo de área a que pertenece (V, F, A)
c_area	char(8)	no	Clave de la tienda a la que pertenece
c_impresora	char(20)	no	Comando para Impresora

CLIENTES			
Campo:	Tipo:	Nulos:	Descripción:
c_cliente	smallint	no	Clave del Cliente
d_nombre	varchar(30)	si	Nombre del Cliente

d_rfc	char(14)	si	Registro Federal de Causantes
d_calle	char(30)	si	Calle del Cliente
d_numero	varchar(10)	si	Número
d_colonia	varchar(20)	si	Colonia
d_ciudad	varchar(10)	si	Ciudad
d_estado	char(4)	si	Estado
d_cp	integer	si	Código Postal
d_represent	varchar(30)	si	Nombre del Representante
d_tel	char(12)	si	Teléfono 1 del Cliente
d_calle2	char(30)	si	Calle para Correspondencia
d_numero2	varchar(10)	si	Número
d_colonia2	varchar(20)	si	Colonia
d_ciudad2	varchar(10)	si	Ciudad
d_estado2	char(4)	si	Estado
d_cp2	integer	si	Código Postal
d_contacto	varchar(30)	si	Contacto del Cliente
d_tel2	char(12)	si	Teléfono 2 del Cliente

<i>PROGRAMAS</i>			
Campo:	Tipo:	Nulos:	Descripción:
n_programa	smallint	no	Número de programa
c_nombre_fisico	char(8)	no	Nombre físico del programa
d_titulo_prog	char(30)	no	Título del programa
c_tipo	char(1)	no	Tipo de programa (M, E)
f_alta	date	no	Fecha de alta del programa

<i>PARAMETROS</i>			
Campo:	Tipo:	Nulos:	Descripción:
c_tipo_param	char(3)	no	Clave del parámetro
c_id_param	char(10)	si	Identificador del parámetro
d_parametro	char(60)	no	Descripción del parámetro

<i>CONTRATOS</i>			
Campo:	Tipo:	Nulos:	Descripción:
c_region	smallint	no	Clave de la Región dueña del Contrato
c_contrato	integer	no	Clave del Contrato
c_documento	integer	no	Clave de la Factura con Asignación de Línea
c_cliente	smallint	no	Clave del Cliente que contrató la línea
f_contrato	date	no	Fecha del contrato
c_condpago	char(1)	si	Condiciones de pago
n_docum	varchar(16,0)	si	No. de documento (Tarjeta ó cheque).
d_banco	varchar(30,0)	si	Nombre del banco
f_vencim	date	si	Fecha de vencimiento en caso de tarjeta
d_monto	decimal(12,2)	si	Monto en caso de cheque

c_asesor	integer	no	Clave del Asesor que realizó la Venta
d_serie	char(20)	no	Número de Serie del Teléfono del Cliente
d_esn	char(20)	si	Número de Serie Electrónico (E.S.N.)
d_marca	varchar(15,0)	si	Nombre de la marca del teléfono
d_modelo	varchar(15,0)	si	Nombre del modelo del teléfono
c_plan	smallint	no	Nombre del plan contratado
d_asignado	varchar(12)	no	Número Telefónico asignado
b_llamada	char(1)	si	Contrata llamada en espera (S/N)
b_trans	char(1)	si	Contrata transferencia de llamada (S/N)
b_confe	char(1)	si	Contrata conferencia tripartita (S/N)
b_detalle	char(1)	si	Contrata detalle de llamadas (S/N)
b_restric	char(1)	si	Contrata restricciones (S/N)
b_resplus	char(1)	si	Rescatel plus (S/N)
b_buzona	char(1)	si	Contrata buzón (A,B ó U)
b_paquete	char(1)	si	Contrata paquete (S/N)
b_plus	char(1)	si	Contrata paquete Plus (S/N)
b_otros	char(1)	si	Contrata otros servicios (S/N)
d_otros	varchar(30)	si	Descripción de los otros servicios
c_status	char(1)	si	Status del contrato (A, C)

4.4. SEGURIDAD DEL SISTEMA

Dentro de las principales consideraciones que deben tomarse para el diseño de una aplicación es la integridad de su información. De nada serviría todo el esfuerzo de planeación, análisis, diseño y desarrollo de una aplicación si la vida del sistema resulta muy corta debido a continuas fallas, poca integridad y pérdida de información. En esta línea de ideas es que surge el concepto de *perfiles*. Contar con un *perfil* o con una versión personalizada de la aplicación permite de una manera muy segura canalizar el acceso de los usuarios hacia los programas que deben explotar, y restringirlos de aquellos con las cuales no deben interactuar. Es entonces el administrador de sistemas quien "arma" o compone el *perfil* de cada usuario según sus actividades.

Un elemento adicional a la seguridad de este sistema es el manejo de *passwords* por usuario; así solo el usuario y el administrador del sistema conocen aquella clave que permitirá el acceso del usuario autorizado hacia el perfil que tiene asignado.

Es mediante estos conceptos que se pretende robustecer la integridad de la información del SIFACOR. "A cada cual, sólo lo que le corresponde".

Como una nota adicional cabe mencionar que independientemente de la aplicación cada usuario posee también una cuenta personal de acceso al equipo, obviamente a nivel sistema operativo UNIX, que si se viene a ver representa un triple control de acceso a la información: password a nivel UNIX, password a nivel aplicación y perfil restringido de programas.

4.5 DISEÑO DE PANTALLAS, CONSULTAS Y REPORTES

La calidad de la información a la salida del sistema está determinada por la calidad del ingreso de información. Un buen diseño de pantallas debe satisfacer los siguientes objetivos:

- **Eficacia.** Que significa que las pantallas satisfagan objetivos específicos para el buen desempeño del sistema.
- **Precisión.** Que se refiere a un diseño tal que asegure una captura satisfactoria.
- **Facilidad de uso.** Que implica que las pantallas sean explícitas y no requieran tiempo adicional para descifrarse.
- **Consistencia.** Que significa que las pantallas deben permitir la captura o el despliegado de datos de una manera similar en toda la aplicación.
- **Sencillez.** Se refiere a mantener un mínimo de elementos indispensables que centren la atención del usuario.
- **Atracción.** Implica que el usuario disfrute del uso o tránsito a través de pantallas cuyos diseños deben de resultarle cómodos y atractivos.

Tomando en cuenta estos atributos se diseñarán las pantallas de la aplicación, la siguiente es una descripción de la distribución de las áreas básicas de las pantallas en general, así como un ejemplo de cómo se aplica esta distribución específicamente en las pantallas del programa.

En la figura 4.13. se pueden observar las tres secciones que son ya un estándar para ubicar la información en una pantalla. En la sección 1 deben aparecer títulos, encabezados de pantalla, opciones de menú y todo aquello que ofrezca datos sobre la intención de la pantalla. La sección 2 es la parte central de la pantalla, es en esta región donde aparecerán los campos de captura o donde se desplegará la información consultada. En la sección 3 debe aparecer toda indicación de uso de la pantalla: instrucciones, teclas especiales de función, ayudas, etc.

En la figura 4.13. se muestran las tres secciones básicas de una pantalla en general.

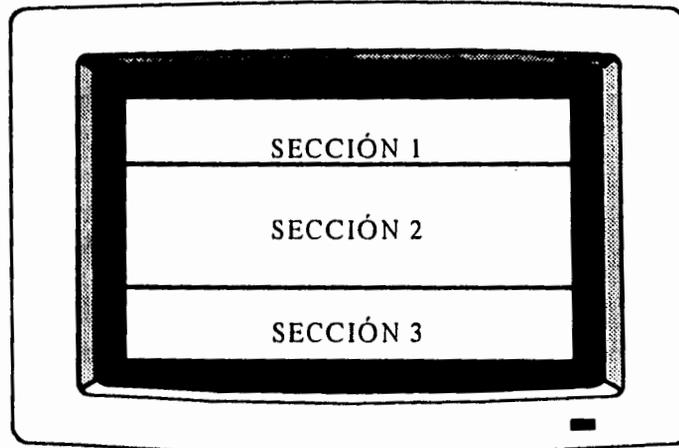


FIGURA 4.13.
SECCIONES BÁSICAS DE UNA PANTALLA

En la figura 4.14. se muestra como se aplican estas zonas para la pantalla de captura de un documento (factura, nota de crédito, nota de cargo) dentro de la aplicación.

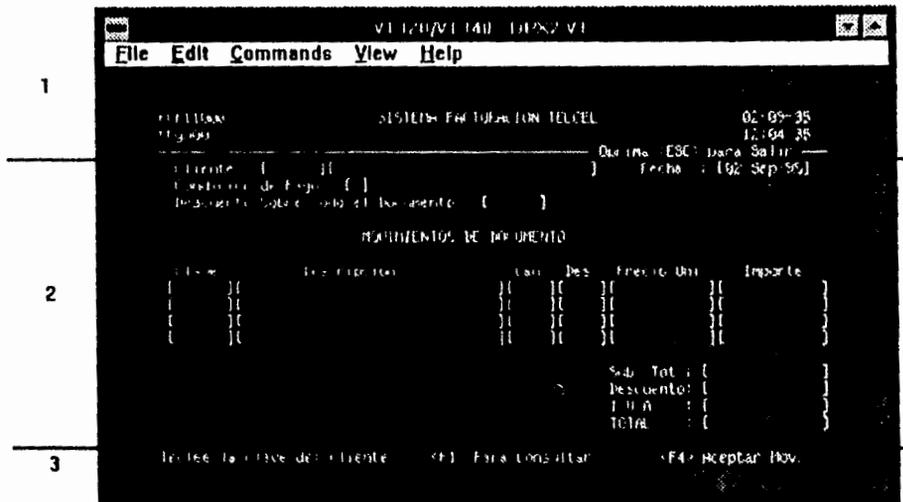


FIG. 4.14.
COMO SE APLICAN LAS SECCIONES BÁSICAS EN UNA PANTALLA DE LA APLICACIÓN

En la figura 4.14. se muestran las secciones y lo que en cada una de ellas debe aparecer: En la sección 1 aparece el título de la pantalla, así como información adicional: el nombre del programa de que se trata y el identificador de terminal (extremo izquierdo superior), la fecha y la hora (extremo derecho superior). En la sección 2 es donde se despliegan los campos de captura, ventanas de ayuda, de complemento de captura, etc. Finalmente en la sección 3 aparecen instrucciones y la descripción del uso de teclas especiales.

Otro tipo básico de pantalla resulta de un standard de manejo de *Informix* que provee mucha sencillez de programación para el manejo de menús de los llamados de "anillo", como se muestra en la figura 4.15., que corresponde al catálogo de clientes, por ende, todos los catálogos de la aplicación guardan este estilo de diseño.

En la figura 4.15. se puede observar la disposición lineal de las opciones del menú (Altas, Bajas, Cambios, Consultas y Reportes). Si se presiona alguna de las flechas de dirección sería posible observar un corrimiento del recuadro que resalta la opción en el mismo sentido de la flecha presionada. En caso de tratarse de la flecha izquierda (a partir de Altas) la siguiente opción que se resaltaría sería Reportes a la extrema derecha; de aquí el nombre de menús de anillo, ya que si la tecla se dejara oprimida indefinidamente se vería como las opciones se resaltarían una tras otra en un ciclo hasta que la tecla se libere.

Por otro lado, las consultas de SIFACOR heredan características específicas del lenguaje de programación utilizado, en este caso *Informix*, dado que provee mucha potencia en algo abiertos de búsqueda. Así si un asesor de venta requiere conocer los datos de un artículo en particular pero no recuerda la totalidad de la clave, basta que genere criterios de que se ha dado en llamar "**consultas dinámicas**".

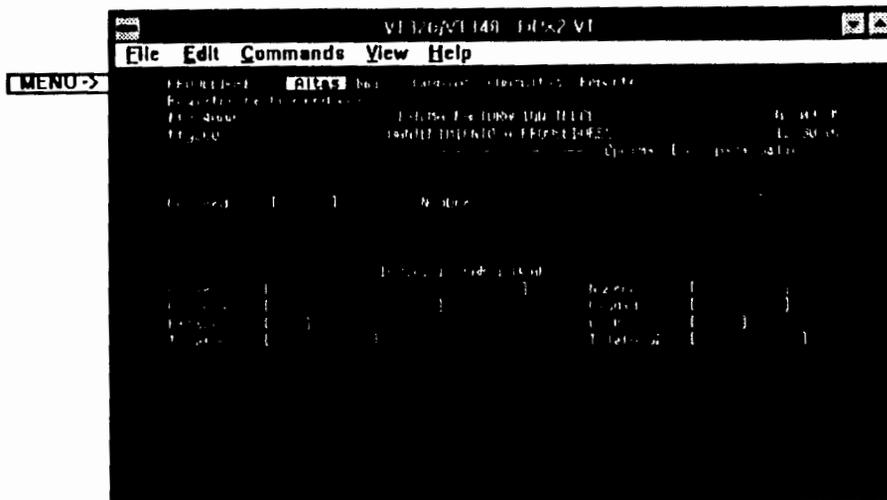


FIGURA 4.15.
PANTALLA DE CATÁLOGO CLÁSICO GENERADO EN INFORMIX.

Una consulta dinámica resulta de criterios de búsqueda como el siguiente:

Supongamos que la clave del artículo en cuestión es "VCCN012", pero el asesor sólo recuerda la primera parte: "VCC". Mediante una sencilla programación es posible crear un criterio de búsqueda que permita generar una ventana de ayuda en la cual se digite sólo la parte de la clave que se recuerda, seguida de un asterisco, como sigue: "VCC*", donde el asterisco funciona como un comodín que substituye cualquier cadena de caracteres y que por tanto lanza una instrucción que se podría interpretar más o menos como "Localiza todo aquello que comience con "VCC" y continúe con lo que sea". Así el asesor tendrá en su pantalla de consulta una lista de todos los artículos cuya clave comience con "VCC", él sólo tendrá que recorrer este pequeño listado para reconocer el artículo que le interesa.

El manejo de asteriscos permite insertar los comodines, al principio, al final, o a la mitad de la cadena de caracteres. Así, la búsqueda de claves, clientes, artículos, etc. puede ser muy ágil.

Otra característica importante de las consultas del SIFACOR es que se diseñan para una funcionalidad completa; es decir, dentro de cada opción de consulta, de cualquier catálogo, como el de la figura 4.15., es posible generar los criterios de búsqueda anteriormente descritos, sobre cualquier campo de la pantalla, de esta forma es posible que las consultas tengan como resultado despliegues por pantalla o por impresora de datos muy específicos relacionados con el género de información que albergue ese catálogo.

La potencia de los reportes que tendrá el SIFACOR prácticamente se ha descrito en el apartado anterior, ya que como se mencionó cualquier consulta puede convertirse en un reporte -entiéndase como reporte cualquier salida impresa- ó quedarse a nivel de consulta en pantalla. Sin embargo, los reportes o salidas impresas que generará el SIFACOR son de dos tipos y se determinan por su uso: el primero ya descrito se dirige a los usuarios del sistema, es

decir el personal de TELCEL que necesite imprimir información específica de la base de datos según su actividad. Y el segundo tipo de reportes se dirigirá a la clientela de TELCEL, y son básicamente los cuatro documentos mencionados en el análisis: Facturas, Notas de Crédito, Notas de Cargo y Contratos.

El primer tipo de reportes es un formato clásico que cuenta con una región de encabezados, donde aparece el nombre del reporte, el número de página, fecha, hora, etc. y el cuerpo del reporte que clasifica la información en columnas y renglones, como se ilustra en la figura 4.16.

FECHA:		PAG:
HORA:	TELCEL	
<p style="text-align: center;">REPORTE DE VENTAS POR ASESOR</p>		
COLUMNA1	COLUMNA2	COLUMNA-n
XXXXXXXXXX	XXXXXXXXXX	
	XXXXXXXXXX	

FIGURA 4.16.
REPORTE CONVENCIONAL

En la figura se muestra un reporte convencional, que clasifica la información a base de encabezados de columna y detalle a nivel de renglones.

El otro tipo de documentos que el SIFACOR deberá generar representa una de las principales encomiendas de este desarrollo ya que, como se mencionó, se trata de los documentos que se hará llegar a manos del cliente al momento de contratar un servicio y/o adquirir un bien en TELCEL.

Como se recordará, una condicionante es que estos documentos contarán con valor fiscal en caso de requerirlo, como es el caso de las Facturas, Notas de Crédito y Notas de Cargo. El cuarto tipo de documento que genera la aplicación es el Contrato y como tal, no requiere de características como la cédula de identificación fiscal ó el lugar de emisión del documento.

Para tal motivo la empresa decidió adquirir -en una valuación previa a este proyecto- cierto modelo de impresora XEROX que cuenta con un lenguaje propio de programación de formatos llamado "UDK" (*User Defined Key*), que funciona de manera similar a las secuencias de escape con que cuentan todas las impresoras, sólo que con mayor potencia en cuento al

meneo de "FONTS" (un FONT es algún tipo específico de carácter que una impresora puede representar en papel). UDK puede reconocer como FONTS segmentos de línea, uniones entre líneas y demás caracteres que permiten generar el formato del documento, como se muestra en la figura 4.17. Es por este medio por el cuál también se logrará la impresión de la cédula fiscal del documento, ya que la imagen del original será digitalizada ("escaneada") para posteriormente dividirla en partes más pequeñas; cada uno de estos fragmentos de la imagen se declara como un FONT, que programados en un orden correcto permiten "armar" como en un rompecabezas la imagen e imprimirla.

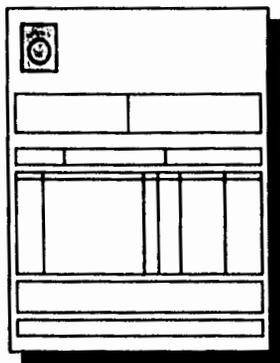


FIGURA 4.17.
FORMATO DE LAS FACTURAS, NOTAS DE CRÉDITO Y NOTAS DE CARGO.

La figura muestra el formato básico de distribución de la información de los documento mencionados, éste es el formato que deberá programarse en UDK, además de algunos encabezados fijos que determinan el uso de cada recuadro.

4.6 PSEUDOCÓDIGO

A continuación se incluye el pseudocódigo de cinco de los programas más representativos de la aplicación, a saber:

- **Catálogo de perfiles de usuario**
- **Catálogo de Clientes**
- **Inicializador de folios**
- **Reporte de ventas por asesor**
- **Elaboración de documentos**

PROGRAMA CATÁLOGO DE PERFILES DE USUARIO

Mantenimiento al archivo de vistas del sistema por usuario.
Altas, Bajas y Reportes con criterio de Búsqueda para generar reporte por usuario.

Programa Principal

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Obtiene parámetros de control y auditoría

Datos de: usuario, terminal, sistema, tienda, programa, impresora, área

Abre y despliega forma de mantenimiento a vistas

Despliega Menú de mantenimiento

1) Altas

Llama a la función **Altas**

2) Bajas

Llama a la función **Bajas**

3) Copia

Llama a función **Copiador**

4) Reportes

Llama a función **Reportes**

5) |

Llama a función **Comando**

Finaliza ciclo

Cierra forma de mantenimiento a vistas

Fin del Programa Principal

Función Altas

Da de alta un nuevo registro en la tabla de vistas

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Abre y despliega forma de alta de vistas

Pide clave de usuario

Valida clave de usuario en el catálogo de Usuarios

Pide el número de programa a dar de alta

Valida que el programa exista en el catálogo de Programas

Asigna nombre físico y tipo al programa elegido

Si pulsa tecla <F4>

Llama función **catálogo_lista**

Despliega datos del programa elegido

Llama a función **inserta_prog**

Inicializa campos en pantalla

Fin de condición si

Finaliza ciclo

Cierra ventana y forma de alta de vistas

Fin de Función Altas

Función Bajas

Da de baja un registro o programa de un menú de la tabla

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Abre y despliega una ventana con aviso de operación de las bajas: Borrar primero los ejecutables de un menú para luego borrar a éste pero vacío

Abre y despliega forma de Bajas de vistas

Pide clave de usuario

Valida clave de usuario en el catálogo de Usuarios

Llama a función **Borra_programa**

Fin de ciclo
Cierra ventana y forma de baja de vistas
Fin de Función Bajas

Función Copiador

Copia la vista final de un usuario a otro

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>
Abre y despliega forma de copia de vistas
Pregunta por la clave del usuario origen (de quien se copiará el perfil)
Valida que la clave de usuario origen exista en el catálogo de Usuarios
Valida que el usuario origen tenga definida una vista
Pregunta por clave de usuario destino (a quien se le copiará el perfil)
Valida que la clave de usuario destino exista en el catálogo de Usuarios
Valida que el usuario destino no tenga definida una vista
Si el usuario destino tiene una vista definida y se desea sustituirla entonces
Sale del ciclo
Fin de condición

Fin de ciclo
Llama a función **Copia_vista**
Despliega confirmación de copia hecha
Cierra forma de copia de perfil
Fin de Función Copiador

Función Valida_usu

Valida que exista la clave de usuario en el Catálogo de usuarios
Selecciona registros de SysUsers donde clave usuario coincida con la dada
Deposita resultados en buffer de Tabla de usuario
Fin de Función Valida_Usu

Función Cuenta

Verifica si el usuario tiene ya un perfil
Cuenta registros de Perfiles donde clave de usuario coincida con clave de usuario dada
Fin de Función Cuenta

Función Copia_Vista

Lee la vista de un usuario origen y se la copia al usuario destino
Borra de Perfiles los registros donde la clave de usuario coincida con la clave de usuario destino
Selecciona registros de Perfiles donde la clave de usuario coincida con la clave de usuario origen
Copia perfil de usuario origen en registro de usuario destino
Fin de Función Copia_Vista

Función Valida_pro

Valida que exista o no un registro en la tabla de programas
Selecciona registros de Inventario_de_Programas donde el número de programa coincida con el número de programa en pantalla
Fin de Función Valida_pro

Función Comando

Ejecuta un comando del sistema operativo sin salir de la aplicación

Limpia pantalla

Acepta comando

Ejecuta comando

Despliega mensaje 'Oprima <RETURN> para continuar'

Limpia pantalla

Fin de Función Comando

Función Inserta_prog

Despliega una vista en la ventana e inserta un nuevo programa en una vista dada con una variable que es el nivel de profundidad del programa en un perfil

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Llama a función **Despliega_vista**

Solicita la posición del programa en el menú

Llama a la función **Valida_programa** para determinar si el programa existe

Si el primer programa que se va a dar de alta en un perfil es un ejecutable, entonces despliega mensaje de error

Fin de condición

Si se desea dar de alta el programa un nivel abajo entonces

Asigna una profundidad más alta: profundidad*100

Llama a **Inserta_prog**

Fin de condición si

Inserta clave de usuario, profundidad, secuencia y número de programa en Perfiles

Llama a función **Display_vista**

Pide que se confirme el movimiento

Si el movimiento no se confirma entonces

Borra de Perfiles los registros donde clave de usuario, profundidad, secuencia y número de programa coincidan con los dados

Despliega mensaje de 'Movimiento abortado'

De otra manera

Despliega mensaje de 'Movimiento realizado'

Fin de condición

Fin de ciclo

Fin de Función Inserta_prog

Función Valida_programa

Valida a través de un ciclo si el programa que se desea insertar ya está dado de alta

Si el programa ya está dado de alta en alguna de las 12 posiciones en el perfil de usuario, entonces

Despliega mensaje de 'Error, ya está dado de alta'

Fin de condición

Fin de Función Valida_programa

Función Borra_prog

Despliega a través de una ventana la vista de un nivel dado y borra el programa seleccionado de la vista

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Llama a Display_vista

Solicita la posición del programa en el menú

Si existe una vista para el usuario dado entonces

Valida la existencia del programa seleccionado

Si se desea dar de baja un programa que es menú, entonces

Verifica si está vacío el menú

Si el menú está vacío, entonces

Llama a función **Borra_prog**

Fin de condición

Fin de condición

Llama a función **Display_vista** que muestra cómo quedó el nuevo menú

Pide que se confirme el movimiento

Si el movimiento no se confirma, entonces

Despliega mensaje de 'Movimiento abortado'

De otra manera

Borra de perfiles el registro donde la clave de usuario, profundidad, secuencia y número de programa coincidan con los datos

Despliega mensaje de 'Movimiento Realizado'

De otra manera

Despliega mensaje de 'No existe alguna vista para dar de baja'

Fin de condición

Fin de ciclo

Fin de Función Borra_prog

Función Display_Vista

Despliega a través de una ventana la vista de un nivel

Despliega letreros 'Nivel secuencia No. Prog. Nombre físico Título Tipo'

Lee cada módulo al que tiene acceso según su profundidad y los despliega

Fin de Función Display_Vista

Función Valida_existencia

Valida si existen vistas para dar de baja

Fin de Función Valida_existencia

Función Menú _vacío

Valida si el menú seleccionado está vacío o no

Bandera de búsqueda es igual a verdadera

Selecciona registros de Perfiles donde la clave de usuario y la profundidad coincidan con los datos

Si no encontró ningún registro entonces

Devuelve bandera de búsqueda falsa

Fin de condición

Fin de Función Menú _vacío

Función Reportes

Inicializa el reporte en pantalla, impresora o archivo

Comienza un ciclo hasta que el asesor pulsa la tecla <ESC>

Abre y despliega forma de Reportes

Pregunta el destino del reporte: <P>antalla o <I>mpresora

En caso de opción seleccionada

p: Envía el reporte a pantalla

l: Inicializa el reporte por impresora

De otra manera: Despliega mensaje: 'Opción Inválida'

Fin de En caso

Construye criterio de selección de registros a ser desplegados desde Perfiles

Si oprime tecla <F1>, entonces

Despliega catálogo de inventario de programas, número de programa, título y tipo

Guarda en memoria los campos del registro seleccionado en pantalla

Incluye los campos seleccionados en el criterio de selección de registros

Fin de condición

Selecciona los registros en base al criterio dado

Lee los registros seleccionados e Imprime o despliega el reporte

Cambia el privilegio del archivo seleccionado en el reporte para que todos tengan acceso a la información

Fin de ciclo

Fin de Función Reportes

Formato de Salida del Reporte

Título: Reporte de Perfiles

Entradas: Registros seleccionados de perfiles

Encabezado: Identificador de programa, nombre del empleado, Identificador de oficina, Nombre del usuario final, hora, número de página

Campos desplegados: Clave de usuario, Profundidad o path, Programa, Tipo, Título del Programa

Pie de Página: Total de programas tipo Menú, Total de programas Ejecutables, Total de Programas por Usuario, Total de Registros Leídos.

PROGRAMA CATALOGO DE CLIENTES

PROGRAMA DE MENÚ CLIENTES

PROGRAMA PRINCIPAL

Limpia pantalla

Muestra forma del menú clientes

MENÚ " CLIENTES"

- 1) Opción " Altas Cliente "
- 2) Opción " Baja Cliente "
- 3) Opción " Cambios a Clientes "
- 4) Opción " Consulta a Clientes "
- 5) Opción " Reporte de Clientes "

En el CASO de

opción = "1"
Llama a la Función "altas clientes"
opción = "2"
Llama a la Función "bajas clientes"
opción = "3"
Llama a la Función "cambios clientes"
opción = "4"
Llama a la Función "consulta clientes"
opción = "5"
Llama a la Función "imprime reporte"

Fin del CASO
Cierra la forma
Fin de PROGRAMA PRINCIPAL.

FUNCIÓN altas clientes

Limpia pantalla
Abre la forma para captura los datos de alt_cliente
Muestra la Forma

Mientras sea Verdadero

Inicializa las variables del registro cliente
Captura los datos del cliente según la forma

Si se capturó el nombre del cliente entonces
se podrá capturar el teléfono 2 del cliente
válida si se oprime la tecla esc
Fin de condición

Una vez capturado el Registro Federal de Contribuyentes (RFC)

S el RFC capturado es igual al RFC de la tabla Clientes entonces
manda mensaje de que el cliente ya existe
regresa la clave del cliente y el nombre del cliente
Fin de condición

Después de capturar nombre del representante
Si nombre del representante capturado es =nombre del cliente entonces
copia todos los valores del cliente en los campos del representante
de lo contrario
inicializa variables para datos del representante
muestra los datos del representante
Fin de condición

valida si se interrumpe el alta
cierra la Forma

regresa la clave del cliente y su nombre

salida del mientras

Inserta los datos en la Tabla

Fin del mientras

borra los datos de la pantalla.

Fin de la Función alta clientes

FUNCIÓN baja clientes

Limpia pantalla

Inicializa las variables del registro clientes

Limpia pantalla

Captura la clave del cliente

valida si se oprime ESC

termina la captura.

Si no se tiene la clave del cliente entonces

Llama lista el catálogo por nombre y clave del cliente

Fin de condición

Si la clave del cliente = 0 entonces

despliega mensaje: cliente ISIS no puede ser dado de baja

Fin de condición

Si no se valida la clave del cliente entonces

manda mensaje de que el cliente no existe

inicializa las variables del registro cliente

borra forma

Fin de condición

despliega el registro del cliente en la pantalla

Confirma si se desea dar dar de baja (S/N)

Si la respuesta fue si entonces

se confirma la baja

Si la clave del cliente de la tabla "Documentos" es = clave del cliente del registro

cliente y/o la clave del cliente de la Tabla "Contratos" es = la clave del cliente del

registro cliente entonces

manda mensaje de que el cliente no se puede dar de baja puesto que existen Dctos.

y/o Contratos.

Fin de condición

Fin de condición

Borra los datos de la pantalla

Limpia pantalla

Fin de la FUNCIÓN baja clientes

FUNCIÓN cambios clientes

Limpia pantalla
Inicializa las variables del registro clientes
Limpia la Forma
Captura la clave del cliente a modificar
valida si se oprime esc
termina la captura

Si no se cuenta con la clave del cliente entonces
muestra Lista Catálogo
muestra por clave del cliente y nombre del mismo
Fin de condición

Después de capturar la clave del cliente
Si no valida la clave del cliente entonces
manda mensaje de que el cliente no existe
Inicializa las variables del registro clientes
Limpia forma
Próxima clave a capturar
Fin de condición
muestra todos los campos del registro clientes

Si se captura el nombre del cliente entonces
en seguida permitirá capturar teléfono 2 del cliente
Fin de condición

Después de capturado el campo nombre del representante
Si nombre del representante =nombre del cliente entonces
carga los valores de clientes en los del representante
De lo Contrario
inicializa las variables de representante
muestra los datos del representante
Fin de condición

Actualiza la tabla con los datos cambiados
Limpia la forma
Manda mensaje de que el registro a sido modificado
Fin de la FUNCIÓN cambios clientes

FUNCIÓN consulta clientes

Limpia pantalla
valida si se oprime esc
valida si se interrumpe el reporte
Borra los datos de la forma
Limpia pantalla

Captura clave del cliente

Si clave del cliente no es encontrado entonces
 manda mensaje de que "cliente No Localizado"
 Borra forma
 Limpia pantalla
 regresa al menú consultas
de lo contrario
 muestra la información del catálogo
Fin de condición
borra los datos de la forma
Limpia pantalla
Fin de la FUNCIÓN consulta clientes

PROGRAMA INICIALIZADOR DE FOLIOS

Mantenimiento al catálogo de folios, con la condición de que sólo pueden ser modificados los datos de los números de folio existentes en la base, si los números de folio fueron dados de baja fuera del sistema de facturación, este programa los crea en forma automática.

Programa principal

Siempre que ocurra un error
 llama al manejador de errores
Se define la tecla F4 para aceptación
Se suspenden interrupciones
Se suspende salida del sistema
Se abre forma y la despliega en pantalla
 -llama función que despliega forma, fecha y hora del sistema en pantalla e identificadores de terminal y programa
 -llama la función alta_folios que registra los diferentes números de folio para cada una de las regiones existentes
Cierra la forma
Fin del programa principal

Función alta_folios

Se definen los arreglos de 5 elementos
 de tipo de folio a seleccionar
 de número de folio a seleccionar
siempre que ocurra un error
 llama al manejador de errores
se teclea la opción
 se limpia y despliega mensaje para indicar el tipo de movimiento hecho
verifica la existencia de los números de folio
se inicia el arreglo de tipo de folio a seleccionar
 tipo de folio1=contratos
 tipo de folio2=romming
 tipo de folio3=notas de crédito
 tipo de folio4=notas de cargo

tipo de folio5=facturación
 para cada apuntador a la tabla regiones.clave de región
 de 1 a 5 *****se llenan cada uno de los 5 folios, puesto que por cada
 región se da inicio a 5 folios
 selecciona el campo numero de folio
 de folio
 donde la clave de la región=clave de la región de la tabla regiones, y
 tipo de folio=tipo de folio seleccionado
 si no se encuentra
 insertar en folios los valores de
 clave de la región y,
 tipo de folio seleccionado de la tabla
 regiones
 fin
 siguiente
 fin de para cada
 mientras sea verdadero
 inicializa los registros de la tabla regiones a utilizar a valores nulos
 se captura el registro a modificar de la tabla regiones, y valida si se oprime "esc"
 llama a la lista de catálogos y regresa los valores de
 clave de la región y,
 descripción de la región
 despliega los campos de la tabla regiones para modificarlos
 valida si se encuentra la clave de la región
 si no existe despliega el mensaje "no existe la región"
 si existe, despliega el campo descripción de la región
 valida el cambio si se interrumpe, y salir del mientras
 las 5 variables de número de folio seleccionado se inicializan a cero
 de 1 a 5
 selecciona un número de folio de la tabla folios
 donde la clave de la región=clave de la región de la tabla regiones
 seleccionada y, su tipo de folio=variable tipo de folio
 siguiente
 igualar los registros de cada número de folio=a cada número de folio a seleccionar
 introducir cada uno de éstos números de folio a los registro de cada número de folio
 de 1 a 5
 actualizar los folios
 sea número de folio=número de folio de variable seleccionada y, donde clave
 de la región=clave de la región de la tabla seleccionada y, donde tipo de folio=a
 variable tipo de folio seleccionado
 siguiente
 borra los datos de la forma
 despliega el mensaje "registro modificado"
 fin de mientras sea verdadero
 borra los datos de la forma
 limpia mensajes de la pantalla
Fin de la función alta folios

PROGRAMA. ELABORACIÓN DE DOCUMENTOS

Comienza ciclo hasta que el asesor pulse la tecla <Esc>

Abre la forma de captura de documentos

Despliega la forma de captura de documentos

Si el asesor no conoce la clave del cliente entonces

Llama a la ventana de consulta

Si en la ventana de consulta no aparece el cliente entonces

Regresa a la forma de captura de documentos

Pregunta si desea dar de alta al cliente

Si desea darlo de alta entonces

Llama a la función de alta de clientes

Fin de condición

Fin de condición

De lo contrario

Trae la clave y la descripción del cliente

Fin de Condición

Acepta la clave del cliente y busca la descripción

Si no aparece entonces

Pregunta si desea darlo de alta

Si desea darlo de alta entonces

Llama a la función de alta de clientes

De lo contrario

Cierra la forma de captura de documentos

Sal del programa

Fin de condición

Fin de condición

**Pregunta por la condición de pago : E=Efectivo, C=Cheque, T =Tarjeta, R= Crédito,
S=Sin costo**

Si la condición de pago es T

Llama a la función de registro de tarjeta

Fin de condición

Casos

Cuando el documento sea Factura

Pregunta por el descuento que se hará por el total del documento

Comienza un ciclo hasta que el asesor termine captura de artículos

Pregunta por la clave del artículo

Si el asesor no la conoce entonces

Llama a la ventana de consulta

Trae la clave y la descripción

De lo contrario

Acepta la clave que el asesor digite y búscala en el catálogo

Si la encuentras entonces
 Despliega la descripción del artículo
 Si la clave corresponde a un teléfono entonces
 llama a la función de desplegar números de serie
 Fin de condición
De lo contrario
 Manda error
Fin de condición
Fin de condición

Acepta la cantidad de artículos que capture el asesor
Acepta el descuento por producto que capture el asesor
Despliega el precio unitario asociado al artículo
Despliega el importe que resulte de multiplicar el precio unitario por la cantidad de artículos

Termina ciclo

Cuando el documento sea una nota de crédito

Llama a la forma de de captura de referencias-1

Pregunta por la procedencia del documento

Si la procedencia es Factura entonces
 Pregunta por el folio de la factura de referencia
 Si no la encuentras entonces
 Despliega mensaje de error
 Regresa a preguntar por la Factura de referencia
 Fin de condición

De lo contrario

 Pregunta por la clave de la cuenta en Isis

Fin de ciclo

Pregunta por el iva a aplicar

Cierra la forma

Busca la factura y trae la información

Despliega la información correspondiente en los campos de la forma

Espera a que el usuario confirme el movimiento

Cuando el documento sea una nota de cargo

Llama a la forma de captura de referencias-2

Pregunta por la procedencia del documento

Si el documento es Factura o Nota de crédito entonces

 Pregunta por el folio de la Factura o nota de crédito de referencia

 Busca en la BD la Factura o Nota de crédito de referencia

 Si no la encuentras entonces

 Despliega mensaje de error

 Regresa a preguntar por la factura de referencia

 Fin de condición

De lo contrario

 Pregunta por la clave de la cuenta en Isis

Fin de ciclo

Pregunta por el iva a aplicar

Cierra la forma
Busca la factura y trae la información
Despliega la información correspondiente en los campos de la forma
Espera a que el usuario confirme el movimiento
Fin de casos

Despliega la ventana de comentarios de la factura
Acepta los comentarios de la factura
Llama a la función de confirmar documentos
Si el documento se confirma entonces
 Mándalo a la cola de impresión que tenga asociada el usuario
De lo contrario
 Regresa al principio de la captura
Fin de condición
Llama a la función de salida o reimprime
Termina ciclo

PROGRAMA REPORTES DE VENTA POR ASESOR

Comienza ciclo hasta que el usuario pulse <Esc>

Abre la forma de captura
Despliega la forma de captura

Pregunta por el tipo de documento

Si el tipo de documento no se define entonces
 Incluye en la búsqueda a todos los documentos
De lo contrario
 Incluye en la búsqueda el documento específico
Fin de condición

Pregunta por la clave de la región

Si el la región no se define entonces
 Incluye en la búsqueda a todas las regiones
De lo contrario
 Incluye en la búsqueda la región específica
Fin de condición

Pregunta por la clave del asesor

Si la clave del asesor no se define entonces
 Incluye en la búsqueda a todos los asesores de esa región
De lo contrario
 Incluye en la búsqueda a el asesor específico
Fin de condición

Pregunta por la clave del documento (folio)

Si la clave del documento se define entonces
Incluye en la búsqueda a todos los documentos de ese asesor
De lo contrario
Incluye en la búsqueda el documento específico
Fin de condición

Pregunta por la fecha de elaboración del documento

Si la fecha no se define entonces
Incluye en la búsqueda a todas las fechas de los documentos de ese asesor
De lo contrario
Incluye en la búsqueda el documento específico
Fin de condición

Pregunta por la condición de pago

Si la condición de pago no se define entonces
Incluye en la búsqueda a todas las condiciones de los documentos generados por
ese asesor
De lo contrario
Incluye en la búsqueda la condición específica
Fin de condición

Con los anteriores criterios construye una búsqueda

Busca en la BD

Si no encuentras condición con tales criterios entonces
despliega mensaje de error
De lo contrario

Muestra como título el nombre de la región

Despliega el monto acumulado de la región

Despliega las opciones de detalle del siguiente nivel (asesores)

1. Detalle de asesores
2. Imprime el presente nivel

Si el usuario solicita mayor detalle entonces

Indica al usuario que marque la región de interés

Con este parámetro construye una búsqueda de todos los asesores de esa
región

Muestra como título en nombre de la región

Despliega la clave, el nombre y el monto de ventas de cada asesor

Despliega las opciones de detalle a siguiente nivel (facturas)

1. Detalle de facturas por asesor
2. Imprime este nivel

Si el usuario solicita mayor detalle entonces

Indica al usuario que marque el asesor de interés

Con este parámetro construye una búsqueda de todas las facturas de ese
asesor

Muestra como título el nombre de la región y el nombre del asesor

**Despliega como listado las claves de los documentos y sus montos
totales**

Despliega las opciones de detalle a siguiente nivel (partidas)
1. Detalle de las partidas de la factura
2. Imprime este nivel
Si el usuario solicita mayor nivel detalle entonces
Indica al usuario que marque el documento de interés
Con este parámetro construye una búsqueda de todas las partidas de esa factura
Muestra como título el nombre del asesor y la clave de la factura
Despliega como listado el número de partidas de la factura, la descripción del concepto y el monto
Muestra opción de impresión a este nivel
Fin de condición
Fin de condición
Fin de condición
Fin de condición
Fin de ciclo

CONCLUSIONES

De este capítulo podemos concluir que la fase de diseño es una de las más creativas en la etapa de vida del software.

Nos dimos cuenta de la estrecha relación que existe entre análisis y diseño. El análisis estructurado aportó una base sólida sobre la cual el diseño se llevó a cabo con menos dificultades.

El uso de herramientas en esta fase como los diagramas entidad-relación y el diccionario de datos permitieron visualizar con mayor facilidad las relaciones y atributos de los objetos que intervienen en el proceso de facturación.

La culminación de esta fase estuvo dada por el desarrollo del pseudocódigo de los programas que compondrán al SIFACOR, junto con la estructura de las tablas que lo apoyarán.

El paso siguiente es la implementación en lenguaje de cuarta generación de los programas aquí diseñados.

CAPÍTULO 5

DESARROLLO

A través del desarrollo se logra traducir el conjunto de requisitos y especificaciones del diseño en el elemento operativo del sistema que llamamos software, es decir, la función primordial del desarrollo es la codificación.

Basándose en las especificaciones de diseño, se lleva a cabo la generación de un programa en un lenguaje de programación apropiado.

Para el desarrollo del SIFACOR se utilizaron las herramientas de programación de INFORMIX, con las que ya contaba TELCEL.

Este capítulo repasará, en términos generales, las características más importantes de las herramientas de INFORMIX que fueron utilizadas durante el desarrollo de la aplicación.

5.1. LA HERRAMIENTA DE PROGRAMACIÓN INFORMIX

INFORMIX es un sistema manejador de base de datos relacionales desarrollado por INFORMIX Inc., que utiliza los estándares del lenguaje de datos SQL, desarrollado por IBM, para proporcionar enunciados de definición y manipulación de bases de datos relacionales.

Un sistema manejador de base de datos es un conjunto de programas que administran el uso eficiente de la información contenida en una base de datos y permite la interacción de ésta con el usuario. Entre sus funciones se encuentran:

- ⇒ Mantener la información disponible en el lugar y en el momento donde se necesite.
- ⇒ Mantener la consistencia e integridad de la información.
- ⇒ Manejar adecuadamente la concurrencia a la información.
- ⇒ Regular el acceso a la información de acuerdo a las seguridades establecidas.
- ⇒ Proporcionar interfases universales a la información.

GENERALIDADES

INFORMIX trabaja bajo dos ambientes de operación:

INFORMIX SE (STANDARD ENGINE) : Este ambiente de operación es el más simple, permite conectar las aplicaciones a las Bases de Datos, en el sentido de que cada una de ellas utiliza una copia del manejador, dejando todas las tareas administrativas de espacio, memoria y utilerías de respaldo al sistema operativo. Sólo se proporcionan servicios de la administración de la información.

INFORMIX ON-LINE: Este ambiente, al igual que el anterior, proporciona una copia de él a las aplicaciones de Bases de Datos, pero con la diferencia de que brinda servicios adicionales a la administración de la información. Todas las Bases de Datos, a pesar de ser diferentes, comparten un ambiente común. Se proporcionan servicios de administración de la información, administración de la memoria y control de concurrencia e integridad de la información.

ARQUITECTURA DE INFORMIX

INFORMIX utiliza la arquitectura cliente-servidor o también llamada de "Dos Procesos", ya que atiende por separado las tareas de la aplicación del usuario y las transacciones que tiene que llevar a cabo como servidor de Base de datos, por eso es posible resumir la gama de herramientas INFORMIX en dos géneros:

- Las herramientas para el desarrollo de aplicaciones que funcionan como interfases para construir y correr aplicaciones.
- Los servidores de Base de Datos que llevan a cabo toda la administración de la información, ya sea procesos de almacenamiento, recuperación, etc.

En la figura 5.1. se ilustra la filosofía de trabajo de la arquitectura de dos procesos. En ella se puede apreciar cómo el lenguaje SQL es la interfaz entre el usuario y el sistema manejador de bases de datos.

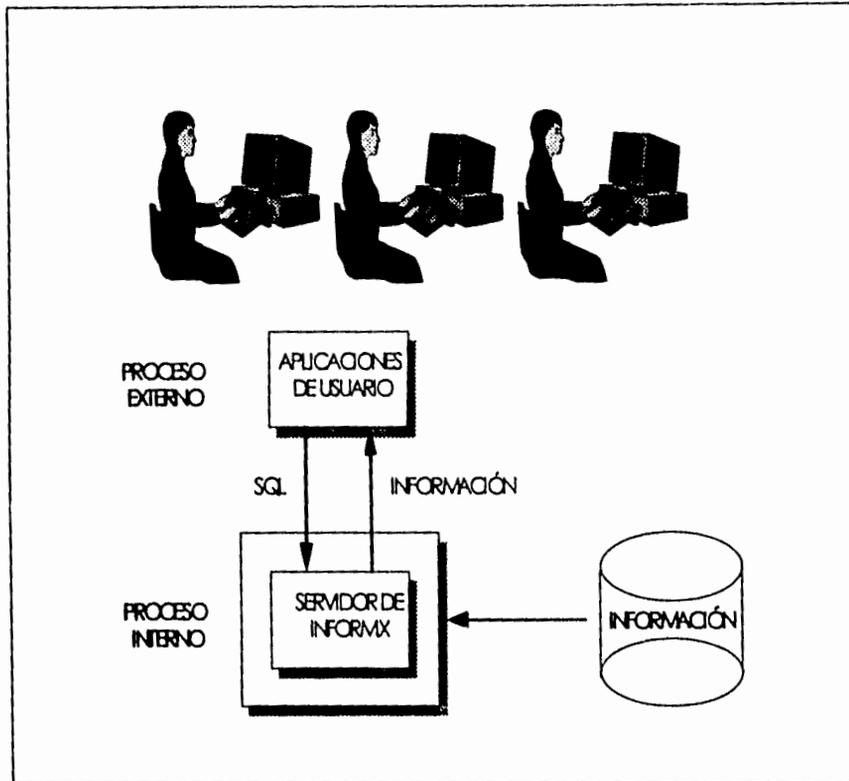


FIGURA 5.1.
ARQUITECTURA DE DOS PROCESOS

Otro aspecto interesante incluido en el funcionamiento de INFORMIX es que se apega a los estándares SQL para realizar todas las solicitudes de información desde la aplicación del usuario hacia el servidor, dicha estandarización permite que el usuario de otros manejadores de Bases de Datos puedan adaptarse fácilmente a INFORMIX, que con su arquitectura de dos procesos permite que usuarios y servidor residan en el mismo o en diferentes equipos en cuyo caso se crea una arquitectura cliente-servidor.

En la figura 5.2. se muestra cómo los productos INFORMIX se acoplan a la arquitectura de dos procesos o cliente - servidor misma que presenta varias ventajas:

- ⇒ **Desempeño.** Al separar los procesos que realiza directamente el usuario de los que tiene que llevar a cabo el servidor de Base de Datos, permite que el primero no tenga que detener su trabajo, mientras que el segundo lleva a cabo transacciones con la Base de Datos, por tanto, se puede optimizar el funcionamiento del hardware y software al dirigirse a funciones específicas.

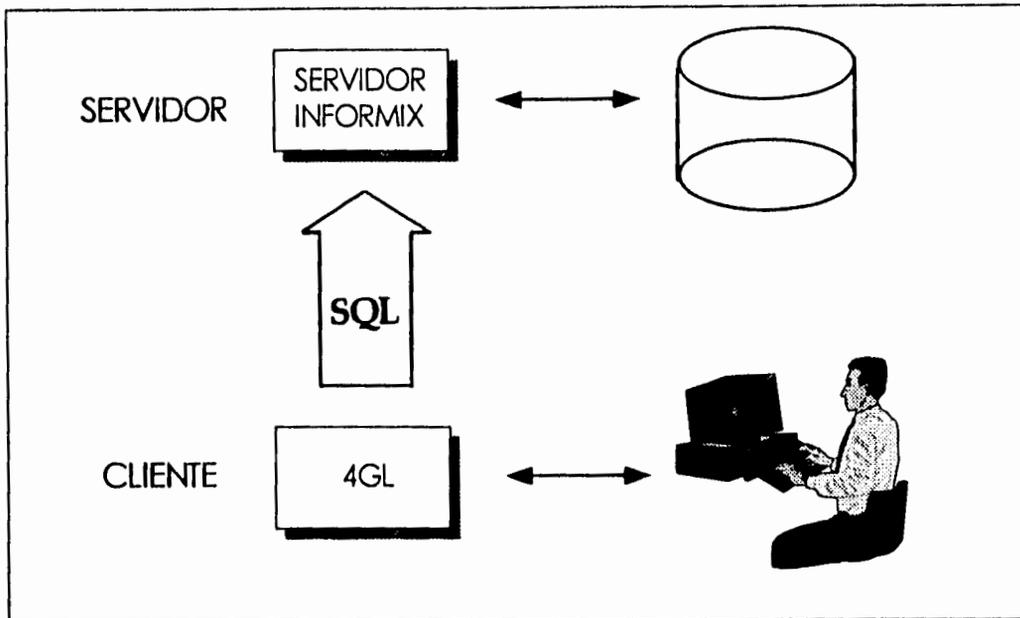


FIGURA 5.2.
INFORMIX-4GL Y ARQUITECTURA CLIENTE-SERVIDOR

- ⇨ **Flexibilidad.** Con la arquitectura de dos procesos, las herramientas de INFORMIX orientadas a los procesos del usuario pueden ser utilizadas por cualquier servidor de INFORMIX, permitiendo que el usuario elija el servidor que mejor se adapte a su entorno de hardware; a la larga este tipo de arquitectura abierta permitirá interactuar a las herramientas de proceso de usuario de INFORMIX con servidores de Base de Datos de distintos fabricantes.
- ⇨ **Reducción de Costos.** El hardware puede ser seleccionado según los requerimientos específicos de la aplicación para sacar el mayor provecho de la relación Precio-Desempeño.
- ⇨ **Portabilidad.** Las aplicaciones INFORMIX pueden transportarse fácilmente entre diferentes equipos.

INFORMIX 4GL Y LA ARQUITECTURA CLIENTE-SERVIDOR

INFORMIX provee un modo de interacción sencillo y seguro con la Base de Datos al mismo tiempo que la protege de los posibles errores del usuario final. Esta característica es posible dado que el código generado actualmente por INFORMIX es un híbrido de 4GL/SQL que contiene:

- ⇨ Lógica de 4GL para la captura y el diseño de la interfase de usuario.

- ◊ Comandos de SQL que son atendidos por el servidor de INFORMIX para ser procesados.

Además, INFORMIX posee un amplio rango de herramientas para el desarrollo de aplicaciones que se adecuan a los diferentes tipos de aplicaciones y de organizaciones.

HERRAMIENTAS

INFORMIX proporciona una línea completa de productos avanzados para el desarrollo e implementación de aplicaciones de administración de información de sistemas abiertos. Sus principales productos incluyen herramientas para el desarrollo de aplicaciones, servidores de bases de datos y software de conectividad. Todos los productos están basados en los estándares de SQL ANSI.

Las herramientas de INFORMIX pueden correr bajo la mayoría de ambientes operativos, incluyendo UNIX, DOS, Macintosh, OS/2, NeXT, etc. Brevemente describiremos las Herramientas de Aplicación, Productos OpenCase y Servidores de Bases de Datos.

A continuación se detallan las características más relevantes de éstas herramientas.

HERRAMIENTAS DE APLICACIÓN

El INFORMIX 4GL es un lenguaje de base de datos de cuarta generación basado en los estándares SQL. Se puede usar para crear aplicaciones complejas que resultan completamente adaptables, fáciles de modificar y con sólidos alcances.

INFORMIX-4GL Rapid Development System: Incorpora un procedimiento para compilar y correr programas INFORMIX-4GL que elimina la necesidad del compilador C y el ligador C, reduciendo de esta manera el tiempo de compilación. Las aplicaciones son completamente transportables entre INFORMIX-4GL Rapid Development System e INFORMIX-4GL.

INFORMIX-4GL Interactive Debugger: Trabaja en conjunción con el anterior, y permite al analista ver y probar el código INFORMIX-4GL en una pantalla dividida mientras corre una aplicación.

INFORMIX-4GL/RF: Permite al analista o programador desarrollar aplicaciones que se comunican en forma clara utilizando tecnología de radio frecuencia (RF) a computadoras a distancia.

INFORMIX-4GL/GX: Nos proporciona un medio gráfico en ambiente Windows para aplicaciones escritas en INFORMIX-4GL. Es compatible con todos los productos de INFORMIX-4GL.

INFORMIX-TP/ToolKit: Permite al programador en INFORMIX-4GL crear aplicaciones OLTP (On Line Transaction Processing) que corren en conjunción con el administrador de transacciones.

INFORMIX-4GL Forms: Es un generador de código y representador de formas que permite al analista generar la entrada de datos en forma óptima y rápida.

INFORMIX-Menus: Es una herramienta de desarrollo de sistemas de menú que permite a los programadores crear menús rápidamente en aplicaciones INFORMIX de bases de datos.

INFORMIX-SQL: Es un manejador de bases de datos relacionales basado en SQL, diseñado para programadores y no programadores. Provee herramientas para crear aplicaciones sin conocimientos avanzados en lenguajes de programación.

INFORMIX-ESQL (embedded SQL): Permite a los programadores de SQL tener acceso directo y manipular bases de datos desde los programas en C, COBOL, FORTRAN o Ada. Esto les facilita la manipulación de rutinas en fracciones de tiempo en comparación con lenguajes convencionales.

PRODUCTOS OPEN CASE

Proporcionan una arquitectura de sistemas abierta y flexible que puede acomodar múltiples herramientas CASE, así como sus técnicas, con INFORMIX y otros lenguajes.

INFORMIX OpenCase/ToolBus: Es un ambiente para el desarrollo de software integrado para el sistema operativo UNIX. Soporta muchas de las mejores herramientas CASE.

INFORMIX-OpenCase Encapsulator: Permiten al usuario ampliar y optimizar el ambiente de desarrollo de la aplicación OpenCase/ToolBus añadiendo desarrollos del usuario, tercera partición y herramientas UNIX existentes sin modificaciones al código fuente.

INFORMIX-4GL for ToolBus: Es un lenguaje de cuarta generación gráfico para ambientes desarrollados en OpenCase/ToolBus y es completamente compatible con toda la familia INFORMIX-4GL.

Wingz: Es una hoja de cálculo fácil de usar y de alto desempeño. Incluye HyperScript, un ambiente poderoso de programación similar al Inglés para crear aplicaciones gráficas de usuario final.

SERVIDORES DE BASES DE DATOS

INFORMIX-SE: Es un servidor para ambientes donde no son necesarios una alta disponibilidad o desempeño de nivel OLTP (On Line Transaction Processing), pero necesitan los servidores de bases de datos SQL multiusuario. Ofrece facilidad de operación en ambientes de bases de datos particulares, corriendo aplicaciones pequeñas o medianas. INFORMIX-SE está basado en el estándar de UNIX para acceso de archivos: INFORMIX C-ISAM.

C-ISAM: Es una librería flexible X/OPEN de funciones en C para la creación y manipulación de archivos secuenciales indexados.

INFORMIX-OnLine: Es un servidor de bases de datos OLTP de alto desempeño y tolerante a fallas. Este producto abarca el procesamiento de datos en ambiente multimedia permitiendo el almacenamiento de BLOBs (Binary Large Objects) como campos de la base de datos e incorporarlos a las aplicaciones. También ofrece las facilidades de bases de datos distribuidas.

INFORMIX-TP/XA: Enlaza el INFORMIX-OnLine a los manejadores de transacciones para soportar transacciones globales. Las transacciones globales son transacciones que involucran múltiples bases de datos y tal vez múltiples sistemas de cómputo.

INFORMIX-OnLine/Secure: Ofrece las capacidades de INFORMIX-OnLine con características adicionales para aplicaciones con múltiples niveles de seguridad.

INFORMIX-OnLine for NetWare: Es un servidor para bases de datos tipo OLTP, de alto desempeño para ambientes NetWare 3.11 cliente/servidor. Proporciona alta disponibilidad, integridad de datos y capacidades para manejo de datos multimedia sin los procedimientos complicados de administración de bases de datos.

INFORMIX-OnLine/Optical: Integra las capacidades de almacenamiento masivo de subsistemas de disco óptico con las capacidades de multimedia de OnLine. Es una API (Application Programmers Interface) de alto nivel que permite a los subsistemas ópticos WORM (Write-Once-Read-Many) para conectarse directamente al ambiente OnLine.

INSTRUCCIONES DE MANIPULACIÓN DE BASE DE DATOS

Dentro de la herramienta de programación INFORMIX 4GL se conocen como instrucciones de manipulación de Base de Datos a todas aquellas que permiten la actualización de información. En INFORMIX 4GL todas las operaciones de entrada/salida de información se concentran en cuatro comandos básicos de SQL que permiten:

- ⇒ Consultar Registros (SELECT)
- ⇒ Modificar Registros (UPDATE)
- ⇒ Insertar Registros (INSERT INTO)
- ⇒ Borrar Registros (DELETE)

CONSULTAR REGISTROS

El comando SELECT recupera datos de un Base de Datos y los devuelve en forma de resultado de la consulta. Se pueden utilizar columnas, constantes y funciones. El comando SELECT consta de siete cláusulas:

SELECT

```
INTO {variable}
FROM {especificación de la tabla}
WHERE { condición de búsqueda}
GROUP BY { columna de agrupación }
HAVING { condición de búsqueda}
ORDER BY { especificación de ordenación}
INTO TEMP { nombre de la columna }
```

Para el comando SELECT es necesaria la cláusula FROM. Las seis cláusulas restantes son opcionales.

Cláusula **INTO**: Los datos leídos por la instrucción **SELECT** son almacenados en las variables indicadas. Las variables tienen que estar definidas y sus tipos deben corresponder a los tipos de las columnas indicadas en el **SELECT**.

Cláusula **FROM**: Se deben indicar las tablas que contengan las columnas especificadas en la sección del **SELECT**. Las tablas deben existir en la Base de Datos.

Cláusula **WHERE** : Para definir la condición se puede utilizar cualquier operador condicional de **INFORMIX**. Se puede especificar más de una condición con los operadores lógicos **AND** y **OR**. Mediante una condición de igualdad entre dos columnas de diferentes tablas se puede crear una tabla compuesta (**join**).

OPERADORES CONDICIONALES DE INFORMIX

Permiten definir el alcance que tendrán los enunciados de **DELETE**, **UPDATE** y **SELECT**. Se dividen en seis tipos principales:

- .. **OPERADORES RELACIONALES**
- .. **BETWEEN**
- .. **IN**
- .. **IS NULL**
- .. **LIKE o MATCHES**
- .. **SUBQUERYS (in, exist, all/any/some)**

OPERADORES RELACIONALES

Dentro de éstos se cuenta con:

- ↔ **>** Mayor que , **>=** mayor o igual que
- ↔ **<** Menor que, **<=** menor o igual que
- ↔ **=** igual que, **<>** diferente a

BETWEEN: Checa un valor intermedio entre un rango.

IN: Checa un valor dentro de un rango.

IS NULL: Checa el valor nulo de una columna.

LIKE o MATCHES: Checa que se cumpla con una cadena de características específicas (**%** y ***** respectivamente).

SUBQUERYS: Los Subqueries son enunciados **SELECTS** utilizados dentro de una condición.

IN.- Checa que el valor exista en el resultado del subquery.

EXIST.- Valida la existencia de una o más columnas en el subquery.

ALL.- La condición es verdadera si cumple para todos los valores del subquery.

ANY.- La condición es verdadera si al menos uno la cumple.

SOME.- Sinónimo de **ANY**.

Cláusula **GROUP BY** .- Permite realizar agrupaciones de registros para sumar resultados. Todas las columnas listadas en la sección **Select** deben aparecer en esta sección.

Cláusula HAVING.- Permite especificar condiciones para los grupos realizados en la sección de GROUP BY. Se puede utilizar cualquier operador condicional de INFORMIX , así como operadores lógicos para especificar más de una condición.

Cláusula ORDER BY.- Permite mostrar la información ordenada para las columnas indicadas en esta sección. El orden puede ser ASCendente ó DESCendente, según se indique; el default es ASCendente. Si las columnas para ordenar pertenecen a diferentes tablas hay que indicar la tabla de cada una de ellas.

Cláusula INTO TEMP.- Crea una tabla temporal con el resultado de la consulta. La tabla desaparece cuando termina la sesión o con DROP TABLE.

MODIFICAR REGISTROS

UPDATE.- Se deberá hacer referencia al nombre de una tabla de la Base de Datos. El alcance del enunciado está dado por una condición, si ésta es omitida se modificarán todos los registros de la tabla.

INSERTAR REGISTROS

INSERT INTO.- Se debe hacer referencia al nombre de una tabla de la Base de Datos. Los valores a dar de alta pueden ser literales o variables separadas por comas y en el mismo orden en el que fueron declaradas, en caso de omitir las columnas después del nombre de la tabla se deberá registrar los valores en el orden físico de las columnas, en la tabla, separando por comas cada valor.

BORRAR REGISTRO

DELETE FROM.- Deberá hacerse referencia al nombre de una tabla de una Base de Datos. El alcance del enunciado está dado por una condición, si ésta es omitida se borrarán todos los registros de la tabla.

INTEGRIDAD

La potencia de un manejador de Base de Datos se mide no sólo por la rapidez de acceso a la información o por la facilidad que ofrezca para programar las aplicaciones, sino también se toma en cuenta qué tanto ofrece en el rubro de la seguridad de la información.

INFORMIX provee una serie de herramientas dedicadas al respaldo, preservación y recuperación tanto de la información como de la estructura de la BD que la contiene.

Son básicamente tres los métodos que se pueden llevar a cabo para generar un respaldo:

Método 1. Uso de la Utilería Archive

Archive permite generar respaldos con tres niveles de especificidad:

Archive de nivel 0. Se corre para respaldar íntegramente una instancia de INFORMIX; en él se registra desde la configuración que para instancia específica guarda

INFORMIX, hasta la estructura, configuración y permisos de la base de datos, incluyendo obviamente la información que contenga.

Archive de nivel 1. En el Archive de nivel 1 se respaldan sólo las diferencias que INFORMIX identifica en el estado actual de la instancia contra la que encuentra respaldada en el Archive de nivel 0. Así un Archive de nivel 1 sólo respaldará variaciones de tamaño de tablas, modificaciones de configuración, etc. De modo que para obtener un recuperación completa de la instancia en su última forma será necesario restaurar primero el Archive de nivel 0 y después su complemento, el archivo generado por el Archive de nivel 1.

Archive de nivel 2. Viene a ser lo que el Archive de nivel 1 es el de nivel 0, Archive de nivel 2 genera un archivo en el que se registran los cambios encontrados en relación al generado por el Archive de nivel 1. INFORMIX recomienda correr este nivel de Archive sólo para BD demasiado grandes, que registran enormes volúmenes de transacciones al día de modo de perder la menor información posible en caso de presentarse algún problema.

Método 2. El uso de espejos

Durante el proceso de creación de una BD, INFORMIX permite generarla -si así conviene al administrador del sistema- con un manejo de espejo, que significa que existirá una copia fiel de la BD actualizándose en todo momento que ocurra un cambio en la original. Este espejo puede existir en el mismo disco que da cabida al original o se puede declarar para que resida en un disco diferente, lo que permite que si algún problema se presentara en el funcionamiento de alguno de los discos, pueda continuarse con el uso de la aplicación de forma transparente para el usuario, ya que INFORMIX automáticamente continuará con las transacciones en el espejo o viceversa.

Hasta la versión 5 de INFORMIX el espejo aún tiene que residir en algún disco del mismo equipo, ya que no se provee ninguna opción para el manejo de espejos remotos; es decir, que existan en discos de otro equipo.

Método 3. La migración

La migración no se considera un método de respaldo simplemente; sin embargo, sí coadyuva a la preservación de la información cuando se presenta alguna de las siguientes necesidades:

- ⇨ Mover la BD del ambiente de desarrollo al ambiente de producción.
- ⇨ Mover la BD a una plataforma de Hardware distinta.
- ⇨ Por necesidades de distribución de la información a los usuarios.
- ⇨ Por la necesidad de reorganizar el espacio en disco.

En la siguiente lista se resumen la utilerías y comandos que INFORMIX provee para tal fin.

Comando Unload. Permite "bajar" de la BD información específica a un archivo ASCII, dicha información se delimita por separadores que pueden ser definidos en el momento de ejecutar la descarga.

Comando Load. Funciona como la contraparte del comando anterior, permitiendo grabar en la base de datos la información, a partir del archivo en ASCII generado previamente.

Utilería DBSchema. DBSchema escribe en un archivo ASCII todas las instrucciones SQL necesarias para reproducir una BD y sus tablas. Este archivo puede ejecutarse desde las herramientas INFORMIX apropiadas y como resultado se obtendrá una copia fiel de la estructura de la BD original incluyendo sus tablas.

Utilería Dbexport y Dbimport. Dbexport puede direccionar su salida para escribir en cinta o en disco, creando un directorio que lleva el nombre de la BD. En el interior creará un archivo ASCII por cada tabla incluida en la BD. Adicionalmente crea otro archivo que contiene el lenguaje de definición de datos y un resumen de la información necesaria para regenerar la información de la BD en otro servidor INFORMIX. **Dbimport** toma la información del directorio generado por Dbexport, y reproduce la estructura de las tablas y BD para luego grabar en ella la información.

Utilerías Tbunload y Tbload. A diferencia de las opciones anteriormente descritas, las presentes sólo se utilizan para llevar a cabo respaldos en dispositivos de cinta, por tal motivo requieren de consideraciones adicionales en cuanto al manejo de la paginación de las unidades de cinta.

5.2. PROGRAMACIÓN

A continuación se muestran los listados de los programas más representativos del SIFACOR. Esta codificación se apoyó en los pseudocódigos que aparecieron en el capítulo de diseño.

- **Módulo de Catálogos**
Se lista el programa FTM75001.4GL que permite dar de alta clientes en el catálogo respectivo.
- **Módulo de Facturación**
Se presenta el programa FTP10000.4GL Pedidos. Tiene como fin permitir la captura de la información para generar los documentos y el procesamiento de la información capturada.
- **Módulo de Consultas y Reportes**
Se muestra el programa FTM75000.4GL que permite llevar a cabo altas, bajas consultas, cambios y elaborar reportes del catálogo Clientes.
- **Módulo de administración del sistema**
Se lista el programa SPMA5000.4GL, cuya función es dar mantenimiento al archivo de vistas del sistema por usuario y programa. Presenta funciones de altas y bajas del archivo Vistas. Se lista también el SPMA6000.4GL que da mantenimiento al catálogo de Folios.

```

# .....
# SISTEMA: FACTURACION
# PROGRAMA: frm75001.4g
# FUNCION: alta_clientes
# DESCRIPCION: Permite dar de alta Clientes, recibiendo como parametro
# una bandera que identifica si el proceso a realizar será
# ejecutado solamente una vez o hasta que el usuario pulse
# la tecla de ESCAPE
# DESCRIPCION PARAMETROS:
# gr_clientes # Registro igualado al de clientes
# wc_bandera # Bandera para realizar el proceso indefinido o no
# Si es TRUE se está llamando de Cat. Clientes. Hasta que se pulse ESCAPE
# Si es FALSE se está llamando de facturación. Una vez
# .....
DATABASE factura

FUNCTION alt_clientes(wc_bandera,pc_idpro,pc_idtty,pc_nom)
DEFINE wc_bandera,
pc_cat CHAR,
pc_idpro,
pc_idtty CHAR(8),
pc_nom LIKE parametros_d_parametro,
ws_repres SMALLINT,
gr_clientes RECORD LIKE clientes,
wc_clientes RECORD LIKE clientes,
gs_ok SMALLINT,
wn_clave LIKE clientes_c_cliente

WHENEVER ERROR CALL err_handle

OPTIONS INPUT WRAP

IF NOT wc_bandera THEN
OPEN FORM frm75000 FROM "frm75000"
DISPLAY FORM frm75000
END IF

CALL limpia("ALTAS",TRUE,pc_idpro,pc_idtty,pc_nom) # Desp de movimiento
DISPLAY "" AT 1,1
DISPLAY "" AT 2,1
LET int_flag=FALSE
LET ws_repres=0
WHILE TRUE
INITIALIZE gr_clientes TO NULL
CLEAR FORM

INPUT BY NAME gr_clientes_d_nombre THRU gr_clientes_d_tel2
WITHOUT DEFAULTS

ON KEY(ESC)
LET int_flag=TRUE # Valida si se oprime esc
EXIT INPUT

AFTER FIELD d_rfc
#DECLARE apu01 CURSOR FOR
SELECT * INTO gr_clientes
FROM clientes
WHERE d_rfc = gr_clientes_d_rfc

IF STATUS<>NOTFOUND THEN
ERROR * Esta cliente ya fue dado de alta
DISPLAY BY NAME gr_clientes
SLEEP 3
RETURN FALSE, gr_clientes_c_cliente, gr_clientes_d_nombre
END IF

AFTER FIELD d_represent
IF gr_clientes_d_represent = "" THEN
# Carga los valores del cliente en los del representante
# en caso de que sea el mismo
LET gr_clientes_d_represent=gr_clientes_d_nombre
LET gr_clientes_d_calle2=gr_clientes_d_calle
LET gr_clientes_d_numero2=gr_clientes_d_numero
LET gr_clientes_d_colonia2=gr_clientes_d_colonia
LET gr_clientes_d_ciudad2=gr_clientes_d_ciudad
LET gr_clientes_d_estado2=gr_clientes_d_estado
LET gr_clientes_d_cp2=gr_clientes_d_cp
LET gr_clientes_d_tel2=gr_clientes_d_tel
DISPLAY BY NAME gr_clientes_d_represent,
gr_clientes_d_calle2,gr_clientes_d_numero2,
gr_clientes_d_colonia2,gr_clientes_d_ciudad2,
gr_clientes_d_estado2,gr_clientes_d_cp2,
gr_clientes_d_tel2
NEXT FIELD d_contacto
ELSE
INITIALIZE gr_clientes_d_calle2,gr_clientes_d_numero2,
gr_clientes_d_colonia2,gr_clientes_d_ciudad2,
gr_clientes_d_estado2,gr_clientes_d_cp2,
gr_clientes_d_contacto,gr_clientes_d_tel2 TO NULL
DISPLAY BY NAME gr_clientes_d_represent,
gr_clientes_d_calle2,gr_clientes_d_numero2,
gr_clientes_d_colonia2,gr_clientes_d_ciudad2,
gr_clientes_d_estado2,gr_clientes_d_cp2,
gr_clientes_d_contacto,gr_clientes_d_tel2

gr_clientes_d_contacto,gr_clientes_d_tel2
END IF

END INPUT

IF int_flag THEN # Valida si se interrumpe el alta
LET int_flag=FALSE
IF NOT wc_bandera THEN
CLOSE FORM frm75000
RETURN FALSE, gr_clientes_c_cliente, gr_clientes_d_nombre
ELSE
EXIT WHILE
END IF
END IF

# Inserta los datos en la tabla

SET LOCK MODE TO WAIT
BEGIN WORK
LOCK TABLE clientes IN EXCLUSIVE MODE
LET wn_clave = get_next_clave("clientes","c_cliente")
DISPLAY wn_clave TO c_cliente
LET gr_clientes_c_cliente = wn_clave

INSERT INTO clientes VALUES(gr_clientes,")
ERROR * Registro dado de alta,

COMMIT WORK
SLEEP 2
IF NOT wc_bandera THEN
CLEAR FORM
CLOSE FORM frm75000
RETURN TRUE, gr_clientes_c_cliente, gr_clientes_d_nombre
END IF
END WHILE
CLEAR FORM # Borra datos de la pantalla
CALL limpia(" ",FALSE,pc_idpro,pc_idtty,pc_nom) # Limpia avisos pantalla
RETURN TRUE, gr_clientes_c_cliente, gr_clientes_d_nombre
END FUNCTION # Fin alt_clientes

(
# .....
** FUNCION : limpia
** DESCRIPCION: Despliega un mensaje para indicar el tipo de movimiento hecho
# .....
)
FUNCTION limpia(pc_lex,ps_tipo,pc_idpro,pc_idtty,pc_nom)
DEFINE pc_lex CHAR(20), # Define variables locales
ps_tipo CHAR(20),
ps_col SMALLINT,
pc_idpro,
pc_idtty CHAR(8),
pc_nom LIKE parametros_d_parametro

WHENEVER ERROR CALL err_handle

LET ps_col=(80-LENGTH(pc_lex))/2

IF ps_tipo THEN # Manda un mensaje a la pantalla
DISPLAY pc_lex CLIPPED AT 5,ps_col ATTRIBUTE(REVERSE)
ELSE
DISPLAY "" AT 5,1 # Limpia el mensaje
CALL dis_panta(pc_idpro,pc_idtty,pc_nom)
*****Despliega la forma, fecha y hora del sistema e id terminal y programa*****
END IF
END FUNCTION # Fin de limpia

```

```

*****
** SISTEMA FACTURACION
** PROGRAMA: Rp10000 4gl
** DESCRIPCION: Rutina General para FTF10000 4gl Pedidos
** FUNCION: Captura de la informacion para generar los documentos
** y procesa la informacion.
** DESCRIPCION PARAMETROS
*****
)
DATABASE compras
GLOBALS "Rp10000 4gl"
DEFINE
  ga_par ARRAY[20] OF RECORD
    #Detalle
    c_producto LIKE partidas c_producto,
    d_producto LIKE productos d_producto,
    cantidad LIKE partidas cantidad,
    unidad LIKE partidas unidad,
    precio_unit LIKE partidas precio_unit,
    importe LIKE partidas importe
  END RECORD,

  ga_pro RECORD LIKE productos *, #Inf de Detalle
  ga_pdrs RECORD LIKE proveedores *,
  ga_count # Num de Reg de Detalle
  wn_actua #Pos actual de arreglo
  wn_pantalla #Pos en pantalla
  wn_total #Total de Registros capturados
  wf_ciclo #Bandera que indica si se regresa a la captura de datos
  wn_ok SMALLINT, #Bandera OK
  wc_proci LIKE productos c_producto #Contiene la clave del producto
# *****
# Funcion : Rp10000
# Descripción: Ciclo Principal de Proceso de Generación de Documentos
# *****
FUNCTION Rp10000
DEFINE
  wa_ok #Bandera de Ok
  dc_dom #Control de Flujo de Inputs.
  st,
  M,
  tot,
  x,
  y,
  z DECIMAL(7,2)

  # TRUE si el usuario regresa
  #del INPUT ARRAY al INPUT

OPEN FORM Rp10000 FROM "fcd001"

LET wf_ciclo = FALSE

DISPLAY FORM Rp10000

WHILE TRUE
  LET inf_flag = FALSE
  CALL ar_doc() # Captura de datos del pedido.
  IF inf_flag THEN # Usuario Cancelo
    EXIT WHILE # Termina Proceso
  END IF
  CALL partdas()
  RETURNING wf_ciclo, x, y, z # Inf. gen
  IF inf_flag THEN #El Usuario Cancelo el Documen.
    LET wf_ciclo = FALSE #y se permite la captura de uno
    CONTINUE WHILE #nuevo.
  END IF
  IF wf_ciclo THEN # Se desea regresar a mod. los
    CONTINUE WHILE #generales.
  END IF
  CALL dolares(x,y,z) RETURNING st,M,tot,dc_dom
  CALL proceso_wtc(st,M,tot,dc_dom) RETURNING wa_ok #Realiza el documento
  DISPLAY FORM Rp10000
END WHILE
CLOSE FORM Rp10000
END FUNCTION

*****
FUNCTION ar_doc()
*****
DEFINE
  dnombre LIKE compradores nombre,
  wnom LIKE proveedores nombre,
  wdes LIKE areas descripcion,
  gs_ok SMALLINT,
  wbod LIKE bodegas direccion

OPTIONS INPUT NO WRAP

DISPLAY "" AT 1,1
DISPLAY "" AT 2,1
LET inf_flag=FALSE
INITIALIZE gr_documentos * TO NULL
*****
#RUTINA DE ALTA, VALIDACION, Y/O GENERACION DE COMPRADORES
*****
LET gr_documentos f_doc= TODAY

```

```

DISPLAY BY NAME gr_documentos f_doc
INPUT BY NAME
  gr_documentos no_req, gr_documentos f_cobzacion,
  gr_documentos f_entrega,
  gr_documentos c_comprador, gr_documentos c_area,
  gr_documentos c_proveedor, gr_documentos no_bodega,
  gr_documentos cond_pago,
  gr_documentos cond_entrega, gr_documentos cen_cos,
  gr_documentos transporte
  WITHOUT DEFAULTS

ON KEY(ESC)
  LET inf_flag=TRUE # Valida si se oprime esc
  EXIT INPUT
ON KEY(F1)
  ***** (LLAMADAS DEL <F1> PARA CAMPOS DE CONSULTA
  IF INFIELD (c_comprador) THEN
    CALL lista_catalogo (10,20,
  "compradores", "c_comprador", "nombre", TRUE)
    RETURNING gs_ok, gr_documentos c_comprador,
  gr_compradores nombre
    DISPLAY BY NAME gr_documentos c_comprador,
  gr_compradores nombre
    NEXT FIELD c_area
  END IF
  IF INFIELD (c_area) THEN
    CALL lista_catalogo (10,20, "areas", "c_area", "descripcion", TRUE)
    RETURNING gs_ok, gr_documentos c_area, gr_areas descripcion
    DISPLAY BY NAME gr_documentos c_area, gr_areas descripcion
    NEXT FIELD c_proveedor
  END IF
  IF INFIELD (c_proveedor) THEN
    CALL lista_catalogo (10,20, "proveedores", "c_proveedor", "nombre
  ", TRUE)
    RETURNING gs_ok, gr_documentos c_proveedor, wnom
    DISPLAY gr_documentos c_proveedor, wnom TO
  c_proveedor, proveedores nombre
    NEXT FIELD no_bodega
  END IF
  IF INFIELD (no_bodega) THEN
    CALL lista_catalogo (10,20, "bodegas", "no_bodega", "direccion", TRUE)
    RETURNING gs_ok, gr_documentos no_bodega, gr_bodegas direccion
    DISPLAY BY NAME gr_documentos no_bodega, gr_bodegas direccion
    NEXT FIELD cond_pago
  END IF

AFTER FIELD c_comprador
  IF gr_documentos c_comprador IS NULL THEN
    ERROR " Campo requerido .. "
    NEXT FIELD c_comprador
  ELSE
    SELECT nombre INTO dnombre
    FROM compradores
    WHERE c_comprador = gr_documentos c_comprador
    DISPLAY dnombre TO nombre
    NEXT FIELD c_area
  END IF

AFTER FIELD c_area
  IF gr_documentos c_area IS NULL THEN
    ERROR " Campo Requerido .. "
    NEXT FIELD c_area
  ELSE
    SELECT descripcion INTO wdes
    FROM AREAS
    WHERE c_area = gr_documentos c_area
    IF STATUS = NOTFOUND THEN
      ERROR " Area no existente .... "
      NEXT FIELD c_area
    END IF
    DISPLAY wdes TO descripcion
    NEXT FIELD c_proveedor
  END IF

AFTER FIELD c_proveedor
  IF gr_documentos c_proveedor IS NULL THEN
    ERROR " Campo Requendo .. "
    NEXT FIELD c_proveedor
  ELSE
    SELECT nombre INTO wnom
    FROM PROVEEDORES
    WHERE c_proveedor = gr_documentos c_proveedor
    IF STATUS = NOTFOUND THEN
      *****RUTINA DE LLAMADO AL ALTA DE PROVEEDORES*****
      PROMPT "EL PROVEEDOR NO EXISTE Desea su ALTA ? (S/N) "
      FOR CHAR act_cat
      IF act_cat MATCHES "Ss" THEN
        OPEN FORM fcp001 FROM "fcp001"
        DISPLAY FORM fcp001
        CALL at_proveedores()

```

```

RETURNING prov_cat, prov_nom
DISPLAY FORM fcd001
DISPLAY gr_proveedores c_proveedor, gr_proveedores nombre
      TO documentos c_proveedor, proveedores nombre
SLEEP 3
NEXT FIELD no_bodega
ELSE
NEXT FIELD c_proveedor
END IF
DISPLAY wnom TO proveedores nombre
NEXT FIELD no_bodega
END IF
DISPLAY wnom TO proveedores nombre
NEXT FIELD no_bodega
END IF

AFTER FIELD no_bodega
IF gr_documentos no_bodega IS NULL THEN
ERROR "      Campo Requerido. "
NEXT FIELD no_bodega
ELSE
SELECT direccion INTO wbod
FROM bodegas
WHERE no_bodega = gr_documentos no_bodega
IF STATUS = NOTFOUND THEN
ERROR "      Bodega Inexistente "
NEXT FIELD no_bodega
END IF
DISPLAY wbod TO bodegas direccion
NEXT FIELD cond_pago
END IF

BEFORE FIELD cen_cos
LET gr_documentos cen_cos = 603
END INPUT

END FUNCTION          # Fin at_doc

# Funcion : proces_etc
# Parametros : pc_tipodoc -> "F" Generacion de Facturas
#              "C" Generacion de Notas de Credito
#              "A" Generacion de Notas de Cargo
#              "R" Generacion de Facturas Roaming
# Descripcion: Realiza el documento.
FUNCTION proceso_etc(j,l,k,gc_dom)
DEFINE
wag CHAR(50),
gc_dom,
wn_folio SMALLINT, #Numero de Folio
l, #Contador,
rec,
wag,
wb_ok SMALLINT, #Bandera de OK
j,
l,
k DECIMAL(7,2)

IF NOT confirm() THEN #Si no se confirma se cancela el proceso
RETURN FALSE
END IF

BEGIN WORK #Empieza la Transaccion
SET LOCK MODE TO WAIT #Cualquier otra persona que este por
#generar un documento tendra que
#esperar un segundo

LET wb_ok = FALSE
# Se inserta en la tabla de documentos
INSERT INTO documentos VALUES(0, gr_documentos f_doc, gr_documentos no_req,
gr_documentos f_cobzacion, gr_documentos c_comprador, gr_documentos c_area,
gr_documentos c_proveedor, gr_documentos no_bodega, gr_documentos f_entrega,
gr_documentos cond_pago, gr_documentos cond_entrega, gr_documentos cen_cos,
gr_documentos transporte, j, l, k, gc_dom) #Almacena el Documento

# Se busca cual es mayor numero de folio -siempre corresponde al ultimo.
# para grbarlo a manera de que coincida en partidas
SELECT MAX(no_doc)
INTO wn_folio
FROM documentos

# Una vez obtenido el folio se graba el registro correspondiente en partidas
LET ga_count = ga_count - 1 #Quita el registro en blanco provocado

FOR i = 1 TO ga_count #Almacena el detalle
INSERT INTO partidas
VALUES (wn_folio, ga_par[i] c_producto, ga_par[i] d_producto, ga_par[i] cantidad,
ga_par[i] unidad, ga_par[i] precio_unit, ga_par[i] importe )
END FOR
LET wag = "ibrefisraelcompras"

```

```

COMMIT WORK #hagamoslo realidad.
# *****
CALL repcomm(wn_folio, ga_count, j, l, k)
RETURN TRUE
END FUNCTION

# Funcion confirmar
# Descripcion Solicita la confirmacion del proceso.
FUNCTION confirmar()
DEFINE v_flag SMALLINT

OPEN WINDOW conf AT 08,23 WITH 3 ROWS, 35 COLUMNS ATTRIBUTE(ORDER)

MENU "CONFIRMAR PEDIDO "
COMMAND "SI" "Realizar el Pedido "
LET v_flag = TRUE
EXIT MENU
COMMAND "NO" "Cancelar el Pedido "
LET v_flag = FALSE
EXIT MENU

END MENU

CLOSE WINDOW conf

RETURN v_flag
END FUNCTION

# *****
** FUNCION : init_arms()
** DESCRIPCION : Inicializa ga_doc y ga_pro a nulos
FUNCTION init_arms()
DEFINE i SMALLINT

#WHENEVER ERROR CALL err_handle

FOR i = 1 TO 20
INITIALIZE ga_par[i] TO NULL
END FOR
END FUNCTION

# Funcion : btulo
# Descripcion : Despliega el titulo del proceso
FUNCTION btulo()
DEFINE
pc_tipodoc CHAR(1)

#WHENEVER ERROR CALL err_handle

CASE pc_tipodoc
WHEN "F"
CALL limpia_f("FACTURAS",TRUE)
WHEN "C"
CALL limpia_f("NOTAS DE CREDITO",TRUE)
WHEN "A"
CALL limpia_f("NOTAS DE CARGO",TRUE)
WHEN "R"
CALL limpia_f("FACTURACION ROAMING",TRUE)
END CASE
END FUNCTION

# Funcion : limpia
# Descripcion : Despliega datos generales de pantalla
FUNCTION limpia_f(pc_tip, ps_tipo) # Recibe el tipo de mov dado
DEFINE pc_tip CHAR(20), # Define variables locales
ps_tipo, # ps_col SMALLINT,
wf_fec DATE,
wc_hor CHAR(8)

#WHENEVER ERROR CALL err_handle

LET wf_fec= TODAY # Toma la fecha y hora del sistema
LET wc_hor= TIME
DISPLAY FORM fcp10000
DISPLAY gc_idpro AT 3,1 # Despliega los identificadores
DISPLAY gc_idty AT 4,1

LET gs_col=(80-LENGTH(gc_nom))/2 # Centrar el titulo programa
DISPLAY gc_nom CLIPPED AT 3,gs_col # Despliega titulo del programa

DISPLAY wf_fec USING "dd-mm-yy" AT 3,70 # Despliega fecha y hora sistema
DISPLAY wc_hor AT 4,70

```

```

LET ps_colx(80-LENGTH(pc_text CLIPPED))/2
IF ga_bpo THEN # Manda un mensaje a la pantalla
ELSE
  DISPLAY "" AT 5 1 # Limpia el mensaje
END IF

```

```

FUNCTION ind_facturacion()

```

```

DEFINE
  > SMALLINT

```

```

FOR j=1 TO 20
  INITIALIZE ga_par[j] * TO NULL
END FOR

```

```

INITIALIZE gr_documentos * TO NULL

```

```

LET ga_count = 0
END FUNCTION

```

```

)
# Funcion limpia
# Descripcion Despliega datos generales de pantalla
#
FUNCTION borra_ar(pcurr,pscr)

```

```

DEFINE
  pcurr,
  pscr,
  i SMALLINT

```

```

FOR i = pcurr TO 19
  LET ga_par[i] * = ga_par[i+1] *
END FOR

```

```

INITIALIZE ga_par[20] * TO NULL

```

```

FOR i = 1 TO 4
  DISPLAY ga_par[pcurr-pscr + i] * TO scr_docu[i] *
END FOR

```

```

END FUNCTION

```

```

*****
** FUNCION Alta de Proveedores **
*****
FUNCTION alt_proveedores()

```

```

OPTIONS INPUT NO WRAP

```

```

INITIALIZE gr_proveedores * TO NULL
CLEAR FORM
OPTIONS INPUT WRAP

```

```

INPUT BY NAME gr_proveedores nombre THRU gr_proveedores con_area

```

```

WITHOUT DEFAULTS
ON KEY(ESC)
LET int_flag=TRUE # Valida si se oprme esc
EXIT INPUT
END INPUT

```

```

IF int_flag THEN # Valida si se interrumpe el alta
  LET int_flag=FALSE
END IF

```

```

SET LOCK MODE TO WAIT
BEGIN WORK
LOCK TABLE proveedores IN EXCLUSIVE MODE
LET gr_proveedores c_proveedor * get_next_clave("proveedores","c_proveedor")
DISPLAY BY NAME gr_proveedores c_proveedor

```

```

# Inserta los datos en la tabla
INSERT INTO proveedores VALUES(gr_proveedores *)
ERROR " " Registro dado de alta.

```

```

COMMIT WORK
SLEEP 2
RETURN gr_proveedores c_proveedor, gr_proveedores nombre
END FUNCTION # Fin alt_proveedores

```

```

*****
# programa part001 4gl
*****
FUNCTION partdas()

```

```

DEFINE
  vn_folio #Pos actual de arreglo
  vn_actual #Pos en pantalla
  vn_pantalla #Total de Registros capturados
  vn_total #Bandera que indica si se registra
  wf_cicle # a la captura de datos gener
  i #Bandera OK
  vn_ok SMALLINT
  nomprod LIKE productos_d_producto
  wc_pron LIKE productos_c_producto #Contiene la clave del producto
  a.

```

```

b.
c. DECIMAL (15,2)

```

```

LET wf_cicle = FALSE

```

```

FOR j=1 TO 4
  INITIALIZE ga_par[j] * TO NULL
END FOR

```

```

OPEN WINDOW w_parti AT 10,1 WITH FORM "freq001"

```

```

INPUT ARRAY ga_par WITHOUT DEFAULTS FROM scr_docu *

```

```

BEFORE FIELD c_producto
  LET vn_actual = arr_curr() #Obten posiciones de arreglo
  LET vn_pantalla = scr_line()
  LET vn_total = arr_count() #Total de reg capturados y
  LET wc_pron = #El producto que contiene el
  ga_par[vn_actual] c_producto #renglon

```

```

AFTER FIELD c_producto
  IF vn_actual = 1 AND #Si estamos en el primer renglon y
  (gl_lastkey() = 2000 THEN #se presiona la flecha para arriba
  NEXT FIELD c_producto
  END IF

```

```

  IF ga_par[vn_actual] c_producto = "N" THEN
    NEXT FIELD d_producto
  ELSE
    SELECT d_producto INTO nomprod
    FROM productos
    WHERE c_producto = ga_par[vn_actual] c_producto
    IF ga_par[vn_actual] c_producto IS NULL THEN
      EXIT INPUT
    ELSE
      IF STATUS = NOTFOUND THEN
        ERROR " Clave no EXISTENTE..... "
        NEXT FIELD c_producto
      END IF
    END IF
    DISPLAY nomprod TO scr_docu[vn_pantalla] d_producto
    NEXT FIELD cantidad
  END IF

```

```

BEFORE FIELD cantidad
  IF ga_par[vn_actual] c_producto IS NULL THEN
    EXIT INPUT
  END IF

```

```

  DISPLAY ga_par[vn_actual] precio_unit
  TO scr_docu[vn_pantalla] precio_unit
  LET ga_par[vn_actual] cantidad = 1

```

```

AFTER FIELD cantidad
  IF ga_par[vn_actual] cantidad IS NULL THEN
    LET ga_par[vn_actual] cantidad = 1
    DISPLAY ga_par[vn_actual] cantidad
    TO scr_docu[vn_pantalla] cantidad
  END IF

```

```

AFTER FIELD precio_unit
  IF ga_par[vn_actual] precio_unit IS NULL THEN
    ERROR "El Precio Unitario es requerido "
    NEXT FIELD precio_unit
  END IF

```

```

  LET ga_par[vn_actual] importe = ga_par[vn_actual] cantidad *
  ga_par[vn_actual] precio_unit

```

```

AFTER ROW
  DISPLAY ga_par[vn_actual] importe TO scr_docu[vn_pantalla] importe

```

```

CALL calc_totals(vn_total)
RETURNING a,b,c

```

```

AFTER INPUT
  IF NOT Hay_info(arr_count()) THEN # verifica que se halla algo
    ERROR "Por lo menos se debe capturar un renglon"
    NEXT FIELD c_producto
  END IF

```

```

  IF NOT falta_algo(arr_count()) THEN
    ERROR " Informacion incompleta Por favor revisar
    NEXT FIELD c_producto
  END IF

```

```

  #IF NOT revalida() THEN
  # ERROR " Se capturo una clave invalida
  # NEXT FIELD c_producto
  # END IF

```

```

IF ga_par[m_actual] c_producto = "" OR
ga_par[m_actual] c_producto IS NULL THEN
EXIT INPUT
END IF

IF ga_par[m_actual] cantidad IS NULL THEN
LET ga_par[m_actual] cantidad = 1
DISPLAY ga_par[m_actual] cantidad TO
scr_docu[m_pantalla] cantidad
END IF

# DISPLAY ga_par[m_actual] d_producto, ga_par[m_actual] precio_unit
# TO scr_docu[m_pantalla] d_producto, scr_docu[m_pantalla] precio_unit
LET ga_par[m_actual] importe = #Se calcula el importe
ga_par[m_actual] cantidad *
ga_par[m_actual] precio_unit

DISPLAY ga_par[m_actual] importe
TO scr_docu[m_pantalla] importe

#CALL calc_totales(mn_total) #Calcula y Despliega Tots #
RETURNING x, y, z

ON KEY(F1) #Ayudas
IF infield(c_producto) THEN
CALL lista_catalogo(10,20,"productos",c_producto",
"d_producto",TRUE)
RETURNING m_ok, ga_par[m_actual] c_producto,
ga_par[m_actual] d_producto
DISPLAY ga_par[m_actual] c_producto,
ga_par[m_actual] d_producto
TO scr_docu[m_pantalla] c_producto,
scr_docu[m_pantalla] d_producto
END IF
NEXT FIELD cantidad

ON KEY(F2)
CALL borrar(m_actual,mn_pantalla)

ON KEY(F4)
IF infield(c_producto) THEN
EXIT INPUT
END IF

ON KEY(ESC) #El usuario cancela.
LET m_flag = TRUE
LET m_ciclo = TRUE
EXIT INPUT

END INPUT

LET ga_count = m_count()

CLOSE WINDOW w_parti
RETURN w_ciclo, a, b, c
END FUNCTION

# Funcion : hay_info (Auxiliar de input_arreglo)
# Parametros : pn_total -> Segun esto es el total de registros capturados
# Descripcion: Determina si se capturo algo.
#
FUNCTION hay_info(pn_total)
DEFINE
pn_total, #Total de registros indicado por m_count()
i, #Contador.
w_f_hay SMALLINT #Bandera de existencia.
#WHENEVER ERROR CALL err_handle

LET w_f_hay = FALSE #Por el momento no hay nada

FOR i = 1 TO pn_total #Barre el arreglo
IF ga_par[i] c_producto IS NOT NULL THEN #Si encuentra algo
LET w_f_hay = TRUE #marca la bandera
EXIT FOR #y que ya no se busque mas
END IF
END FOR
RETURN w_f_hay
END FUNCTION

# Funcion : calc_totales (Auxiliar de input_arreglo)
# Parametros : pn_count -> Es el total de registros capturados
# Descripcion: Calcula y Despliega el Subtotal, IVA y Total
#
FUNCTION calc_totales(pn_count)
DEFINE
pn_count, #Total de registros
i SMALLINT, #Contador
w_ST, #SubTotal
w_DE, #Descuento
w_SR, #SubTotal - Descuento
w_IV, #IVA de w_SR
w_TO, #Total
x, y, z, w_temp DECIMAL(7,2) #Importe Temporal.

LET w_ST = 0 #Inicializa todos los importes.
LET w_DE = 0
LET w_SR = 0
LET w_IV = 0
LET w_TO = 0

FOR i = 1 TO pn_count #Barre todo el arreglo
IF ga_par[i] importe IS NOT NULL THEN #Si el renglon tiene importe
LET w_ST = w_ST + ga_par[i] importe #Acumulado al Subtotal
END IF
END FOR

LET w_IV = w_ST * 0.15 #Calcula Iva.
LET w_TO = w_ST + w_IV #Calcula Total.
LET z = w_ST
LET y = w_IV
LET x = w_TO

RETURN x,y,z
END FUNCTION

# Funcion falta_algo(p_tot)
#
FUNCTION falta_algo(p_tot)
DEFINE
p_tot
i SMALLINT
#WHENEVER ERROR CALL err_handle

FOR i = 1 TO p_tot
IF ga_par[i] c_producto IS NOT NULL AND
(ga_par[i] importe IS NULL OR
ga_par[i] precio_unit IS NULL OR
ga_par[i] cantidad IS NULL OR
ga_par[i] unidad IS NULL) THEN
RETURN FALSE
END IF
END FOR
RETURN TRUE
END FUNCTION

# Funcion dolares
# DESCRIPCION: Preguntar si la moneda del pedido sera en dolares,
# mostrar el monto, y capturar quien autoriza
#
FUNCTION dolares(m,n,o)
DEFINE
dc_dom SMALLINT,
dol CHAR(1),
m,n,o,
stf,
m,
tot DECIMAL(15,2)

OPEN WINDOW dola AT 08.23 WITH FORM "dolares" ATTRIBUTE(BORDER)
INPUT BY NAME dol WITHOUT DEFAULTS
AFTER held dol
IF dol = "N" THEN
LET dc_dom = 1
ELSE
LET dc_dom = 2
END IF
END IF
DISPLAY STF, M, tot TO STF, M, tot
SLEEP 3
END INPUT
CLOSE WINDOW dola
RETURN STF, M, tot, dc_dom
END FUNCTION

# Funcion repcomint
# DESCRIPCION: elabora el reporte de salida
#
#
FUNCTION repcomint(mn_folio, f_count, m, n, o)
DEFINE
mn_folio,
f_count,
i,
f_copia SMALLINT,
m,n,o DECIMAL(15,2)

START REPORT comint to "raul.txt"
FOR i = 1 TO f_count #Por cada detalle mandalo al reporte.
OUTPUT TO REPORT comint(mn_folio, gr_documentos ?, ga_par[i] *, m,n,o)
END FOR
FINISH REPORT comint
RUN "lp_raul.txt"
# RUN "lp_dia.go3_raul.txt"
END FUNCTION

```

```

REPORT comin(r_folio r_doc, r_det, rm, rn, ro)
DLFIRE
r_folio SMALLINT,
r_doc RECORD LIKE documentos *,
r_prov RECORD LIKE proveedores *,
r_det RECORD:
    c_producto LIKE partidas c_producto,
    d_producto LIKE productos d_producto,
    cantidad LIKE partidas cantidad,
    unidad LIKE partidas unidad,
    precio_unit LIKE partidas precio_unit,
    importe LIKE partidas importe
# no_partida LIKE partidas no_partida
END RECORD,

r_prod RECORD LIKE productos *,
r_bod RECORD LIKE bodegas *,
rm, rn, ro,
r_sub,
r_ivs,
r_tot DECIMAL(7,2),
p_copa,
r_cic SMALLINT,
r_str1,
r_str2,
r_str3,
aju1,
aju2,
aju3,
ajustable1,
ajustable2,
ajustable3 CHAR(80),
car1,
car2,
car3 CHAR(50),
r_str CHAR(240),
cajustable1,
cajustable2,
cajustable3,
ajustado1,
ajustado2,
ajustado3 CHAR(80),
wc_dom CHAR(2),
sobra1,
sobra2,
sobra3,
ajustador1,
ajustador2,
ajustador3 CHAR(80)

PRINT r_doc cond_entrega
PRINT @a550.180Z
PRINT r_doc cond_pago
PRINT @a550.2350Z
PRINT r_bod responsable
PRINT @a050.90Z
PRINT r_doc f_doc
PRINT @a050.052Z
PRINT "1 de 1"
PRINT @a750.1300Z
PRINT r_doc no_rto
PRINT @a750.265Z
PRINT r_doc cen_cos
PRINT @a800.100Z
PRINT r_folio
PRINT @a800.858Z
PRINT "1005"
PRINT @a1150.90Z
PRINT @1

ON EVERY ROW
LET wc_dom = "N5"

SELECT carac1, carac2, carac3 INTO car1, car2, car3
FROM productos
WHERE c_producto = r_det c_producto

LET ajustable1 = r_det d_producto CLIPPED, "", car1 CLIPPED
LET ajustable2 = car2
LET ajustable3 = car3

LET sobra1 = (80-length(ajustable1))
LET sobra2 = (80-length(ajustable2))
LET sobra3 = (80-length(ajustable3))

LET ajustador1 = sobra1 SPACES
LET ajustador2 = sobra2 SPACES
LET ajustador3 = sobra3 SPACES

LET ajustado1 = ajustable1,ajustador1
LET ajustado2 = ajustable2,ajustador2
LET ajustado3 = ajustable3,ajustador3

PRINT COLUMN 006, r_det c_producto CLIPPED,
COLUMN 011, ajustado1,
COLUMN 003, r_det cantidad USING "####",
COLUMN 105, r_det unidad CLIPPED,
COLUMN 110, wc_dom CLIPPED, r_det precio_unit USING "###,### #5",
COLUMN 134, wc_dom CLIPPED, r_det importe USING "###,### #5"

IF ajustado2 IS NOT NULL THEN
PRINT COLUMN 011, ajustado2
LET ajustado2 = ""
END IF
IF ajustado3 IS NOT NULL THEN
PRINT COLUMN 011, ajustado3
LET ajustado3 = ""
END IF

ON LAST ROW
SKIP 3 LINES
PRINT COLUMN 125, "SUBTOTAL", wc_dom CLIPPED, rm USING "###,### #5"
PRINT COLUMN 125, "IVA", wc_dom CLIPPED, rn USING "###,### #5"
PRINT COLUMN 125, "TOTAL", wc_dom CLIPPED, ro USING "###,### #5"

END REPORT

SELECT * INTO r_prov *
FROM proveedores
WHERE c_proveedor = r_doc c_proveedor

SELECT * INTO r_bod *
FROM bodegas
WHERE no_bodega = r_doc no_bodega

PRINT @1
PRINT @a0058.1501Z
PRINT r_prov nombre
PRINT @a0102.1300Z
PRINT "DIRECCION", r_prov calle CLIPPED, " No", r_prov numero CLIPPED, "
COLONIA", r_prov colonia
PRINT @a140.1300Z
PRINT "CIUDAD", r_prov ciudad CLIPPED, " C.P.", r_prov cp CLIPPED " TEL 1",
r_prov tel1 CLIPPED " TEL 2", r_prov tel2 CLIPPED, " FAX", r_prov fax
PRINT @a200.1300Z
PRINT "CONTACTO", r_prov contacto CLIPPED, " AREA", r_prov con_area
PRINT @a400.1255Z
PRINT r_doc transporte
PRINT @a400.1810Z
PRINT r_bod direccion
PRINT @a550.1300Z

```

```

(
.....
** SISTEMA FACTURACION
** PROGRAMA f7m75000 4g
** DESCRIPCION Dar mantenimiento al catalogo de Clientes generar sus
reportes correspondientes
** FUNCION Altas, Bajas, Consultas, Cambios y Reportes del Catalogo
de Clientes
** DESCRIPCION PARAMETROS
** gc_user # Clave del Usuario
** ga_ident # Identificador del Usuario
** gc_idtty # Identificador de Terminal
** gc_idpro # Tíulo del programa
** gc_benda # Nombre de la tienda
** gc_imp # Identificador Impresora
** gc_nom # Identificador del Sistema
** gc_omp # Identificador Empresa
** gc_usufln # Identificación del usuario final
** gc_tparea # Identificación del tipo de area
** gc_cvesrea # Identificador de la clave de la tienda
** gf_tec # Fecha del sistema
** gc_hor # Hora del sistema
** gc_car # Caracter para respuestas
** ga_col # Columna del título
.....
)
DATABASE factura
GLOBALS "g75000 4g"

(=====PROGRAMA PRINCIPAL=====)
MAIN

CALL start_error()
WHENEVER ERROR CALL err_handle

LET gc_user = ARG_VAL(1)
LET ga_ident = ARG_VAL(2)
LET gc_idtty = ARG_VAL(3)
LET gc_idpro = ARG_VAL(4)
LET gc_benda = ARG_VAL(5)
LET gc_imp = ARG_VAL(6)
LET gc_nom = ARG_VAL(7)
LET gc_omp = ARG_VAL(8)
LET gc_usufln = ARG_VAL(9)
LET gc_tparea = ARG_VAL(10)
LET gc_cvesrea = ARG_VAL(11)

OPTIONS ACCEPT KEY F4

DEFER INTERRUPT
DEFER QUIT

OPEN FORM f7m75000 FROM "f7m75000" # Abre la forma y la despliega
CALL dis_panta(gc_idpro, gc_idtty, gc_nom)

(=====Menu principal de mantenimiento=====)

MENU "CLIENTES"
COMMAND KEY (ESC)
EXIT MENU
COMMAND "Altas" "Registro de Clientes"
CALL alt_clientes(TRUE,gc_idpro,gc_idtty,gc_nom)
RETURNING gc_car,gr_clientes c_cliente,gr_clientes d_nombre
COMMAND "Bajas" "Baja de Clientes"
ERROR "Se encuentra desabilitada esta opción "
CALL baj_clientes()
COMMAND "Cambios" "Cambios a Clientes"
CALL mod_clientes()
COMMAND KEY("O") "Consultas" "Consulta a Clientes"
CALL con_clientes()
COMMAND "Reportes" "Reporte de Clientes"
CALL menu_imp()
END MENU
CLOSE FORM f7m75000 # Cierra la forma
END MAIN # Fin del programa f7m75000

(
.....
** FUNCION : baj_clientes
** DESCRIPCION Dar de baja los clientes registrados
.....
)
FUNCTION baj_clientes()
WHENEVER ERROR CALL err_handle

DISPLAY "" AT 1,1
DISPLAY "" AT 2,1

OPTIONS INPUT NO WRAP

CALL limpia("BAJAS FISICAS",TRUE,gc_idpro,gc_idtty,gc_nom)
***** Despliega el tipo de movimiento hecho*****
(=====Captura la clave a borrar=====)
WHILE TRUE

LET int_flag=FALSE

INITIALIZE gr_clientes * TO NULL
CLEAR FORM

INPUT BY NAME gr_clientes c_cliente
ON KEY(ESC)
LET int_flag=TRUE # Valida si se oprime esc
EXIT #INPUT # Termina la captura
ON KEY(F1)
IF #FIELD(c_cliente) THEN
CALL lista_catalogo(10,20,"clientes","c_cliente","d_nombre",TRUE)
RETURNING gc_ok,gr_clientes c_clientes,gr_clientes d_nombre
DISPLAY BY NAME gr_clientes c_cliente
END IF
AFTER FIELD c_cliente
IF gr_clientes c_cliente = 0 THEN
ERROR " Cliente ISIS no se puede dar de baja. "
NEXT FIELD c_cliente
END IF
END INPUT

IF int_flag THEN
LET int_flag=FALSE
EXIT WHILE
END IF

IF NOT valida_cliente() THEN # Valida que exista el registro
ERROR " No existe el cliente."

INITIALIZE gr_clientes * TO NULL
CLEAR FORM # Borra datos pantalla
CONTINUE WHILE
END IF # Despliega el reg. en la pantalla
DISPLAY BY NAME gr_clientes *

(=====Verifica si se borra el registro=====)
LET gc_car=""
PROMPT "Desea borrar el registro (SN) " FOR CHAR gc_car
ON KEY (ESC)
EXIT WHILE
END PROMPT

IF gc_car MATCHES "Ss" THEN # Se confirma la baja
LET gc_ok = 0
LET gc_ek = 0

SELECT COUNT(*) INTO gc_ek
FROM documentos
WHERE c_cliente = gr_clientes c_cliente

SELECT COUNT(*) INTO gc_ok
FROM contratos
WHERE c_cliente = gr_clientes c_cliente

IF gc_ek > 0 OR gc_ek = 0 THEN
ERROR " Este cliente no puede ser dado de baja ya que ",
" está ligado a una factura o contrato "
CONTINUE WHILE
END IF

DELETE FROM clientes # Borra el registro de la tabla
WHERE c_cliente = gr_clientes c_cliente
ERROR " Registro borrado."

END IF
END WHILE
CLEAR FORM # Borra los datos de la pantalla
CALL limpia("FALSE,gc_idpro,gc_idtty,gc_nom) # Limpia avisos de pantalla
END FUNCTION # Fin de baj_clientes

(
.....
** FUNCION mod_clientes
** DESCRIPCION Hacer cambios a los datos ya existentes de un cliente
.....
)
FUNCTION mod_clientes()
DEFINE ws_repres SMALLINT

WHENEVER ERROR CALL err_handle

OPTIONS INPUT WRAP

CALL limpia("CAMBIOS",TRUE,gc_idpro,gc_idtty,gc_nom) # Despliega tipo de
# movimiento

LET ws_repres=0
WHILE TRUE
DISPLAY "" AT 1,1
DISPLAY "" AT 2,1

LET int_flag=FALSE

INITIALIZE gr_clientes * TO NULL

```

```

CLEAR FORM

INPUT BY NAME gr_clientes c_cliente
ON KEY(ESC) # Captura registro a modificar
LET int_flag=TRUE # Valida si se oprime esc
EXIT INPUT
ON KEY(F1)
IF INFIELD(c_cliente) THEN
CALL lista_catalogo(10,20 "clientes",c_cliente,"d_nombre",TRUE)
RETURNING gs_ok,gr_clientes c_cliente,gr_clientes d_nombre
DISPLAY BY NAME gr_clientes c_cliente
END IF
AFTER FIELD c_cliente
IF NOT valida_cliente() THEN
ERROR " " # No existe el cliente.
" "
INITIALIZE gr_clientes TO NULL
CLEAR FORM # Borra datos pantalla
NEXT FIELD c_cliente
END IF
DISPLAY BY NAME gr_clientes *
END INPUT

IF int_flag THEN # Valida si se interrumpe el cambio
LET int_flag=FALSE
EXIT WHILE
END IF

INPUT BY NAME gr_clientes d_nombre THRU gr_clientes d_tel2
WITHOUT DEFAULTS
ON KEY(ESC) # Captura registro a modificar
LET int_flag=TRUE # Valida si se oprime esc
EXIT INPUT
AFTER FIELD d_represent
IF gr_clientes d_represent = "" THEN
# Carga los valores del cliente en los del representante
# en caso de que sea el mismo
LET gr_clientes d_represent=gr_clientes d_nombre
LET gr_clientes d_calle2=gr_clientes d_calle
LET gr_clientes d_numero2=gr_clientes d_numero
LET gr_clientes d_colonia2=gr_clientes d_colonia
LET gr_clientes d_ciudad2=gr_clientes d_ciudad
LET gr_clientes d_estado2=gr_clientes d_estado
LET gr_clientes d_cp2=gr_clientes d_cp
LET gr_clientes d_tel2=gr_clientes d_tel
DISPLAY BY NAME gr_clientes d_calle2,gr_clientes d_numero2,
gr_clientes d_colonia2,gr_clientes d_ciudad2,
gr_clientes d_estado2,gr_clientes d_cp2,
gr_clientes d_tel2
NEXT FIELD d_contacto
ELSE
INITIALIZE gr_clientes d_calle2,gr_clientes d_numero2,
gr_clientes d_colonia2,gr_clientes d_ciudad2,
gr_clientes d_estado2,gr_clientes d_cp2,
gr_clientes d_contacto,gr_clientes d_tel2 TO NULL
DISPLAY BY NAME gr_clientes d_calle2,gr_clientes d_numero2,
gr_clientes d_colonia2,gr_clientes d_ciudad2,
gr_clientes d_estado2,gr_clientes d_cp2,
gr_clientes d_contacto,gr_clientes d_tel2
END IF
END INPUT

IF int_flag THEN # Valida si se interrumpe el cambio
LET int_flag=FALSE
CONTINUE WHILE
END IF

(=====Actualiza la tabla con los datos cambiados=====)
UPDATE clientes
SET clientes * = gr_clientes *
WHERE c_cliente = gr_clientes c_cliente
CLEAR FORM # Borra los datos de la forma

(=====Mensaje registro modificado=====)
ERROR " " # Registro modificado.
" "

END WHILE

CLEAR FORM # Borra los datos de la forma
CALL limpia(" ",FALSE,gc_idpro,gc_idtty,gc_nom) # Limpia mensajes de la
pantalla
END FUNCTION # Fin de mod_tip_ser

(
=====Funcion con_clientes=====
** FUNCION con_clientes
** DESCRIPCION Consultar la informacion del catalogo
=====)
FUNCTION con_clientes()
DEFINE wc_sel_cliente CHAR(350) # Almacena select
wc_con_cliente CHAR(300) # Almacena where
wc_car_cliente CHAR(1)
WHENEVER ERROR CALL err_handle

DISPLAY " AT 1.1
DISPLAY " AT 2.1

LET int_flag=0

CALL limpia("CONSULTAS DINAMICAS",TRUE,gc_idpro,gc_idtty,gc_nom) #
Despliega tipo de movimiento

(=====Construye criterio de seleccion=====)
CONSTRUCT BY NAME wc_con_cliente ON clientes *
ON KEY(ESC) # Valida si se oprime esc
LET int_flag=TRUE
EXIT CONSTRUCT
DISPLAY " AT 23.1
DISPLAY " AT 24.1
END CONSTRUCT

IF int_flag THEN # Valida si se interrumpe el reporte
CLEAR FORM # Borra datos de la forma
CALL limpia(" ",FALSE,gc_idpro,gc_idtty,gc_nom) # Limpia mensaje de
pantalla
RETURN
END IF

LET wc_sel_cliente = "SELECT * FROM clientes "
"WHERE " ,wc_con_cliente CLIPPED, " ORDER BY c_cliente"

PREPARE w_read FROM wc_sel_cliente # Criterio seleccionado

(=====Declaro apuntador=====)
DECLARE cur_con_cliente SCROLL CURSOR FOR w_read
OPEN cur_con_cliente # Se posiciona sobre el 1er registro

FETCH FIRST cur_con_cliente INTO gr_clientes *
IF STATUS=NOTFOUND THEN # Si no lo encuentra
ERROR " " # REGISTRO NO LOCALIZADO.
" "
CLEAR FORM
CALL limpia(" ",FALSE,gc_idpro,gc_idtty,gc_nom)
RETURN # Regresa al menu de consulta
END IF # Muestra el primer registro
DISPLAY BY NAME gr_clientes *

(=====Menu de consultas=====)
MENU "CONSULTA"
COMMAND "Prmer reg" "Consulta el primer registro"
FETCH FIRST cur_con_cliente INTO gr_clientes *
DISPLAY BY NAME gr_clientes * # Muestra el 1er Reg
NEXT OPTION "Siguiente"
COMMAND "Siguiente" "Consulta el siguiente registro"
FETCH NEXT cur_con_cliente INTO gr_clientes *
IF STATUS=NOTFOUND THEN # Si no lo encuentra
ERROR " " # NO HAY MAS REGISTROS.
" "
END IF
DISPLAY BY NAME gr_clientes * # Muestra el sig registro
COMMAND "Anterior" "Consulta el registro previo"
FETCH PRIOR cur_con_cliente INTO gr_clientes *
IF STATUS=NOTFOUND THEN # Si no lo encuentra
ERROR " " # NO HAY MAS REGISTROS.
" "
END IF
DISPLAY BY NAME gr_clientes * # Muestra el reg Anterior
COMMAND "Ultimo" "Consulta el ultimo registro"
FETCH LAST cur_con_cliente INTO gr_clientes *
DISPLAY BY NAME gr_clientes * # Muestra el ultimo registro
COMMAND KEY(ESC)
EXIT MENU
END MENU
CLEAR FORM # Borra datos de la forma
CALL limpia(" ",FALSE,gc_idpro,gc_idtty,gc_nom) # Limpia mensaje pantalla
END FUNCTION # Fin con_tip_ser

(
=====Funcion menu_imp=====
** FUNCION menu_imp
** DESCRIPCION Generar un reporte del catalogo
=====)
FUNCTION menu_imp()
DEFINE wc_pant CHAR(100),
wc_lugar CHAR(100)
WHENEVER ERROR CALL err_handle

LET int_flag=FALSE

DISPLAY " AT 1.1
DISPLAY " AT 2.1
LET gc_car=""

(=====Ruta de acceso del reporte=====)
LET gc_path = "d4p1/bd/factura/PR/factura/bt/" ,gc_idpro CLIPPED
"gc_bparea CLIPPED sprellena(gc_cvearea)

```

```

LET wc_lugar= gc_imp CLIPPED, " " gc_path # Nombre impresora
LET gc_modor="chmod 666", " " gc_path # Cambia al modo del reporte
LET wc_pant="more -f:", gc_palh CLIPPED

(=====Dispositivo al que se manda el reporte=====)
PROMPT "Desee el reporte por <P>antalla, <I>impresora " FOR CHAR gc_car
ON KEY(ESC)
LET int_flag=TRUE # Valida si se oprimio esc
END PROMPT

IF int_flag THEN # Valida si se interrumpe el reporte
LET int_flag=FALSE
RETURN
END IF

CASE
WHEN gc_car MATCHES "[Pp]" # Valida el dispositivo de impresion
CALL construye()
RUN "echo '033[73h'" # Manda un caracter de reduccion
RUN wc_pant # Manda el reporte a pantalla
RUN "echo '033[73i'" # Regresa al caracter normal
WHEN gc_car MATCHES "[I]"
CALL construye()
MESSAGE "Espere un momento.. Reporte Generandose"
RUN wc_lugar # Imprime reporte
OTHERWISE
ERROR " " gc_car, " OPCION INVALIDA "
END CASE
CALL dis_panta(gc_idpro, gc_idty, gc_nom) *****
***** Despliega la forma, fecha y hora del sistema e id de terminal y programa *****
END FUNCTION # Fin menu_imp

(
*****
** FUNCION : construye
** DESCRIPCION Construye el criterio de seleccion del reporte
** PARAMETROS : Ninguno
** VARIABLES GLOBALES AFECTADAS gr_of_tgr, gr_pob, gc_path, gc_modor
*****
)
FUNCTION construye()
WHENEVER ERROR CALL err_handle

DECLARE cur_rep CURSOR FOR # Selecciona los registros
SELECT clientes *
FROM clientes
ORDER BY c_cliente

START REPORT fhm75000 TO gc_path # Manda los datos al reporte
FOR EACH cur_rep INTO gr_clientes *
OUTPUT TO REPORT fhm75000(gr_clientes *)
END FOR EACH
FINISH REPORT fhm75000 # Fin reporte
RUN gc_modor # Cambiar permisos al archivo
END FUNCTION # Fin construye

(
*****
** FUNCION : REPORT fhm75000
** DESCRIPCION Formatea la salida del reporte
** PARAMETROS : Registro de Clientes
** VARIABLES GLOBALES AFECTADAS gs_col, gc_emp, gc_area, gc_nom, gc_idpro
*****
)
REPORT fhm75000(wr_cliente)

(=====Defina las variables a usar=====)
DEFINE wr_cliente RECORD LIKE clientes,
wc_bt1 CHAR(44),
wc_bt2 CHAR(35),
wt_count INTEGER, # Para desplegar en el reporte el Total de Regs
ws_col1,
ws_coln,
ws_col2 SMALLINT

OUTPUT
LEFT MARGIN 0
RIGHT MARGIN 0
FORMAT
PAGE HEADER
LET wc_bt1 = "CATALOGO DE CLIENTES"
LET gs_col = (132-LENGTH(gc_nom))/2 # Centra los titulos
LET ws_coln = (132-LENGTH(gc_emp))/2
LET ws_col2 = (132-LENGTH(wc_bt1))/2
PRINT ASCII 15
PRINT COLUMN 001, gc_idpro,
COLUMN ws_coln, gc_emp CLIPPED,
COLUMN 116, "FECHA : ", TODAY USING "dd/mm/yy"
PRINT COLUMN gs_col, gc_nom CLIPPED,
COLUMN 116, "HORA : ", TIME
PRINT COLUMN 116, "PAGINA : ", PAGENO USING "#####"
PRINT COLUMN ws_col2, wc_bt1 CLIPPED
PRINT COLUMN 001, replicate("-", 132)

PRINT COLUMN 001, "Cve.",
COLUMN 011, "Nombre",
COLUMN 047, "Domicilio",
COLUMN 080, "Telefono",
COLUMN 104, "Representante"
PRINT COLUMN 011, "R F C",
COLUMN 106, "Contacto"
PRINT COLUMN 001, replicate("-", 132)

ON EVERY ROW
NEED 5 LINES
PRINT COLUMN 001, wr_cliente c_cliente USING "#####",
COLUMN 005, wr_cliente d_nombre,
COLUMN 038, wr_cliente d_calle CLIPPED, " # ",
wr_cliente d_numero CLIPPED,
COLUMN 080, wr_cliente d_tel,
COLUMN 104, wr_cliente d_represent
PRINT COLUMN 005, wr_cliente d_rfc,
COLUMN 038, "Col. ", wr_cliente d_colonia CLIPPED,
COLUMN 080, wr_cliente d_tel2,
COLUMN 104, wr_cliente d_contacto
PRINT COLUMN 038, wr_cliente d_ciudad CLIPPED, " ",
wr_cliente d_estado CLIPPED, " C P. ",
wr_cliente d_cp USING "&&&&"
SKIP 1 LINE

ON LAST ROW
LET wt_count = COUNT(*)
SKIP 1 LINE
PRINT COLUMN 001, wt_count USING "Total de Clientes : #####"
SKIP 2 LINE
PRINT COLUMN 005, "FIN DE REPORTE"
END REPORT # Fin report fhm75000

```

```

#####
# PROGRAMA SPMAS000 4GL
# OBJETIVO Dar mantenimiento al archivo de vistas del sistema por usuario
# y programa
# FUNCION : Altas, bajas del archivo de vistas
# MODIFICACIONES
# 1) Adaptacion del programa de acuerdo a los estandares de
# Telefonos de Mexico, la Creación de un reporte (General) de perfiles e
# Implementar la opción de COPY de perfiles
# 2) Se le agrago el tipo de programa en la pantalla, para que se
# sepa el tipo de programa que se trate MENU o es EJECUTABLE, se anexo criterio
# de busqueda en el reporte para que se pueda generar un reporte por usuario,
# por programa o por tipo de programa segun se elija
#####
DATABASE factura

GLOBALS
DEFINE
  gr_prog RECORD LIKE inventario_progs * #Buffer para la tabla de progras
  gr_vista RECORD LIKE perfiles * #Buffer para la tabla de vistas
  gr_usuar RECORD LIKE sys_users * #Buffer para la tabla de usuario
  ga_panta RECORD #Buffer para la pantalla
  c_sysuser LIKE sys_users c_sysuser,
  n_programa LIKE inventario_progs n_programa,
  d_titulo_prog LIKE inventario_progs d_titulo_prog,
  c_tipo LIKE inventario_progs c_tipo
END RECORD,
ga_cont,
gs_sec SMALLINT,
ga_modulo ARRAY [14] OF LIKE inventario_progs n_programa,
ga_tipo ARRAY [14] OF LIKE inventario_progs c_tipo,
gc_u CHAR(1), #Variable de uso general
ga_centro SMALLINT, #Columna para centrar el titulo del sistema
gc_nom_sis CHAR(132), #Nombre del sistema
gc_user CHAR(7), #Clave del usuario
ga_identi SMALLINT, #Identificador del usuario
gc_idty CHAR(8), #Identificador de la terminal
gc_idpro CHAR(8), #Identificador de programa
gc_ofc CHAR(8), #Identificador de la oficina
gc_nomemp LIKE parametros d_parametro, #Identificador del nombre empl
gc_usrfin LIKE parametros d_parametro, #Identificador del usuario final
gc_tipo_area LIKE sys_users c_tipo_area, #Identificador del tipo de area
gc_impresora LIKE sys_users c_impresora, #Identificador del tipo de area
gc_area LIKE sys_users c_area #Identificador del area

END GLOBALS
(===== Programa principal =====)
MAIN
  DEFINE wc_btempo CHAR(15)
  DEFINE wf_fecha DATE #Fecha del sistema
  (Obtiene los parametros de control y auditoria. Clave de usuario, identi-
  ficador de usuario, identificador de la terminal, identificador del prog:-)
  (ma y el nombre del sistema )
  CALL start_error()
  LET gc_user = ARG_VAL(1)
  LET gs_identi = ARG_VAL(2)
  LET gc_idty = ARG_VAL(3)
  LET gc_idpro = ARG_VAL(4)
  LET gc_ofc = ARG_VAL(5)
  LET gc_impresora = ARG_VAL(6)
  LET gc_nom_sis = ARG_VAL(7)
  LET gc_nomemp = ARG_VAL(8)
  LET gc_usrfin = ARG_VAL(9)
  LET gc_tipo_area = ARG_VAL(10)
  LET gc_area = ARG_VAL(11)
  DEFER INTERRUPT
  DEFER QUIT
  OPTIONS ACCEPT KEY F4
  (===== Abre y despliega el formato de mantenimiento =====)
  OPEN FORM forma01 FROM "spma5000"
  OPEN FORM forma02 FROM "spma5100"
  DISPLAY FORM forma01
  INITIALIZE ga_panta * TO NULL
  DISPLAY gc_idpro AT 3.1
  LET gs_centro = (80 - LENGTH(gc_nom_sis))/2
  LET wf_fecha = TODAY
  DISPLAY gc_nom_sis CLIPPED AT 3.gs_centro
  DISPLAY wf_fecha USING "dd-mm-yy" AT 3.72
  LET wc_btempo = TIME
  DISPLAY wc_btempo AT 4.72
  DISPLAY gc_idty AT 4.1
  (===== Se construye menu para mantenimiento =====)
  MENU "VISTAS"
  COMMAND "Altas" "Asignacion de Nuevos Programas"
  CALL altas()
  COMMAND "Bajas" "Baja de Programas"
  CALL bajas()
  COMMAND "Copia" "Copia una Vista de un usuario a otro"
  CALL copiar()
  COMMAND "Reportes" "Reporte General de perfiles"
  CALL reportes()

COMMAND KEY (F1)
  CALL comando()
COMMAND KEY (ESC)
  CLEAR SCREEN
  EXIT MENU
END MENU
END MAIN
(===== Fin del programa principal =====)

#####
# DESCRIPCION DE PARAMETROS Ninguno
# MANEJO DE EXCEPCIONES : Primero se inserta en la tabla de perfiles la vista
# y luego se confirma, esto es debido a la recursividad
# que existe en el programa
# VARIABLES GLOBALES AFECTADAS 'ga_panta'
# SUPOSICIONES Ninguna
# FUNCION altas
# OBJETIVO : Dar de alta un nuevo registro en la tabla de vistas
#####
FUNCTION altas()
  DEFINE ws_ok SMALLINT
  (===== Proceso interactivo para dar de alta registros =====)
  CALL limpia("ALTAS:1") **Limpia y Despliega el tipo de Movimiento hecho
  INITIALIZE ga_panta * TO NULL
  WHILE TRUE
    INPUT BY NAME ga_panta * WITHOUT DEFAULTS
    AFTER FIELD c_sysuser #Despues de teclear al usuario
      IF NOT valida_usu() THEN #Valida que no exista el registro
        ERROR * La clave del usuario que tecleo NO EXISTE
        SLEEP 1
        CLEAR FORM
        NEXT FIELD c_sysuser
      END IF
    AFTER FIELD n_programa
      DISPLAY "" AT 23.1
      IF ga_panta n_programa IS NULL THEN
        NEXT FIELD n_programa
      ELSE
        IF NOT valida_prog() THEN #Valida que no exista el registro
          ERROR * El Programa que desea dar de alta, NO EXISTE en catalogo
          SLEEP 1
          NEXT FIELD n_programa
        END IF
        LET ga_panta d_titulo_prog = gr_prog d_titulo_prog,
        gr_prog c_nombre_fisico
        LET ga_panta c_tipo = gr_prog c_tipo # asigna tipo programa
        DISPLAY BY NAME ga_panta d_titulo_prog ga_panta c_tipo
      END IF
    BEFORE FIELD c_tipo # no captura tipo prog
      NEXT FIELD c_sysuser
    ON KEY (ESC)
      LET int_flag = TRUE
      EXIT INPUT
    ON KEY (F1)
      IF INFIELD (n_programa) THEN
        CALL catalogo_lista(10,20,"inventario_progs",n_programa,
        d_titulo_prog,"c_tipo",TRUE)
        RETURNING ws_ok ga_panta n_programa ga_panta d_titulo_prog,
        ga_panta c_tipo
        DISPLAY BY NAME ga_panta n_programa,
        ga_panta d_titulo_prog,
        ga_panta c_tipo
      END IF
    END INPUT
    (===== Valida si desea abortar el proceso de alta =====)
    IF int_flag THEN
      LET int_flag = FALSE
      EXIT WHILE
    END IF
    OPEN WINDOW w1 AT 8.2 WITH 16 ROWS, 78 COLUMNS
    ATTRIBUTES(MESSAGE LINE LAST,BORDER)
    DISPLAY FORM forma02
    CALL inserta_prog()
    INITIALIZE ga_panta n_programa, ga_panta d_titulo_prog, # limpia campos
    ga_panta c_tipo TO NULL
    CLOSE WINDOW w1
  END WHILE
  CALL limpia(" ")
  CLEAR FORM
END FUNCTION

```

```

#####
# DESCRIPCION DE PARAMETROS: Ninguno
# MANEJO DE EXCEPCIONES: Ninguno
# VARIABLES GLOBALES AFECTADAS: ga_panta *
# SUPOSICIONES: Ninguna
# FUNCION: bajas
# OBJETIVO: Del de baja un nuevo registro o programa de un menu de la tabla
# de vistas
#####
FUNCTION bajas()
  CALL limpia("BAJAS",1)

  (=ABRE UNA VENTANA Y DESPLIEGA UN AVISO DE OPERACION DE LAS BAJAS
  *)
  OPEN WINDOW w2 AT 11,20 WITH 10 ROWS, 42 COLUMNS
  ATTRIBUTES(PROMPT LINE LAST,BORDER)
  DISPLAY " AVISO " AT 2,7 ATTRIBUTE (REVERSE)
  DISPLAY "Si desea borrar un menu, este" AT 4,7
  DISPLAY "debera estar vacio previamente" AT 5,7
  DISPLAY "Es decir, primeramente borra" AT 6,7
  DISPLAY "los programas ejecutables que" AT 7,7
  DISPLAY "pertenezen al menu " AT 8,7
  PROMPT " Oprima cualquier tecla para continuar " FOR CHAR gc_
  ON KEY (ESC)
    LET int_flag = TRUE
  END PROMPT

  CLOSE WINDOW w2

  IF int_flag THEN
    LET int_flag = FALSE
    CALL limpia("",0)
    RETURN
  END IF

  INITIALIZE ga_panta * TO NULL

  WHILE TRUE
    (===== Proceso interactivo para dar de baja registros =====)
    INPUT BY NAME ga_panta c_sysuser WITHOUT DEFAULTS

    AFTER FIELD c_sysuser #Despues de teclear el usuario
    IF NOT valida_usu() THEN #Valida que no exista el registro
      ERROR " La clave del usuario que tecleo NO EXISTE "
      SLEEP 1
      CLEAR FORM
      NEXT FIELD c_sysuser
    END IF

    ON KEY (ESC)
      LET int_flag = TRUE
      EXIT INPUT

    END INPUT

    (===== Valida si desea abortar el proceso de baja =====)
    IF int_flag THEN
      LET int_flag = FALSE
      EXIT WHILE
    END IF

    OPEN WINDOW w1 AT 8,2 WITH 16 ROWS, 76 COLUMNS
    ATTRIBUTES(MESSAGE LINE LAST,BORDER)
    DISPLAY FORM forma02
    CALL borra_prog(0)
    CLOSE WINDOW w1
  END WHILE
  CLEAR FORM
  CALL limpia("",0)
END FUNCTION

```

```

#####
# FUNCION copiadot
# DESCRIPCION DE PARAMETROS: Ninguno
# MANEJO DE EXCEPCIONES: Ninguno
# VARIABLES GLOBALES AFECTADAS:
# SUPOSICIONES: Ninguna
# OBJETIVO: Copia la vista final de un usuario a otro
#####
FUNCTION copiadot()

```

```

  DEFINE wc_sysusers LIKE sys_users c_sysuser,
  wc_sysuserb LIKE sys_users c_sysuser

```

```

  OPTIONS INPUT WRAP,
  PROMPT LINE 14

```

```

  CALL limpia("COPIA PERFILES",1)

```

```

  INITIALIZE ga_panta * TO NULL

```

```

  WHILE TRUE
    (===== Proceso interactivo para copiar perfiles =====)
    INPUT BY NAME ga_panta c_sysuser

```

```

    BEFORE FIELD c_sysuser

```

```

  DISPLAY "Teclee el Usuario origen " AT 10,1

```

```

  AFTER FIELD c_sysuser #Despues de teclear el usuario
  IF NOT valida_usu() THEN #Valida que no exista el registro
    ERROR " La clave del usuario que tecleo NO EXISTE "
    SLEEP 1
    CLEAR FORM
    NEXT FIELD c_sysuser
  END IF
  LET wc_sysusera = ga_panta c_sysuser

```

```

  IF cuenta(wc_sysusera) = 0 THEN
    ERROR "ERROR " Este Usuario no tiene definida ninguna vista o perfil "
    NEXT FIELD c_sysuser
  END IF
  DISPLAY wc_sysusera AT 10,32 attribute(reverse)

```

```

  ON KEY (ESC)
    LET int_flag = TRUE
    EXIT INPUT

```

```

  END INPUT

```

```

  (===== Valida si desea abortar el proceso de copia =====)
  IF int_flag THEN
    LET int_flag = FALSE
    EXIT WHILE
  END IF

```

```

  (===== lee el usuario receptor =====)
  INPUT BY NAME ga_panta c_sysuser

```

```

  BEFORE FIELD c_sysuser
  DISPLAY "Teclee el Usuario receptor ..." AT 12,1

```

```

  AFTER FIELD c_sysuser #Despues de teclear el usuario

```

```

  IF wc_sysusera = ga_panta c_sysuser THEN
    ERROR "ERROR " El Usuario Receptor no puede ser igual al Usuario
  Origen "

```

```

    SLEEP 1
    CLEAR FORM
    NEXT FIELD c_sysuser
  END IF

```

```

  IF NOT valida_usu() THEN #Valida que no exista el registro
    ERROR " La clave del usuario que tecleo NO EXISTE "
    SLEEP 1
    CLEAR FORM
    NEXT FIELD c_sysuser
  END IF

```

```

  LET wc_sysuserb = ga_panta c_sysuser
  IF cuenta(wc_sysuserb) > 0 THEN
    PROMPT "El usuario tiene ya una vista definida, desea", " sustituirla SN "
    FOR CHAR gc_x

```

```

      ON KEY (ESC)
        LET int_flag = TRUE
      END PROMPT
      IF int_flag THEN
        LET int_flag = FALSE
        EXIT WHILE
      END IF
      IF gc_x MATCHES "[sS]" THEN
        EXIT INPUT
      END IF
    NEXT FIELD c_sysuser
  END IF
  DISPLAY wc_sysuserb AT 12,32 attribute(reverse)

```

```

  ON KEY (ESC)
    LET int_flag = TRUE
    EXIT INPUT

```

```

  END INPUT

```

```

  (===== Valida si desea abortar el proceso de copia =====)
  IF int_flag THEN
    LET int_flag = FALSE
    EXIT WHILE
  END IF

```

```

  DISPLAY "Espere un momento, estoy copiando el perfil ..." AT 18,1

```

```

  SLEEP 1
  CALL copia_vista(wc_sysusera wc_sysuserb)
  DISPLAY " AT 18,1
  ERROR " -----> PERFIL COPIADO <----- "

```

```

  SLEEP 2
  DISPLAY "" AT 10,1
  DISPLAY "" AT 12,1
  END WHILE
  CLEAR FORM
  CALL limpia("",0)

```

```

  OPTIONS INPUT NO WRAP,
  PROMPT LINE FIRST

```

```

  END FUNCTION

```

```

#####
# DESCRIPCION DE PARAMETROS : Ninguno
# MANEJO DE EXCEPCIONES : Ninguno
# VARIABLES GLOBALES AFECTADAS : gr_usuar *
# SUPOSICIONES : Ninguna
# FUNCION : Valida_usu
# OBJETIVO : Validar que exista o no un registro de la tabla de usuarios
#####
FUNCTION valida_usu()

```

```

DECLARE cur_valusvano SCROLL CURSOR FOR
SELECT *
FROM sys_users
WHERE c_sysuser = ga_panta_c_sysuser
OPEN cur_valusvano
FETCH FIRST cur_valusvano INTO gr_usuar *
IF status = NOTFOUND THEN
CLOSE cur_valusvano
RETURN 0 #No encontro el registro
END IF
CLOSE cur_valusvano
RETURN 1 #Si encontro el registro
END FUNCTION

```

```

#####
# FUNCION : cuenta
# OBJETIVO : verifica si tiene una vista o perfil un usuario dado
# PARAMETROS De entrada : Usuario
# salida : Numero de vistas
#####
FUNCTION cuenta(wc_sysuser)
DEFINE wc_sysuser LIKE sys_users c_sysuser,
ws_numreg SMALLINT
LET ws_numreg = 0
SELECT COUNT(*) INTO ws_numreg FROM perfiles
WHERE c_sysuser = wc_sysuser
RETURN ws_numreg
END FUNCTION

```

```

#####
# FUNCION : copia_vista
# OBJETIVO : lee la vista del usuario A y se la copia al usuario B dados como
# parametros
# PARAMETROS De entrada : Usuario A, Usuario B
#####
FUNCTION copia_vista(wc_sysusera,wc_sysuserb)
DEFINE wc_sysusera LIKE sys_users c_sysuser,
wc_sysuserb LIKE sys_users c_sysuser,
wr_perfilA RECORD LIKE perfiles *,
wr_perfilB RECORD LIKE perfiles *
DELETE FROM perfiles
WHERE c_sysuser = wc_sysuserb
DECLARE apun01 CURSOR FOR
SELECT * INTO wr_perfilA * FROM perfiles
WHERE c_sysuser = wc_sysusera
FOREACH apun01
LET wr_perfilB * = wr_perfilA *
LET wr_perfilB c_sysuser = wc_sysuserb
INSERT INTO perfiles VALUES(wr_perfilB *)
END FOREACH
END FUNCTION

```

```

#####
# DESCRIPCION DE PARAMETROS : Ninguno
# MANEJO DE EXCEPCIONES : Ninguno
# VARIABLES GLOBALES AFECTADAS : gr_progr *
# SUPOSICIONES : Ninguna
# FUNCION : Valida_pro
# OBJETIVO : Validar que exista o no un registro de la tabla de programas
#####
FUNCTION valida_pro()
SELECT *
INTO gr_progr *
FROM inventario_progs #Lee un solo registro
WHERE n_programa = ga_panta_n_programa
IF status = NOTFOUND THEN
RETURN 0 #No encontro el registro
END IF
RETURN 1 #del último leído
END IF
RETURN 1 #Si encontro el registro
END FUNCTION

```

```

#####
# DESCRIPCION DE PARAMETROS : Ninguno
# MANEJO DE EXCEPCIONES : Ninguno
# VARIABLES GLOBALES AFECTADAS : gr_progr *

```

```

# SUPOSICIONES : Ninguna
# FUNCION : Comando
# OBJETIVO : Ejecutar un comando del sistema operativo, sin salir de la
# aplicacion
#####
FUNCTION comando()
DEFINE w_comando CHAR(80) #Aqui se deposita el comando a ejecutar
CLEAR SCREEN
LET gc_x = "1"
WHILE (gc_x = "1")
PROMPT "w" FOR w_comando
RUN w_comando
PROMPT "Teclee RETURN para continuar" FOR CHAR gc_x
END WHILE
CLEAR SCREEN
DISPLAY FORM forma01
END FUNCTION

```

```

#####
# DESCRIPCION DE PARAMETROS : pi_path, que indica el nivel de profundidad
# MANEJO DE EXCEPCIONES : Ninguno
# VARIABLES GLOBALES AFECTADAS : gs_sec,gc_x
# SUPOSICIONES : Ninguna
# FUNCION : Inserta_prog
# OBJETIVO : Desplegar a través de una ventana la vista de un nivel dado
# e inserta un nuevo programa en una vista dada
#####
FUNCTION inserta_prog(pi_path)
DEFINE pi_path INTEGER,
ws_existe SMALLINT

```

```

(===== Despliega la vista o menu segun el nivel dado =====)
LET gs_sec = 0
CALL display_vista(pi_path)
(= Proceso interactivo para solicitar la posicion del programa en el menu =)
INPUT BY NAME gs_sec
BEFORE FIELD gs_sec
CALL valida_programa() RETURNING ws_existe
IF ws_existe THEN
LET int_flag = TRUE
EXIT INPUT
END IF
AFTER FIELD gs_sec
IF gs_sec IS NULL THEN
NEXT FIELD gs_sec
ELSE
IF (gs_tipo[gs_sec] = "E") THEN
ERROR * *** ERROR *** No se puede dar de Alta sobre otro Ejecutable *
NEXT FIELD gs_sec
END IF
END IF
ON KEY (ESC)
LET int_flag = TRUE
EXIT INPUT

```

```

END INPUT
(===== Valida si desea abortar el proceso de alta =====)
IF int_flag THEN
LET int_flag = FALSE
RETURN
END IF

```

```

(=== Desea dar de alta el programa en un nivel mas abajo ===)
IF (gs_tipo[gs_sec] = "M") THEN
LET pi_path = pi_path * 100 + gs_sec
CALL inserta_prog(pi_path)
RETURN
END IF
IF (gs_sec+1) > gr_vista_n_sec THEN
LET gs_sec = gr_vista_n_sec + 1 #Aplica el sig. Consecutivo
END IF

```

```

INSERT INTO perfiles
VALUES (ga_panta_c_sysuser pi_path gs_sec ga_panta_n_programa)
CALL display_vista(pi_path)
PROMPT "Confirma el movimiento ? (S/N)" FOR CHAR gc_x
(===== Inserta el nuevo registro en la tabla de vistas =====)
IF gc_x NOT MATCHES "S" THEN
DELETE FROM perfiles
WHERE c_sysuser = ga_panta_c_sysuser
AND n_path = pi_path
AND n_sec = gs_sec
MESSAGE " " Movimiento Abortado

```



```

END IF
END FOR
RETURN wf_existe_p
END FUNCION

#####
# DESCRIPCION DE PARAMETROS pi_path, nivel de profundidad
# MANEJO DE EXCEPCIONES Ninguno
# VARIABLES GLOBALES AFECTADAS gr_vista
# SUPOSICIONES Ninguna
# FUNCION Menu_vacio
# OBJETIVO Valida si el menu seleccionado esta vacio o no
#####
FUNCTION menu_vacio(pi_path)

DEFINE pi_path INTEGER
DECLARE cur_perfil SCROLL CURSOR FOR
SELECT *
FROM perfiles
WHERE c_sysuser = gr_usrar c_sysuser
AND n_path = pi_path

OPEN cur_perfil
FETCH FIRST cur_perfil INTO gr_vista *
IF status = NOTFOUND THEN
CLOSE cur_perfil
RETURN 0 #No encontro el registro
END IF
CLOSE cur_perfil
RETURN 1 #Si encontro el registro

END FUNCION

#####
# FUNCION : REPORTES
# OBJETIVO INICIALIZAR EL REPORTE YA SEA EN PANTALLA, IMPRESORA O UI
# ARCHIVO EN DISCO
#####
FUNCTION reportes()

DEFINE
wc_path CHAR(50), # Ruta donde se almacena el reporte
wc_crit CHAR(500), # Criterio de busqueda
wc_select CHAR(500), # Criterio del select
wc_proc CHAR(100), # Manda reporte al dispositivo escogido
wr_vista RECORD LIKE perfiles *, # registro para reporte
wc_nombre LIKE inventario_progs d_titulo_prog,
wc_tipo LIKE inventario_progs c_tipo,
ws_ok, # Bandera todo ok
ws_b SMALLINT # Lugar donde manda reporte

OPTIONS
ACCEPT KEY F4
INPUT WRAP # esta opcion es para que cursor regrese a home
# despues del ultimo campo de la pantalla

CALL limpia("REPORTES",1)

LET wc_path = "d4p1\bd\factura\PR\factura\b4\gc_dprio CLIPPED.*",
gc_tipo_area CLIPPED, spirelana(gc_area)

PROMPT "Desea el reporte por <P>antalla o <I>mpresora "
FOR CHAR gc_x
ON KEY (ESC)
LET int_flag = TRUE
END PROMPT
IF int_flag THEN
LET int_flag = FALSE
RETURN
END IF
LET ws_b = 1 # Lugar donde manda reporte
CASE
WHEN gc_x MATCHES "[pP]"
START REPORT rep1 TO wc_path
LET ws_b = 1 # Lugar donde manda reporte
WHEN gc_x MATCHES "[iI]"
START REPORT rep1 TO wc_path # inicializa el reporte por impresora
LET ws_b = 2 # Lugar donde manda reporte
OTHERWISE
ERROR " " gc_x " OPCION INVALIDA "
CLEAR FORM
END CASE

CONSTRUCT BY NAME wc_crit
ON perfiles c_sysuser,
perfiles n_programa,
d_titulo_prog,
c_tipo

ON KEY(ESC)
LET int_flag = TRUE
EXIT CONSTRUCT

ON KEY (F1)
IF INFIELD (n_programa) THEN
CALL catalogo_lista(10,20,"inventario_progs","n_programa",
"d_titulo_prog","c_tipo",TRUE)
RETURNING ws_ok ga_panta n_programa ga_panta d_titulo_prog,
ga_panta c_tipo
DISPLAY BY NAME ga_panta n_programa,
ga_panta d_titulo_prog,
ga_panta c_tipo
END IF
END CONSTRUCT
(===== Valida si desea abortar el reporte =====)
IF int_flag THEN
LET int_flag = FALSE
CALL limpia(" ",0)
RETURN
END IF

LET wc_select = "SELECT perfiles * d_titulo_prog c_tipo ",
"FROM perfiles inventario_progs ",
"WHERE perfiles n_programa = inventario_progs n_programa ",
"AND ",
"ORDER BY c_sysuser, n_path"

PREPARE w_read FROM wc_select

# ===== Se declara un apuntador para leer los registros del =====
# ===== criterio seleccionado en el criterio anterior =====
DECLARE cur_apun2 CURSOR FOR w_read

MESSAGE "Espere un momento... Reporte Generandose"

# ===== lee reg. por reg. y los envia al formato de reporte
FOREACH cur_apun2 INTO wr_vista *,wc_nombre,wc_tipo
OUTPUT TO REPORT rep1(wr_vista *,wc_nombre,wc_tipo)
END FOREACH

CLEAR FORM
FINISH REPORT rep1

# Cambia el privilegio del archvo seleccionado en el reporte para que
# todos tengan acceso a la informacion

LET wc_proc = "chmod 777 *.wc_path CLIPPED"
RUN wc_proc

IF ws_b = 1 THEN # en caso de que se quiera el reporte por pantalla
LET wc_proc = "more -f *.wc_path CLIPPED"
RUN wc_proc
ELSE
IF ws_b = 2 THEN # en caso de que se quiera el reporte impreso
LET wc_proc = "gc_impresora *.wc_path CLIPPED"
RUN wc_proc
END IF
CALL limpia(" ",0)
(===== TERMINA EL REPORTE =====)
END FUNCION

#####
# FUNCION : REP1
# DESCRIPCION FORMATEO DE LA SALIDA DEL REPORTE
# PARAMETROS registros leidos de perfiles
# VARIABLES AFECTADAS :
#####
REPORT rep1(pr_f_pc_name,pc_tipo)

(===== se declaran las variables que se reciben por parametro =====)
DEFINE pr_f RECORD LIKE perfiles *,
pc_name LIKE inventario_progs d_titulo_prog,
pc_tipo LIKE inventario_progs c_tipo,
wc_nombre CHAR(80),
ws_impres,
ws_longitud SMALLINT

(===== Detalles del reporte elegido =====)
OL 'PUT
LEFT MARGIN 0
RIGHT MARGIN 80

FORMAT
PAGE HEADER
LET ws_longitud = (80-LENGTH(gc_nombre))/2
PRINT COLUMN 1,gc_dprio,
COLUMN ws_longitud,gc_nombre CLIPPED,
COLUMN 64,TODAY USING "FECHA dd-mm-yy"
LET ws_longitud = (80-LENGTH(gc_usfin))/2
PRINT COLUMN 1,gc_ofic,
COLUMN ws_longitud,gc_usfin CLIPPED
COLUMN 64,"HORA " TIME

LET ws_longitud = (80-LENGTH(gc_nombre))/2
PRINT COLUMN ws_longitud,gc_nombre,ws CLIPPED,
COLUMN 64,PAGENO USING "PAGINA ====="
LET wc_nombre = "REPORTE DE PERFILES"
LET ws_longitud = (80-LENGTH(wc_nombre))/2
PRINT COLUMN ws_longitud,wc_nombre CLIPPED

```

```

PRINT replicate("-",80)
PRINT COLUMN 2,"SysUser",
      COLUMN 16,"Path",
      COLUMN 27,"Programa",
      COLUMN 40,"Tipo",
      COLUMN 48,"Titulo del Programa"

PRINT replicate("-",80)

IF NOT(ws_impre) THEN          # Si cambio de sysuser
PRINT COLUMN 3,pr_f_c_sysuser, # Imprime sysuser y path
      COLUMN 10,pr_f_n_path,
END IF
LET ws_impre = FALSE          # Apaga bandera

BEFORE GROUP OF pr_f_c_sysuser # Antes de cambio sysuser
IF ws_impre THEN              # Si esta prendida bandera
SKIP 3 LINES                   # Salta 3 lines
END IF

BEFORE GROUP OF pr_f_n_path    # Antes de cambio path
IF ws_impre THEN              # Si esta prendida bandera
PRINT COLUMN 3,pr_f_c_sysuser, # Imprime sysuser y path
      COLUMN 10,pr_f_n_path,
END IF

AFTER GROUP OF pr_f_n_path     # Cambio de path
IF LINEENO <= 62 AND LINEENO <> 0 THEN # Si la linea <-> tope
SKIP 1 LINE                   # Brinca 1 linea
LET ws_impre = TRUE          # Prende bandera
END IF
IF LINEENO = 0 THEN           # Si coincide cambio path y salto
LET ws_impre = FALSE         # Apaga bandera
SKIP TO TOP OF PAGE          # Brinca pagina
END IF

ON EVERY ROW
PRINT COLUMN 26,pr_f_n_programa,
      COLUMN 42,pc_tipo,
      COLUMN 48,pc_name
LET ws_impre = FALSE          # Apaga bandera

AFTER GROUP OF pr_f_c_sysuser
NEED 4 LINES
SKIP 1 LINE
PRINT COLUMN 11,"TOTAL DE PROGRAMAS MENUS      ",
      GROUP COUNT (*) WHERE pc_tipo = "M" USING "###"
PRINT COLUMN 11,"TOTAL DE PROGRAMAS EJECUTABLES ",
      GROUP COUNT (*) WHERE pc_tipo = "E" USING "###"
PRINT COLUMN 11,"TOTAL DE PROGRAMAS PARA ESTE USUARIO ",
      GROUP COUNT (*) USING "###"
IF LINEENO <= 62 AND LINEENO <> 0 THEN # Si la linea <-> tope
SKIP 1 LINE                   # Brinca 1 linea
LET ws_impre = TRUE          # Prende bandera
END IF
IF LINEENO = 0 THEN           # Si coincide cambio sysuser y salto
LET ws_impre = FALSE         # Apaga bandera
SKIP TO TOP OF PAGE          # Brinca pagina
END IF

ON LAST ROW
SKIP 1 LINE
PRINT COLUMN 11,"TOTAL DE REGISTROS LEIDOS      ",
      COUNT(*) USING "###"
SKIP 2 LINES
PRINT COLUMN 05,"*** FIN DE REPORTE ***"
END REPORT

##### FIN DEL PROGRAMA #####

```

```

(
.....
** SISTEMA FACTURACION
** PROGRAMA spma8000 4gl
** DESCRIPCION Dar mantenimiento al catalogo de Folios
** FUNCION Solo pueden ser modificados los datos de los números de
** folios existentes en la base
** Si los números de folio fueron dados de baja fuera del
** sistema de facturación, este programa los crea en forma
** automática
** DESCRIPCION PARAMETROS
** gc_user # Clave del Usuario
** gs_ident # Identificador del Usuario
** gc_idty # Identificador de Terminal
** gc_idpro # Título del programa
** gc_benda # Nombre de la tienda
** gc_imp # Identificador Impresora
** gc_nom # Identificador del Sistema
** gc_emp # Identificador Empresa
** gc_usufin # Identificación del usuario final
** gc_bparea # Identificación del tipo de área
** gc_cvearea # Identificador de la clave de la tienda
** gf_fec # Fecha del sistema
** gc_hor # Hora del sistema
** gc_car # Caracter para respuestas
** gs_col # Columna del título
.....
)
DATABASE factura

(***** Declaración de variables globales *****)
GLOBALS
DEFINE gr_regiones RECORD LIKE regiones *, # Registro de regiones
gc_nom LIKE parametros_d_parametro, # Título del sistema
gc_ident LIKE sys_users n_id_tysuser, # Identificador del usuario
gc_imp LIKE sys_users c_impresora, # Identificador Impresora
gc_benda LIKE bendas d_benda, # Nombre de la tienda
gc_usufin LIKE parametros_d_parametro, # Usuario Final
gc_bparea LIKE sys_users c_tpo_area, # Tipo de área
gc_cvearea LIKE sys_users c_area, # Clave de la tienda
gs_folio1 LIKE folios n_folio, # Número de folio 1
gs_folio2 LIKE folios n_folio, # Número de folio 2
gs_folio3 LIKE folios n_folio, # Número de folio 3
gs_folio4 LIKE folios n_folio, # Número de folio 4
gs_folio5 LIKE folios n_folio, # Número de folio 5
gf_fec DATE, # Fecha del sistema
gc_emp CHAR(35), # Identificador Empresa
gc_hor, # Hora del sistema
gc_idpro, # Identificador programa
gc_idty CHAR(8), # Identificador de Terminal
gc_user CHAR(7), # Clave del Usuario
gc_car CHAR(1), # Caracter para respuestas
gs_ok,
gs_col SMALLINT
END GLOBALS

(***** PROGRAMA PRINCIPAL *****)
MAIN
CALL start_error()
LET gc_user = ARG_VAL(1)
LET gs_ident = ARG_VAL(2)
LET gc_idty = ARG_VAL(3)
LET gc_idpro = ARG_VAL(4)
LET gc_benda = ARG_VAL(5)
LET gc_imp = ARG_VAL(6)
LET gc_nom = ARG_VAL(7)
LET gc_emp = ARG_VAL(8)
LET gc_usufin = ARG_VAL(9)
LET gc_bparea = ARG_VAL(10)
LET gc_cvearea = ARG_VAL(11)

OPTIONS ACCEPT KEY F4

DEFER INTERRUPT
DEFER QUIT

OPEN FORM spma8000 FROM "spma8000" # Abre la forma y la despliega
CALL dis_pant()
CALL alta_folios()
CLOSE FORM spma8000 # Cierra la forma
END MAIN # Fin del programa spma8000

(
.....
** FUNCION alta_folios()
** DESCRIPCION Registrar los diferentes números de folios para cada una de
** las regiones existentes
** PARAMETROS Ninguno
.....
)
FUNCTION alta_folios()
DEFINE wa_tpo_folios ARRAY[5] OF CHAR(1)
wa_folios ARRAY[5] OF INTEGER

##WHENEVER ERROR CALL err_handle
OPTIONS INPUT WRAP
CALL limpia("CAMBIOS",TRUE) # Despliega tipo de movimiento

ERROR "Espere un momento, verificandc la existencia de los números de folio",

LET wa_tpo_folios[1]="O" # Contratos
LET wa_tpo_folios[2]="R" # Romming
LET wa_tpo_folios[3]="C" # Notas de Crédito
LET wa_tpo_folios[4]="A" # Notas de Cargo
LET wa_tpo_folios[5]="F" # Facturación

DECLARE curs_regiones CURSOR FOR
SELECT c_region
FROM regiones

FOREACH curs_regiones INTO gr_regiones c_region
FOR gs_ok = 1 TO 5
SELECT n_folio INTO gs_folio1
FROM folios
WHERE c_region = gr_regiones c_region AND
c_folio_tpo = wa_tpo_folios[gs_ok]

IF STATUS = NOTFOUND THEN
INSERT INTO folios VALUES(gr_regiones c_region,
wa_tpo_folios[gs_ok], 0)

END IF
END FOR
END FOREACH

ERROR ""

WHILE TRUE
DISPLAY "" AT 1,1
DISPLAY "" AT 2,1

LET int_flag=FALSE

INITIALIZE gr_regiones TO NULL

CLEAR FORM

INPUT BY NAME gr_regiones c_region
ON KEY(ESC) # Captura registro a modificar
(LET int_flag=TRUE # Valida si se oprime esc
EXIT INPUT
ON KEY(F1)
IF INFIELD (c_region) THEN
CALL lista_catalogo(10,20,"regiones","c_region","d_region",TRUE)
RETURNING gs_ok, gr_regiones c_region, gr_regiones d_region
DISPLAY BY NAME gr_regiones *
END IF
AFTER FIELD c_region
IF NOT valida_region() THEN
ERROR " " No existe la region",
" "
NEXT FIELD c_region
END IF
DISPLAY BY NAME gr_regiones d_region
END INPUT

IF int_flag THEN # Valida si se interrumpe el cambio
LET int_flag=FALSE
EXIT WHILE
END IF

LET wa_folios[1]=0
LET wa_folios[2]=0
LET wa_folios[3]=0
LET wa_folios[4]=0
LET wa_folios[5]=0

FOR gs_ok = 1 TO 5
SELECT n_folio INTO wa_folios[gs_ok]
FROM folios
WHERE c_region = gr_regiones c_region AND
c_folio_tpo = wa_tpo_folios[gs_ok]

END FOR

LET gs_folio1=wa_folios[1]
LET gs_folio2=wa_folios[2]
LET gs_folio3=wa_folios[3]
LET gs_folio4=wa_folios[4]
LET gs_folio5=wa_folios[5]

INPUT BY NAME gs_folio1, gs_folio2, gs_folio3, gs_folio4,
gs_folio5 WITHOUT DEFAULTS
ON KEY(ESC) # Captura registro a modificar
(LET int_flag=TRUE # Valida si se oprime esc
EXIT INPUT
END INPUT

IF int_flag THEN # Valida si se interrumpe el cambio
LET int_flag=FALSE
CONTINUE WHILE

```

```

END IF

LET wa_folios[1]=gs_folio1
LET wa_folios[2]=gs_folio2
LET wa_folios[3]=gs_folio3
LET wa_folios[4]=gs_folio4
LET wa_folios[5]=gs_folio5

FOR gs_ok = 1 TO 5
  UPDATE folios
    SET n_folio = wa_folios[gs_ok]
    WHERE c_region = gr_regiones c_region AND
          c_folio_tipo = wa_tipo_folios[gs_ok]
END FOR

CLEAR FORM          # Borra los datos de la forma

(***** Mensaje registro codificado *****)
ERROR "          Registro modificado", .
END WHILE

CLEAR FORM          # Borra los datos de la forma
CALL limpia("FALSE) # Limpia mensajes de la pantalla
END FUNCTION

```

```

{
*****
** FUNCION : valida_region()
** DESCRIPCION: Validar si se encuentro la clave de region
** PARAMETROS : Regresa Bandera verdadero o falso
*****
}
FUNCTION valida_region()

SELECT * INTO gr_regiones *      # Selecciona los registros
FROM# regiones
WHERE c_region = gr_regiones c_region

IF STATUS=NOTFOUND THEN # Valida si encuentra el registro
RETURN FALSE           # Si no lo encuentra asigna
ELSE                   # Verdadero a la bandera
RETURN TRUE            # Asigna falso a la bandera
END IF
END FUNCTION

```

CONCLUSIONES

A lo largo del capítulo se establecieron las características más relevantes de los productos INFORMIX, mismas que nos permiten ver que se trata de herramientas que proporcionan un apoyo muy sólido en el desarrollo de aplicaciones administrativas.

El proceso de codificación no deja de ser una tarea ardua y en ocasiones compleja, pero la utilización de la herramienta adecuada permite minimizar las dificultades.

No está por demás señalar el hecho de que el desempeño exitoso de la aplicación depende no solamente de la buena programación, sino de la organización física de la información, métodos de acceso a ésta, administración de la memoria y optimizaciones realizadas al manejador de bases de datos.

CAPÍTULO 6

IMPLANTACIÓN, LIBERACIÓN Y MANTENIMIENTO

El presente capítulo tratará acerca de las etapas que se llevarán a cabo en la última fase del ciclo de vida de desarrollo de sistemas; la primera de ellas corresponde a la implantación, proceso mediante el cual se hacen los preparativos para la primera operación "en vivo" del nuevo sistema, dicha etapa abarca la verificación e instalación del equipo en que funcionará la aplicación, el entrenamiento necesario del personal para que sea capaz de usar y operar el nuevo sistema, incluyendo las pruebas pertinentes para dejar al sistema libre de errores. En segundo término debe continuarse con la liberación, es decir, aquella actividad que ocurre cuando el sistema alcanza su mayor nivel de eficiencia en relación a los requerimientos que debe atender y, por lo tanto, es dejado en las manos de los usuarios finales para su operación cotidiana.

Por último, debido a que los procesos administrativos de toda organización experimentan cambios de manera continua, los sistemas de información deberán mantenerse siempre al día; motivo por el cual, es indudable que deberá darse mantenimiento a la aplicación. El mantenimiento se centrará en los cambios asociados a la corrección de errores, las modificaciones debidas a la evolución del entorno y otros dirigidos a reforzar o ampliar al sistema. En este capítulo se citan algunos casos a este respecto.

Instalación del sistema

Instalar un sistema implica hacer los preparativos para ponerlo al alcance del usuario, y son varios los pasos que corresponden a esta actividad:

- ⇒ **Copiado de la estructura de la Base de Datos al ambiente de producción**
- ⇒ **Transferencia de archivos ejecutables al ambiente de producción**

Ambientes de desarrollo y producción

Comúnmente existen dos instancias en las que reside la aplicación desde que comienza su desarrollo hasta el fin de su vida útil, éstas son: desarrollo y producción; aunque el nombre puede variar según la organización, estas denominaciones son las más indicadas debido a su orientación.

En el primer ambiente -desarrollo- se codifica la aplicación, es creada la BD en su forma más primitiva, comúnmente la estructura final de la base de datos siempre difiere de la inicial en mayor o menor escala, según la calidad del análisis y diseño. En esta BD de desarrollo se graban todo tipo de datos prueba, sin formato y sin validación que pronto se tornarán en basura que habrá que depurar constantemente.

En esta primer instancia las herramientas de desarrollo, en este caso INFORMIX, guardan una configuración específica distinta a la que debe privar en los ambientes de producción, ya que se cuenta con herramientas de compilación, depuradores y demás que no deben encontrarse disponibles en otro ambiente. En el caso de la programación en Informix-4GL, el ambiente de desarrollo consiste en una serie de menús y pantallas diseñadas para facilitar la creación de una aplicación INFORMIX. En él se pueden crear, modificar, borrar y compilar programas y formas, así como ejecutar enunciados de SQL o ligar programas de módulos múltiples

El ambiente de producción es donde reside la BD que captará la información producto de la operación real del sistema, aquella en la que el usuario estará invirtiendo su fuerza de trabajo a lo largo de la jornada laboral. Por ende, la aplicación no debe transferirse a esta instancia hasta haberse probado reiteradamente y determinar que se encuentra libre de errores.

Una vez considerado lo anterior, la instalación del sistema comienza con la creación de la BD de producción que, obviamente debe ser una copia de la última versión conseguida en el desarrollo.

Copiado de la estructura de la Base de Datos al ambiente de producción

Cuando finalmente se ha alcanzado lo que se considera la forma más estable (o menos cambiante de la BD) se procede a crear una copia vacía con herramientas como las mencionadas en el apartado de integridad del capítulo 5, véase DBschema, generando así una copia de la estructura de la BD que, además de funcionar como respaldo, permite reproducirla fácilmente en el ambiente de producción. Asimismo, se deben dar los permisos correspondientes para que sólo las cuentas de usuarios autorizados tengan privilegios de uso de esta nueva BD.

Transferencia de archivos ejecutables al ambiente de producción

Al compilar una aplicación con INFORMIX-4GL se crean dos tipos de archivos ejecutables, uno con extensión *4gi* y otro con extensión *4go*. En el programa principal de la aplicación deben existir las instrucciones *main* y *end main*, que marcan el principio y fin del bloque principal del programa, desde él deben de generarse los llamados de todos los demás programas de la aplicación, directa o indirectamente, y es este archivo, con extensión *4gi*, el que debe invocarse para correr la aplicación. Todos los demás programas, que se comportan como llamados del primero presentan extensiones *4go*. Son entonces los archivos con estos dos tipos de extensiones los que deben transferirse al ambiente de producción.

Finalización de la documentación que se ha generado a lo largo del desarrollo

La documentación es necesaria para varios propósitos. Después de la puesta en marcha del sistema, la documentación de diseño es la base para realizar cambios en el sistema. Asimismo, la buena documentación sirve para reducir los conflictos entre los usuarios y sistemas, ya que los usuarios entenderán más fácilmente un sistema bien documentado. Una buena documentación significa que se dispone de una referencia adecuada cuando surjan los problemas, y ayuda a los usuarios a aprender cómo resolverlos. En el caso del SIFACOR, esta información se condensa básicamente en los manuales de :

- ⇨ **Usuario.** Contiene lo relativo a la operación del sistema. Este tipo de información, si se consulta, minimiza la necesidad de estar comunicándose con el área de sistemas y, por lo tanto, reduce los conflictos potenciales. El manual de usuario debe tenerse listo antes de llevar a cabo la capacitación del personal operativo, ya que debe ser el documento base de la misma, hecho que facilita la actividad del instructor y también permite al usuario conocer cómo usar el manual, qué puede encontrar en él y cuándo usarlo, además de contener los antecedentes y justificación del sistema.
- ⇨ **Técnico.** El manual técnico comprende toda la documentación, diagramas e información correspondiente al desarrollo de la aplicación, a saber:
 - Objetivo del sistema
 - Diccionario de datos
 - Esquema de la base de datos
 - Diagrama de contexto
 - Diagramas de nivel
 - Tamaño de las Tablas

Tal vez el parámetro para determinar qué tanto debe contener el manual técnico es pensar que debe contener los suficientes datos como para que el sistema pueda ser reconstruido fácilmente por un grupo de trabajo totalmente ajeno a la organización.

Una vez que el sistema se haya migrado al ambiente de producción comenzará la carga de un mínimo de datos prueba lo más precisos y verídicos posible; es decir, parte de la información real, de tal modo que las pruebas resulten útiles.

Pruebas y evaluación

Las pruebas del sistema conforman la parte medular del esfuerzo técnico del proceso de desarrollo. El objetivo de las pruebas del sistema es descubrir errores de diferentes clases, haciéndolo con la menor cantidad de tiempo y esfuerzo. Como función secundaria, las pruebas demuestran hasta qué punto las funciones del software parecen estar diseñadas de acuerdo con las especificaciones y si cumplen o no los requisitos de rendimiento. Además, los datos que se van recogiendo a medida que se lleva a cabo la prueba proporcionan una buena indicación de la fiabilidad y calidad del software como un todo. En el SIFACOR se efectuaron varios tipos de pruebas:

Pruebas de control y de unidad

Se centraron básicamente en el comportamiento lógico y de funcionalidad de los módulos del programa, por lo que cada vez que se terminaba de codificar un módulo se compilaba y se realizaban corridas para verificar su comportamiento.

Pruebas de Integración

Una vez que cada módulo superaba las pruebas anteriores se procedió a realizar las pruebas de integración que consistieron en unir todos los módulos en un solo programa, y detectar así los errores asociados a la interacción de los mismos. La integración se realizó en forma ascendente, esto es, se combinaron los módulos de bajo nivel en grupos que realizaban una subfunción específica del sistema, probando y coordinando las entradas y salidas de los mismos, posteriormente se combinaron los grupos, moviéndose hacia arriba de la estructura del programa.

Prueba de Resistencia

Para el programa también se realizaron las llamadas pruebas de resistencia, que consistieron en enfrentar al programa con situaciones anormales, por ejemplo, se introdujeron valores cada vez más altos, sobre todo en los campos que registran cantidades, para determinar en que momento el programa no podrá soportar ese valor.

Estas pruebas nos permitieron detectar algunas inconsistencias. Como ejemplo mencionaremos lo ocurrido con el uso de la charola de funciones¹. Como era necesario que distintos analistas estuvieran desarrollando programas en paralelo, se acordó desde el principio cuales funciones estarían incluidas en la charola, como podían ser, la conversión de cantidad en números a letras o las ventanas de ayuda que permiten visualizar el contenido de un catálogo en forma de listado que se recorre dentro de la misma ventana. Sin embargo, las funciones no fueron documentadas adecuadamente. Por lo tanto, se detectó que algunos resultados no eran los esperados al llamar a la función, lo que requirió hacer una pausa en el desarrollo para que el grupo de trabajo quedara de acuerdo en qué tan abiertas o cerradas

¹La charola de funciones es un directorio con un path (ruta) específico desde el cuál se pueden llamar subprogramas, rutinas y funciones, los cuales reciben datos de entrada para procesarlos y devolver un resultado de salida al programa que los llamó.

debían ser estas funciones y documentarlas exhaustivamente para que no hubiera lugar a duda en cuanto a su uso y lo que se podía obtener de ellas.

Pruebas de validación

Las pruebas de validación demuestran qué tan bien se ajusta el nuevo sistema a los requisitos de los usuarios que le dieron origen.

Las pruebas de validación del SIFACOR se llevaron a cabo en varias fases. En un inicio, se efectuaron en el ambiente de desarrollo, y entre los mismos analistas que intervienen en el proyecto. Sin embargo los usuarios desempeñan un papel importante y aunque los analistas realicen muchas de las pruebas, es esencial la participación de los usuarios finales para verificar el funcionamiento correcto del sistema.

Por tanto se invitó a algunos usuarios directos a operar el sistema quienes trataron de realizar capturas de toda índole, por ejemplo, actualizar un campo inválido y utilizar códigos incorrectos de captura. A través de éstas pruebas se verificaron los errores que se presentarían durante la operación real del sistema y se aseguró que el sistema estaba programado para detectarlos de modo que no introducir información irrelevante o equívoca a las tablas que pudiera generar salidas erróneas.

El paso siguiente, una vez hechas las correcciones pertinentes fue cargar un mínimo de información útil para llevar a cabo la presentación del sistema. Para tal fin se realizó una prueba piloto con personal de TELCEL actuando como clientes y con asesores reales operando el sistema para así obtener algunos datos de desempeño en tiempo real. Con esta prueba se puso a consideración de los jefes y gerentes de las áreas que generaron los requerimientos la información plasmada en los documentos y reportes, para así determinar si debían sufrir o no alguna adecuación ó si son realmente útiles para la toma de decisiones tal y como están.

De la evaluación llevada a cabo por los jefes y gerentes de las áreas involucradas se recabó la siguiente lista de observaciones:

- ↻ El sistema deberá permitir al usuario seleccionar un número de serie específico a partir del catálogo de teléfonos.
- ↻ Sólo una persona será la encargada de imprimir tanto notas de crédito como reimpresión de facturas. Anteriormente todos los usuarios podían imprimir notas de crédito o reimprimir facturas, pero se decidió, por motivos de seguridad, controlar esta opción.
- ↻ Cuando los productos llegan a un punto mínimo, debe enviarse un reporte o mensaje a la cuenta de Almacén para que se enteren y surtan el o los productos. En un principio esta señal no se generaba a tiempo porque el límite que se había impuesto estaba muy por debajo del real.
- ↻ Checar que cuando se genere una nota de crédito por devolución de aparatos u accesorios, se de entrada al almacén. Al principio, se daba entrada al almacén a

productos que se devolvían por estar descompuestos o defectuosos cuando en realidad no debían reingresar a este inventario.

- ⇒ Pedir asignación de línea al módulo contrataciones, es decir, permitir la emisión de un contrato de línea sin la necesidad de haber facturado anteriormente un teléfono.
- ⇒ No permitir la modificación de precios unitarios en la facturación para los conceptos de teléfonos y accesorios. Este campo se protegió contra escritura y los datos sólo podían ingresar a partir del catálogo de productos y servicios.

Lo anterior no es una lista exhaustiva de los detalles que se encontraron en la presentación, pero brinda una idea de la clase de limitantes que se puntualizaron.

Depuración y ajustes

Después de recabar las observaciones surgidas en la presentación se procedió a realizar las adecuaciones en los programas, para llegar a una nueva etapa de pruebas con los usuarios directos en donde no se registraron problemas o aportaciones significativas.

Prueba de aceptación

Una vez que se verificó ante los usuarios directos y la gente de toma de decisiones que las funciones del nuevo programa estaban libres de errores en procesamiento de datos normal, se hizo un seguimiento y revisión detallada de todos los problemas anteriores para comparar como fueron resueltos. Después de esta revisión se aceptó el sistema, lo que marcó el inicio de la etapa de capacitación.

Capacitación

La capacitación para el personal que opera SIFACOR se orientó al área del usuario, se capacitó por separado al personal de las áreas de atención a clientes, contabilidad y almacén, y en un futuro se capacitará a las áreas de sistemas de cada regional, ya que ellas deberán ser capaces de administrar el sistema para los usuarios de su región. La relación del personal capacitado por cada área es la siguiente:

Atención a Clientes	Contabilidad	Almacén
20 usuarios	15 usuarios	3 usuarios

Para llevar acabo la capacitación se acondicionó un laboratorio con 5 PC'S conectadas al equipo BULL y a una de las impresoras XEROX 4213. Cuatro de los analistas participantes en el proyecto fungieron como instructores. Se planeó un periodo de una semana para capacitar a todas las áreas, previa creación de los perfiles típicos de cada área para trabajar sobre ellos al momento del curso.

El grupo de atención a clientes se dividió en dos, de forma que cada equipo de cómputo fue utilizado por dos usuarios, y se utilizaron 2 horas por la mañana para un grupo y dos horas en la tarde para otro.

El temario del curso para el área de atención a clientes contempló los siguientes tópicos:

- ⇨ Objetivo del sistema
- ⇨ Presentación
- ⇨ Descripción de las opciones disponibles para el asesor
 - Generación de facturas
 - Generación de contratos celulares
 - Generación de facturación roaming
 - Consulta de productos
 - Planes tarifarios
 - Catálogo de clientes
 - Consulta de facturación diaria
 - Reporte acumulado de ventas por asesor
- ⇨ Generación de documentos prueba
 - Facturas y Contratos
- ⇨ Descripción y uso de la impresora XEROX 4213
 - Configuración
 - Uso
 - Reemplazo de consumibles

Para el grupo de contabilidad también se dividió el grupo en dos.

El temario para el grupo de contabilidad consideró los siguientes tópicos:

- ⇨ Objetivo
- ⇨ Presentación
- ⇨ Descripción de las opciones disponibles para el personal del área
 - Generación de notas de crédito
 - Generación de notas de cargo
 - Consultas de ventas por asesor
 - Reimpresión
 - Reimpresión múltiple
- ⇨ Generación de documentos prueba
 - Notas de crédito, Notas de cargo
- ⇨ Descripción y uso de la impresora XEROX 4213
 - Configuración
 - Uso
 - Reemplazo de consumibles

Para el área de almacén fueron suficientes 5 horas de capacitación dado el reducido número de usuarios.

El temario del curso para el área de Almacén contempló los siguientes tópicos:

- ⇒ Objetivo
- ⇒ Presentación del Sistema
- ⇒ Descripción de las opciones disponibles para el personal de almacén
 - Catálogo de productos y servicios
 - Reporte de inventarios
 - Catálogo de proveedores
 - Consulta de productos
- ⇒ Descripción y uso de la lectora óptica
 - Instalación
 - Configuración
 - Uso

6.2. LIBERACIÓN

Para liberar un sistema como el SIFACOR se alcanzó un estado bastante aceptable de desempeño y calidad en los productos que genera; ya que esta actividad marca oficialmente el inicio de operaciones del sistema, lo que significó hacer una entrega formal a los usuarios, para que a partir de ese momento se hiciera uso en forma rutinaria de éste. La información que ingresa desde este momento es muy valiosa para la organización y es prácticamente imposible dar marcha atrás en el uso del sistema como herramienta cotidiana una vez concluida esta etapa.

Puesta en marcha

Una vez capacitado el personal que opera el SIFACOR se giró un comunicado en la empresa para que las áreas quedaran de acuerdo en fijar una fecha específica de inicio de operaciones del SIFACOR. Para tal fecha los equipos estuvieron dispuestos

en las áreas de los usuarios ya configurados y operando. Ese mismo día muy de mañana se inicializaron los folios a partir de los cuales comenzó la facturación, notas de crédito y de cargo que genera el sistema.

Una situación extraordinaria que privó alrededor de SIFACOR fue la imposibilidad de llevar un tiempo de operación en paralelo como comúnmente se recomienda, ya que este hecho implicaba una doble facturación, actividad que representa un ilícito, de modo tal que, las pruebas y depuraciones previas trataron de ser exhaustivas. Para ese día el líder de proyecto y algunos analistas realizaron continuos recorridos por las áreas de los usuarios para estar al tanto de cualquier contingencia.

La liberación del sistema culminó con la firma formal de un documento por las partes interesadas en el cual se especifica que el sistema está completo y les es completamente satisfactorio. Este formalismo es esencial para evitar malos entendidos posteriores.

6.3 MANTENIMIENTO

Aunque un sistema no es algo vivo, sí es un producto que debe satisfacer las necesidades de algo vivo, la organización. En el caso de las empresas que brindan servicios como TELCEL, la rapidez para adaptarse a los cambios del mercado resulta determinante, por tal motivo, es lógico suponer que de igual forma SIFACOR deberá adaptarse a estos cambios. Para adecuar dichos cambios se contempla la etapa del mantenimiento que, a diferencia de las otras etapas del ciclo de vida de desarrollo de sistemas no tiene una duración específica, ya que se presenta mientras perdure la vida útil del sistema.

El mantenimiento se enfoca principalmente en :

Mantenimiento correctivo

Viene a ser la corrección de errores detectados después de la liberación y que no se detectaron en las pruebas. En este sentido, aparecieron algunos errores o fallas, sobre todo en lo concerniente a interfaces con el usuario.

Mantenimiento adaptativo

Contribuye al mantenimiento que se produce por el rápido cambio inherente a cualquier aspecto de la informática. Regularmente, aparecen nuevos sistemas operativos o nuevas versiones de los antiguos. Se mejoran o modifican los periféricos y otros elementos de los sistemas. Por ejemplo, cualquier versión de sistema operativo UNIX que surja durante los cinco años siguientes, podrá soportar sin problemas al SIFACOR. En el caso de que surja una versión incompatible, tendrá que sopesarse la posibilidad de modificar al sistema actual para que pueda correr sobre la nueva plataforma o bien, la posibilidad de sustituir al SIFACOR por una versión diferente.

Mantenimiento perfectivo

A medida que se usa el software, se reciben recomendaciones de los usuarios sobre las nuevas posibilidades de modificaciones de funciones existentes y sobre mejoras en general. A través del uso en producción del SIFACOR, los usuarios participaron al área de sistemas de algunas modificaciones que mejorarían al mismo, tal es el caso de las que se listan a continuación:

- ↻ Al capturar un cliente nuevo podemos tener al mismo cliente como representante, tener una opción de capturar los mismos datos del cliente.
- ↻ Generar una campo más dentro de la impresión de la factura para añadir cualquier comentario adicional.
- ↻ Aumentar la ayuda en línea para el usuario.
- ↻ Poder navegar dentro de la elección de productos.
- ↻ Agregar la fecha de pago a las facturas.
- ↻ En facturación no borrar los datos capturados si no se confirma la factura permitiendo hacer correcciones.

- ⇒ Permitir modificar precios unitarios por concepto de servicios en notas de crédito y cargo.
- ⇒ Agregar leyenda 'Copia Autorizada' a la reimpresión de facturas.
- ⇒ Agregar la impresión del plan tarifario en el contrato.

Todas estas sugerencias fueron tomadas en cuenta para lograr que el sistema ofreciera el servicio que los usuarios deseaban.

Mantenimiento preventivo

El mantenimiento preventivo observa dos enfoques, el primero consiste en llevar a cabo todas aquellas actividades de depuración que aseguren que no se presentarán problemas de espacio en disco o basura en la BD, el mantenimiento a catálogos que en el caso de la aplicación debe ser mínimo, salvo para el catálogo de productos y servicios que registrará cambios diariamente ó en el caso del catálogo de usuarios dónde el área de sistemas tendrá a su cargo su actualización, realizará altas o bajas, o reasignará usuarios de un centro de atención a otro, ya que se encuentren en operación en caso de tratase de asesores de venta, o se redireccionarán los trabajos de impresión si se trata de movimientos de personal interno etc.

El segundo enfoque consiste en prever las necesidades a futuro y adecuar al sistema para que se adapte a ellas llegado el momento. Proporciona una base para mejoras a futuro. Las herramientas de ingeniería inversa y de reingeniería llegarán a ser una parte importante del proceso de mantenimiento durante este siglo. La ingeniería inversa extrae información de diseño del código fuente de un programa sin disponer de ninguna otra documentación. La reingeniería toma la información obtenida y reestructura el programa para conseguir una mayor calidad y por tanto, una mejor facilidad de mantenimiento en el futuro.

Actualizaciones

Como se mencionó durante la etapa de análisis del SIFACOR, los requerimientos proponían una aplicación substancialmente mayor que la presente; sin embargo, para resolver la necesidad más apremiante se determinó cierto grado de funcionalidad mínima suficiente para obtener un pronto desarrollo. De tal forma que liberada la primer etapa se continuará con la implementación de los requerimientos pendientes, tomándolos uno a la vez ó conformando ciertos bloques, lo que nos permitiría hablar de nuevas versiones o actualizaciones del SIFACOR.

Seguridad de la información

Se considera que la BD de SIFACOR observará un ritmo contante de crecimiento pero no tanto como para considerarla altamente transaccional². Por tal motivo el plan de respaldos diseñado llega hasta *Archive* de nivel 1³ con la siguiente frecuencia:

²Se considera una BD altamente transaccional a aquella que realiza millones de operaciones (transacciones) al día como las de los bancos u otro tipo de instituciones financieras.

³Véase capítulo 5, tema Integridad

- Viernes por la noche correr *Archive* de nivel 0
- Diariamente por la noche *Archive* de nivel 1

Por otra parte, existe otro tipo de respaldo: cada documento generado por el sistema se manda a impresión y además se guarda como un archivo texto cuyo nombre se conforma por el nombre identificador del usuario y la hora en la que se genera el documento, el sistema operativo, en este caso UNIX, asocia por default la fecha y otros parámetros para cada archivo. Estos archivos residirán en un directorio que será copiado cada que se corran los respaldos a nivel UNIX por el área de soporte técnico, una vez elaborado este respaldo se procederá a depurar el directorio en cuestión para evitar problemas de saturación.

CONCLUSIONES

Resulta de suma importancia desarrollar un plan que determine claramente el orden y tiempos de las actividades a realizar durante la etapa terminal de un proyecto de software, como es el caso de SIFACOR; ya que el grado de éxito de cada actividad en esta etapa depende en gran medida de la disponibilidad del usuario. Aunque comúnmente existe buena disposición en general para colaborar, al menos en este proyecto, para el usuario representa restar un tiempo a sus actividades cotidianas y dedicarlos a los asuntos del sistema. Por ende el personal de sistemas debe procurar ser breve y conciso, y expresar claramente la ayuda que espera recibir del usuario, para aprovechar de la mejor manera posible el tiempo de ambas partes, que es a fin de cuentas el tiempo y recursos de la organización.

CONCLUSIONES GENERALES

El proyecto SIFACOR forma parte de un proyecto mayor de la Dirección de Informática que busca la integración, robustecimiento y homogeneidad de las plataformas de software y hardware de la empresa. Por tal motivo, algunas condiciones de infraestructura ya estaban dadas al inicio de este proyecto, como lo eran los posibles equipos donde debería correr la aplicación, según la herramienta de software que se determinara utilizar, principalmente se podía elegir entre COBOL estructurado, INFORMIX, ó algún lenguaje para DOS.

La función de este equipo comenzó con la encomienda de la Dirección de Informática de atender los requerimientos e inconformidades que constantemente manifestaban determinadas áreas de la empresa, por problemas de integridad, dificultad de seguimiento, carencia de reportes automáticos, y algunos más. Cuestión que tornaba su labor difícil y con pobres rendimientos, sobre todo en lo relacionado al seguimiento y control de la facturación de teléfonos, accesorios y servicios.

Por tal motivo, se realizó una investigación preliminar que permitiera conocer y tomar en cuenta al mayor número posible de elementos (personas, documentos y procesos) que tuvieran relación directa ó indirectamente con el proceso de facturación. Una vez que se conoció con cierto detalle el proceso de facturación, se procedió a la evaluación del sistema que en ese momento se encontraba en operación, para determinar su grado de funcionalidad

y posibilidad de adaptación a los nuevos requerimientos, pero al tratarse de una aplicación comercial pronto se estimó una posibilidad nula de adaptación, sobre todo por algo que resultaba de primordial importancia: la posibilidad de que la empresa emitiera sus propios comprobantes fiscales. Se realizó entonces un análisis más profundo de los procesos tratando de llevarlo hasta las actividades más simples. Las características que comenzó a manifestar el análisis, aunado a los planes de crecimiento y estandarización de sistemas en la empresa, convirtieron a INFORMIX en la herramienta óptima de programación para el desarrollo de esta aplicación.

Con la información recabada y analizada hasta este momento, habiendo comprendido correctamente las actividades y procesos que se deberían automatizar, se elaboró un diseño básico que resolviera las necesidades más apremiantes, pero también se realizó una planeación en donde se describieron las actividades, tiempos de duración y responsables de cada una, además del costo de los elementos adicionales de *hardware* y *software*. Con este material se presentó la propuesta ante la Dirección de Informática; después de una revisión el proyecto fue aprobado, no sin antes efectuar algunas recomendaciones que simplificaran y robustecieran la administración del sistema, esto con miras al futuro implante del mismo en centros regionales.

Una vez que estuvimos todos de acuerdo en lo que se tenía que lograr, se comenzaron los trabajos de desarrollo de la aplicación, al mismo tiempo que el área de soporte técnico comenzó a trabajar en la conexión y configuración de las PC's que funcionarían en red como terminales del DPX2.

En términos generales el proyecto no sufrió grandes demoras, salvo por la dificultad de implementar alguna funcionalidad específica en la que las áreas de usuarios no llegaban a un acuerdo, como en el caso de la propuesta hecha por el área contable en el sentido de limitar el contenido de la factura a un solo teléfono y múltiples accesorios y servicios, ya que esto facilitaría su labor al momento del seguimiento de la factura y de los números de serie de los teléfonos. Hecho con el que no estaba de acuerdo el área de Atención a Clientes que solicitaba la facilidad de poder incluir más de un celular por factura, y de esta forma, si un cliente adquiría varios aparatos telefónicos, no elaborar y entregarle una factura por cada uno. Finalmente, la propuesta del área contable tuvo más peso y se tuvo que implementar esa limitante, con lo cual no se estaba de acuerdo en nuestro grupo de trabajo, ya que se considera que será una limitante importante sobre todo para las regionales que manejan atención a subdistribuidores.

Llegada la recta final del desarrollo, comenzó a prepararse la implantación de SIFACOR, se solicitó al área de soporte técnico la creación de la instancia de producción y se trabajó conjuntamente en la transferencia de la estructura, creación de permisos y configuración de INFORMIX, mientras que, por otra parte, el equipo de trabajo realizaba pruebas de desempeño con diversos lectores ópticos que permitieran la rápida captura de los números de serie de teléfonos y accesorios a su llegada al almacén. Finalmente se decidió adquirir un modelo específico del fabricante SYMBOL, que permite una conexión sencilla a la PC, por medio de un adaptador que admite el ensamble, por un lado, del teclado y, por otro, de la lectora óptica, este dispositivo a su vez se acopla a la PC en donde debe ir conectado el teclado, y no requiere instalación de software adicional.

Una vez concluido el desarrollo de este proyecto se procedió a la elaboración de la documentación faltante, como lo es el manual de instalación y configuración del sistema y el

manual de usuario; este último sirvió de base para impartir el curso de capacitación. Se estableció el temario del curso de capacitación para cada área y se elaboraron algunos exámenes que permitieron evaluar a los usuarios una vez capacitados. Para efectuar esta actividad se montó un pequeño laboratorio, y al llevarla a cabo se presentaron algunos problemas en cuanto a disponibilidad de los usuarios en los horarios específicos del curso dadas sus cargas de trabajo, lo que dificultó un poco esta labor.

Durante la realización de este proyecto nos enfrentamos a una serie de problemas y vicisitudes que van desde un pobre conocimiento, casi nulo, de algunos procesos contables en cuanto a aspectos fiscales, hasta el lidiar con usuarios que hicieron necesario hacer gala de nuestros conocimientos en relaciones humanas.

Con la culminación exitosa de este proyecto concluimos que la tarea de desarrollar sistemas de aplicación no es nada fácil, debido a que se involucran demasiados factores, pero principalmente el factor humano, es decir los usuarios que son los que se oponen con mayor fuerza a la automatización de algunos procesos, por no convenir muchas veces a sus intereses. Por eso, antes de llevar a cabo el desarrollo de un sistema de información se tienen que establecer claramente los objetivos a alcanzar, y las personas encargadas del desarrollo de la aplicación deben comprenderlos y tenerlos siempre presentes en la realización del proyecto. Esto con el fin de poder determinar de una manera más práctica, y sin que se modifiquen los objetivos del proyecto, las necesidades reales de los usuarios, pues algunas veces los usuarios no saben exactamente lo que necesitan y otras más ni siquiera expresan una necesidad. Nos dimos cuenta de que es fundamental que el líder de proyecto y algún analista mantengan comunicación constante con los usuarios durante el desarrollo de la aplicación, para poder definir con éste el ámbito del proyecto, es decir las funciones primordiales que debe realizar el software y, lo que es más importante, cuantificarlas de manera que favorezcan y contribuyan al logro de los objetivos. También aprendimos que el analista debe conocer lo mejor posible el proceso a analizar antes de entrevistarse con el usuario para recabar mayor información, pues debe saber qué preguntas hacer, qué recomendaciones dar y hacia dónde debe guiar la investigación, ya que el éxito del proyecto está en gran medida apoyado en la buena comunicación con los usuarios, pues nadie mejor que ellos conoce el ámbito donde trabajará el sistema.

Toda esta labor nos permitió dar la mejor solución al problema dadas las restricciones de fechas de entrega, los límites presupuestales, la disponibilidad del personal y las interfaces técnicas. Además, comprobamos que la calidad de un sistema de información depende de un buen análisis y de su diseño pues, si no se comprende bien lo que se debe hacer y se especifica como se logrará, jamás se podrá desarrollar un sistema eficiente y confiable, es decir, con calidad.

Sin embargo, y a pesar de todos los obstáculos a los que nos enfrentamos, se logró la terminación de un sistema que cumple con todos y cada uno de los objetivos propuestos. Dentro de estos objetivos cabe mencionar la generación e impresión de los comprobantes fiscales de la empresa, documentos con alta calidad que cumplen con las especificaciones requeridas por la Secretaría de Hacienda y Crédito Público, y con los que se logró un substancioso beneficio económico al eliminar la necesidad de mandar a hacerlos con un impresor. Otra importante ventaja que presenta SIFACOR es el crecimiento en el número de usuarios que puede atender; dada la plataforma en la que se ha montado, a medida que se expanda el número de centros de atención deberá expandirse también la red de comunicaciones para poder brindar servicios a los nuevos usuarios; sin embargo, la parte más

ardua en este proceso de crecimiento corresponde a la conexión física de las computadoras que funcionarán como terminales en la red, y crear las cuentas de las mismas, porque la inclusión de los usuarios a los servicios de SIFACOR no representa mayor trabajo que adicionar al catálogo de usuarios los nombres de los nuevos asesores y la cola de impresión a la que deben direccionarse sus trabajos.

El número de usuarios que pueden hacer uso de los servicios de la aplicación se encuentra limitado por la capacidad del equipo, en este caso el DPX2, que soporta aproximadamente a 150 usuarios en condiciones óptimas, lo que significa que en un momento determinado se pueden estar generando tantos documentos simultáneamente como usuarios tenga el sistema.

El tiempo que toma capturar un documento y entregarlo al usuario también se redujo considerablemente. En la elaboración de una factura promedio (la facturación de un teléfono, uno ó dos accesorios y un servicio) se redujo de aproximadamente 20 minutos a 7 minutos, gracias a la gran facilidad que presentan las pantallas de búsqueda y despliegue de información. En el caso de los contratos sucede algo similar, pues el sistema anterior no permitía la elaboración de éstos, es decir, se elaboraban a mano, actualmente con el nuevo sistema se realizan en 2 ó 3 minutos en promedio.

Una desventaja que presenta SIFACOR es que su correcta operación depende en gran medida del buen funcionamiento de la red pues, si la comunicación entre los equipos en la red es lenta, las búsquedas que realiza el sistema toman más tiempo. Cabe mencionar que cuando la red se torna lenta, el despacho de impresiones también reduce su velocidad en forma significativa. Esto hace necesario llevar a cabo un continuo monitoreo de la red para verificar que la velocidad de transmisión sea correcta.

Otro aspecto que, a nuestro parecer, es un inconveniente en el funcionamiento de la aplicación es el hecho de que en una factura únicamente se puede facturar un aparato telefónico; pues cuando el cliente adquiere un número considerable de teléfonos se necesitan generar tantas facturas como teléfonos se venden, lo que hace que el tiempo que se emplea para realizar la venta y registrar en la base de datos toda la información de la transacción crezca desmesuradamente. Esto se puede solucionar realizando pequeñas adecuaciones a los programas de captura de información y de generación de facturas, así como a los que permiten la elaboración de notas de crédito y cargo.

Por otra parte, nosotros proponemos que esta investigación y desarrollo sirva de base para la expansión del alcance del sistema, en el sentido de que se le brinde más apoyo, a las actividades que realizan las áreas de Contabilidad y Almacén, y no sólo se quede al nivel de afectar una cuenta contable en el momento de hacer una venta o realizar algún ajuste, y permitir un control de inventario, respectivamente. Este sistema deberá abarcar otros aspectos contables, involucrando al departamento de compras con el control de inventarios, es decir que el sistema contribuya a que este departamento realice mejor su labor de adquisición de productos y, además, ofrezca un control en el adeudo y pago a los proveedores de productos y accesorios, incorporando a sus procesos automatizados las actividades que realiza el departamento de cuentas por pagar.

APÉNDICES

APÉNDICE 1

LA NECESIDAD DE FACTURAR

Las facturas deben su origen a la necesidad de establecer contratos de adquisición de bienes y servicios para protección del cliente y del prestador del servicio. Para este último, son también un medio de protección fiscal.

DEFINICIÓN DE FACTURA

La factura puede ser considerada como un documento para el comercio de mercancías que extiende el vendedor hacia el comprador, en el cual figuran las características de los géneros vendidos tales como: cantidad en peso y medida, calidad, precio unitario, precio total, descuentos, forma de transporte, etc.

Se pueden considerar también documentos para el comercio de mercancías la Carta de Oferta, la Carta de Pedido, Carta de Envío, Acuse de Recibo y Conformidad, Vale de Pedido, etc.

Dentro del comercio tradicional se han definido diversas clases de facturas:

- **Factura de Plaza:** Cuando el cliente reside en la misma localidad del vendedor, no se producen gastos de envío.
- **Factura de Expedición:** Si el comprador y el vendedor no residen en la misma plaza, entonces se han de tener en cuenta los gastos que origine el envío de mercancías, incorporándose a la factura, según lo convenido.
- **Factura de Consignación:** Es la que remite un comerciante a su comisionista, detallando los géneros vendidos por cuentas de éste.
- **Factura de Venta en Comisión:** Es la que extiende el comisionista para dar cuenta a su comitente de los géneros vendidos por cuenta de éste.

ORÍGENES DE LA FACTURA

Para definir la factura que conocemos en la actualidad como comprobante de venta, protección del cliente y documento fiscal, es necesario hacer un breve recorrido por la historia del comercio.

En las etapas primitivas del comercio, encontramos en primer plano el trueque, que consistía en el intercambio de una mercancía por otra, en éstas nunca fue necesario un comprobante, pues el trato era directo y el compromiso mutuo. Esta actividad comercial existió fundamentalmente en las culturas china, japonesa, india, persa, asiria y mesopotámica.

No fue sino hasta la aparición de la moneda metálica con los fenicios, cuando el comercio alcanzó mayor celebridad, fundamentalmente en la cuenca del Mediterráneo. En Egipto y Grecia se empezaron a utilizar las primeras leyes para regular el comercio; inicialmente se tenían como base para el intercambio diversos medios tales como granos, conchas, sal, tablas de algodón, ganado, etc. pero, a pesar de ello, el intercambio era hecho en términos de buena fe o juramento y el comprobante estrictamente aún no existía, hecho que se prolongó durante el feudalismo hasta la época moderna con la aparición de la contabilidad.

Durante la etapa prehispánica de México, los aztecas efectuaron el comercio por medio del trueque ó, empleando en calidad de monedas, granos de cacao, telas, conchas y oro en polvo; existían además en los mercados juntas de jueces y supervisores que verificaban lo vendido así como los pesos, medidas y calidad de los productos.

El intercambio de mercancías bajo juramento cobró su mayor importancia desde la aparición de los primeros gremios europeos, fundamentalmente en Alemania, al establecerse las artes y oficios. Se dio de manera más formal y estricta durante el siglo de los navegantes españoles y portugueses, con el establecimiento de los "jueces del comercio" o "consulados", en donde se establecía un castigo a quien se descubría violando la venta y entrega de mercancías.

En la Nueva España la Legislación Comercial estaba a cargo de las Leyes de Indias, en donde los mercaderes quedaban obligados a dar "relaciones juradas" de las mercancías que cargaban, bajo pena de encarcelamiento. Posteriormente, el primer contador y el primer tesorero fueron nombrados en la Nueva España por Carlos V: Rodrigo de Albornoz y Alonso de Estrada respectivamente, en Valladolid el 15 de octubre de 1522.

La Real Hacienda de la Nueva España, que reguló el comercio de oro, plata y azogue de fabricantes y mercaderes, registraba en libros las cantidades de las piezas en transacción, las marcaba, y luego las devolvía con testimonio y certificación de asiento y registro.

Posteriormente, con el establecimiento de las sociedades mercantiles, fundamentalmente en Inglaterra, Holanda y E.U.A. , se empezó a utilizar el "papel" o "documento" para el pago de impuestos, los primeros bancos aparecieron, y se dio la necesidad de registrar compras y ventas a crédito y otras transacciones comerciales en que el precio de la mercancía no era pagado en el momento de recibirla o entregarla, así como operaciones de préstamo ó depósito de mercancía o monedas.

Pero no fue sino hasta 1881 en EUA con estudios mercantiles definidos y establecidos formalmente, cuando el documento propiamente conocido como "nota de venta" o factura adquirió las características que actualmente conocemos.

IMPORTANCIA DE LA FACTURA

La factura es el único documento que relaciona a la empresa con su cliente y que avala la legitimidad de un bien o servicio.

Derivado de la definición de factura, podemos decir que su importancia radica en sus alcances. La factura es:

- * Un comprobante de venta, es decir, sirve para definir de manera llana o detallada, la transacción de un bien o servicio (cantidades, pesos, etc.),
- * Protección del cliente, o visto de otra forma, aunque no representa necesariamente la garantía por futuros daños, depreciaciones o desperfectos, sí tiene alcances legales, y
- * Documento fiscal, es decir, ésta le sirve a la empresa para sus registros contables y compromisos hacendarios .

APÉNDICE 2

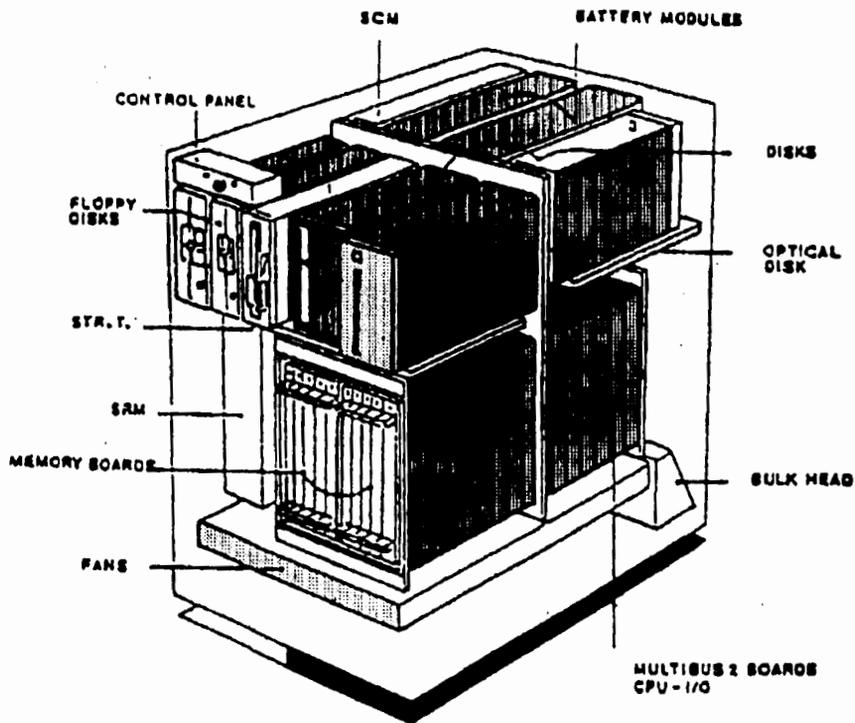
REQUISITOS FISCALES PARA LA AUTOIMPRESIÓN DE COMPROBANTES FISCALES

Dentro de los requisitos fiscales para la autoimpresión que marca el artículo 29-A del Código Fiscal de la Federación y la Resolución Miscelánea publicada el 31 de marzo de 1994, destacan los siguientes:

1. Los comprobantes deberán contener impreso el Nombre, Denominación Social, Domicilio Fiscal, y Clave del Registro de Contribuyentes de quien los expida. Cuando se tenga más de un local o establecimiento, se deberá señalar en el comprobante, el Domicilio del Local en el que se expide al mismo.
2. Deberá ser impreso el Número de Folio, el lugar y fecha de expedición.
3. Contendrá la Clave del Registro Federal de Contribuyentes de la persona a favor de quien se expide y la cantidad y clase de mercancía o bien, descripción del servicio.
4. Incluirá el valor unitario en número, e importe total en número y letra, así como el importe de los impuestos correspondientes.
5. Deberán ser impresos: Las Facturas, Notas de Crédito y Notas de Cargo, los recibos de honorarios, y de arrendamiento, así como cualquier documento que permita la deducción o acreditamiento del mismo.
6. Los comprobantes deberán traer impresa, la Cédula del Registro de Contribuyentes; sobre la cédula, no podrá efectuarse ninguna anotación que impida su lectura.
7. Deberán contener la leyenda "La Reproducción No Autorizada de este Comprobante Constituye un Delito en los Términos de las Disposiciones Fiscales".
8. Cuando las necesidades de la empresa exijan el uso simultáneo de varias series de comprobantes, se podrá utilizar las series que fueren necesarias.
9. La impresión de comprobantes deberá iniciar a partir del Folio núm. 1.

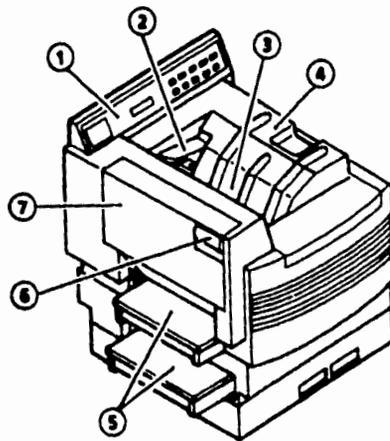
APÉNDICE 3 FICHAS TÉCNICAS

ESQUEMA DE CONFIGURACION DEL MINI BULL DPX2



El esquema muestra la distribución y el arreglo modular de los componentes principales de la minicomputadora BULL modelo DPX2.

IMPRESORA XEROX 4213

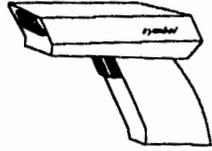


- ❶ **Panel de control:** Consiste en un esquema luminoso de la impresora, cuatro led's indicadores, seis botones y pantalla de mensajes.
- ❷ **Asa-seguro:** para abrir la cubierta superior.
- ❸ **Cubierta superior y charola:** Recibe la hojas impresas apilándolas en el orden correcto.
- ❹ **Cubierta superior:** se levanta para reemplazar los consumibles: tóner, lámpara, etc.
- ❺ **Charolas porta-papel:** Dos charolas que soportan 250 hojas c/u.
- ❻ **Asa -seguro de la puerta izquierda:** Permite abrir la puerta izquierda para servicio.
- ❼ **Puerta izquierda:** Permite remover atascos de papel en el área de fijado de impresión, además de facilitar el servicio de la impresora.

FICHA TÉCNICA DE LA LECTORA DE CÓDIGOS DE BARRAS LS2000II

PERFORMANCE CHARACTERISTICS

Type: Bi-directional scanning, retro-reflective
Scan Element: 17mm max, wedge mirror
Light Source: 675 nm, laser grade
Scan Rate: 46 scans/sec
Field Width: 30 in (762 mm) max
Working Distance: 35 in (891 mm) max
Show Angle: 20° min (front normal)
Pick: 155° min (L/R of normal)
Print Contrast Min: 2% specular reflectance as measured at 675 nm (60% for LS 2000IID)
Decode Capability: LS 2000II - Determined by interface controller; LS 2010II, LS 2020II, LS 2030II, LS 2040II, LS 2090II, LS 2051II, LS 2502II, LPTAN, LPT EAN with Supplementals, Grade 99, Grade 99 Full ASCII, Codabar, Grade 93, Grade 130 Interleaved 2 of 5, Dotsize 2 of 5, MSI Plessey (full symbology), 2D barcode supported - symbology support varies by model and host configuration
Interface Capability: Wand Emulation (Digital Wand Replacement, Symbol, Telson, and Norand terminals), RS-232C, RS-232C-Wedge, IBM 483, 4, PC 9905 9907, 9918 9920 (GCA and RS-232C), Nudorf 8812-301 POS 2000, Decole 0032A and RS-232C, Fujitsu 9030 (RS-232C), NCR 253, 1254, 2122, 2125, 2126-1120, 2126-1120, 2127, 2152, 2154, 2155, 2157, 2257, 2960, 7030, 7051, 7052, 7053 (all OCA formats), Keyboard Wedge (IBM PC, XT, AT, PS 2, IBM 3151, 3160, 3170, 3180, 3190, 317X, DEC 200, 300, 30X, NCR 7052, 3 as a keyboard wedge that support varies by model)



PHYSICAL CHARACTERISTICS

Dimensions: 5.5 in (14.1 cm) H x 4.9 in (12.4 cm) L x 2.6 in (6.5 cm) W
Weight (without cable): LS 2000II - 6.5 oz. (184 gm), LS 2010II, LS 2020II, LS 2030II, LS 2040II, LS 2050II, LS 2090II, LS 2501II, LS 2502II - 7.5 oz. (213 gm)
Electrical: Complies with UL, CSA, and VDE
Laser Class: CDRII Class II, BSI 4013 Class I, Klasse Exo
Voltage Requirements: LS 2000II - 4.5 to 14 VDC; LS 2010II, LS 2020II - 4.75 (min) to 9 (max) VDC; LS 2030II - 5 VDC (typ); LS 2040II - 8 VDC (operating); LS 2090II, LS 2090II - 4.75 (min) to 8 (max) VDC; LS 2501II, LS 2502II - 4.8 (min) to 13.5 (max) VDC
Current Requirements: LS 2000II - 125 mA at 5 VDC; LS 2010II, LS 2020II - 175 mA at 5 VDC (typical); LS 2030II - 300 mA (typ); LS 2040II, LS 2501II - 250 mA (typical); LS 2090II - 300 mA (typical)
Power Supply: LS 2000II - Power is supplied by interface controller; LS 2010II - Host power or 9-volt battery; Optional power supply is available; LS 2020II - Host power or 110-117, 240 VAC conversion to 8 VDC will mount or alkaline battery; Optional power supply is available; LS 2030II - Power supplied by IBM 483; LS 2040II - 110-117, 240 VAC conversion to 8 VDC will mount AC conversion; LS 2050II - Host power or 110-117, 240 VAC conversion to 8 VDC will mount AC conversion; LS 2090II - Host power or 110-117, 240 VAC conversion to 8 VDC will mount conversion; LS 2501II, LS 2502II - Host power, 9-volt alkaline battery, or 110-117, 240 VAC conversion to 8 VDC will mount conversion
Cable: 1 ft. (30 cm) straight section at scanner; 2 ft. (61 cm) straight section at connector plus 5 ft. (1.5 m) of coil cable; LS 2030II, LS 2501II, LS 2502II cable - 9 ft. (2.7 m)

USER ENVIRONMENT

Ambient Light: 1000 ft. candles (720 lux) in direct sunlight
Operating Temperature: 32° to 104° F (0° to 40° C)
Non-operating Temperature: -4° to 140° F (-4° to 60° C)
Humidity: 5% to 95% (non-condensing)
Shock: Unit functions normally after a 3 ft. drop (91 cm) onto concrete surface
EMI Generation: Conforms to FCC Class A limits
Electrostatic Discharge: Conforms to 15 kV rating as per IEC 801-2 (1985)

* For direct host power operation, please check that the host terminal can supply sufficient power for the LS 2000II.



SYMBOL TECHNOLOGIES, INC.
 110 White Plains Road, New York, NY 10714
 TEL: 914/941-1100 FAX: 914/941-2011
 TEL: 1-800-541-2000 (TOLL FREE)
 SYMBOL TECHNOLOGIES CANADA, INC.
 1201 Sheppard Avenue East
 Scarborough, Ontario Canada M1W 4Z2
 FAX: 1-416-291-7676 TEL: 1-416-291-7225
 For sample and details, call 1-800-541-2000
 SYMBOL TECHNOLOGIES INTERNATIONAL
 13 Falkland Park, Edinburgh Road
 Wokingham, Berkshire, England RG40 3TE
 FAX: 01344-770000 TEL: 01344-771000
 For free
 SYMBOL SYSTEMS, INC.
 10000 Holliston, MA 01920
 22 Nashua Street, Exchange
 Nashua, NH
 TEL: 603/881-2200
 FAX: 603/881-2210 TEL: 603/881-2211

For Latin America, call The Caribbean
 FAX: 1-818-960-9122 TEL: 1-818-960-9122
 SYMBOL TECHNOLOGIES INTERNATIONAL
 SINGAPORE
 Singapore, Singapore
 FAX: 65-434-9225 TEL: 65-434-9226
 Sydney, Australia
 FAX: 61-2-22-90-3030 TEL: 61-2-22-90-3031
 Taipei, Taiwan
 FAX: 886-2-27-40-1114 TEL: 886-2-27-40-1115
 Germany (Düsseldorf)
 FAX: 49-212-472740 TEL: 49-212-472741
 (Frankfurt)
 FAX: 49-212-524670 TEL: 49-212-524671
 Italy (Milan)
 FAX: 39-2-4641-805 TEL: 39-2-4641-806
 Spain (Madrid)
 FAX: 34-91-527-1122 TEL: 34-91-527-9977
 USA - New York (New York)
 FAX: 1-212-311-7122 TEL: 1-212-311-7122

WARRANTY AND SERVICE
 PART NO. 1010-PRINTED IN USA
 © 1995 SYMBOL TECHNOLOGIES, INC.

BIBLIOGRAFIA

BIBLIOGRAFIA

Wenceslao Millán Fernández
Enciclopedia de Iniciación Comercial
Casa Editorial Burgos, 1970
Madrid, España

Alberto María Carreño
Breve Historia del Comercio
Editorial Banca y Comercio
México, D.F.

Roger S. Pressman
Ingeniería de Software, un enfoque práctico
Editorial McGraw-Hill, 1973
México

Lavette Teague, Jr. & Christopher W. Pidgeon
Structured Analysis Methods for Computer Information Systems
McMillan Publishing Company, 1985
EUA

Software Engineering
The Production of Quality Software
1991
EUA

Ian Sommerville
Ingeniería de Software
1988
México

Andrew P. Sage & James D. Palmer
Software System Engineering
1989
EUA

Barbee, Te Asley & Mynatt
Software Engineering with Student Project Guidance
EUA

David A. Taylor, Ph. D.
Object-Oriented Technology
A Manager's Guide, 1994
EUA

Dawn Maneval, Kayoe Bonney, Patrick Brown, David Cortesi,
Ron Hendricks, ErikHennum & Elizabet Knittel

Informix 4GL by Example
Informix Press, 1995
USA

Burch-Grudnitski
Diseño de Sistemas de Información
Mega byte

David Budgen
Software Design
Addison-Wesley Publishing Company, 1994
EUA

Henry, C. Lucas, Jr.
The Analysis, Design and Implementation of Information Systems
McGraw-Hill, 1985
EUA

Edward Yourdon
Modern Structured Analysis
Yourdon Press Prentice Hall, 1989
EUA

Henry C. Lucas, Jr.
Conceptos de sistemas de Información para la Administración
McGraw-Hill, 1983
México

George M. Scott
Principios de Sistemas de Información
McGraw-Hill
México

Harold Koontz
Administración
McGraw-Hill
México

Russell L. Ackoff
Planificación de la Empresa del Futuro
Editorial Limusa
México

Henry F. Korth & Abraham Silberchatz
Fundamentos de Bases de Datos
McGraw-Hill, 1993
México

Miguel A. Rodríguez Almeida
Bases de Datos
McGraw-Hill, 1992
México

James A. Senn
Análisis y Diseño de Sistemas de Información
McGraw-Hill, 1992
México

Kendall & Kendall
Análisis y Diseño de Sistemas
Prentice Hall
México

MANUALES

Administración de Informix On Line
Integración de Sistemas Abiertos S.A. de C.V.

Informix SQL
Integración de Sistemas Abiertos S.A. de C.V.

Informix 4GL



Tel. 658 - 73 - 44