



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

RE - INGENIERÍA DE SOFTWARE A  
UNA APLICACIÓN BANCARIA

Seminario de Investigación Informática  
que para obtener el Título de :

LICENCIADO EN INFORMÁTICA

Presentan :

MARÍA BÁRBARA AGUILAR MALDONADO,  
JOSEFINA CASTRO LOZANO y  
SOFÍA MENDOZA FUENTES.

Profesor del Seminario :

C.P. y M.C. MARINA TORÍZ GARCÍA.



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.

**Agradecemos :**

**A nuestra asesora de tesis L.C y M.en C.C Marina Toriz Garcia  
por compartir con nosotras sus conocimientos, experiencia y su  
inapreciable amistad hasta la culminación de éste trabajo.**

**y**

**A la Universidad Nacional Autónoma de México  
por la oportunidad que nos dió de obtener  
conocimientos, ética y calidad profesional,  
que nos hacen ser buenos profesionistas  
y mejores seres humanos.**

***Barbara, Josefina y Sofia.***

**A mis padres, hermanos y amigos  
por todo su apoyo  
y muy en especial  
a Ana Vianey**

*Josefina.*

**A mis padres y hermanos  
por todo su apoyo y comprensión  
en los momentos más difíciles.**

**A todos mis amigos  
que han estado cerca de mí para  
darme su amistad y comprensión.**  
*Bárbara.*

**A mis padres Gil y Virgen  
porque sin su apoyo y comprensión  
no lo hubiese logrado.  
Mami donde quiera que estes  
nuestro sueño se logro.**

**A mis hermanos Mony y Sam  
por estar siempre a mi lado.**

**A mi esposo César  
por el gran amor y apoyo  
que me impulsan a seguir  
siempre adelante.**

**A toda toda mi familia  
por impulsarme y alentarme  
en los momentos más difíciles.**

**A Bárbara y Josefina  
porque sin su apoyo, alcanzar  
esta meta no hubiese sido  
posible.**

**Y a Dios por ponerlos a todos en mi camino.**  
*Sofia.*

## INDICE

Página .

### INTRODUCCIÓN

### CAPITULO I

#### **LA INGENIERIA DE SOFTWARE COMO BASE PARA LA RE-INGENIERIA DE SOFTWARE**

Antecedentes	1
Definiciones de la Ingeniería de Software	1
Objetivos de la Ingeniería de Software	2
Enfoque de la Ingeniería de Software	5
Lenguajes de programación	6
Técnicas de 3a. generación	10
Técnicas de 4a. generación	12
Sistemas de 4a. generación	16

### CAPITULO II

#### **LA RE-INGENIERIA DE PROCESOS DE NEGOCIOS Y LA RE-INGENIERIA DE SOFTWARE**

<b>Re-ingeniería de Procesos de Negocios</b>	
Definición formal de Re-ingeniería de Procesos	25
Reconstrucción de los Procesos	26
Método para realizar la Re-ingeniería de procesos	27
<b>Re-ingeniería de Software</b>	
Antecedentes	31
Definiciones de la Re-ingeniería de Software	32
Contexto de la Re-ingeniería de Software	34
Ciclo de la Re-ingeniería de Software	36
Captura	36
Ingeniería Reversiva	36
Expansión	40
Ingeniería hacia adelante	40
Optimización	40
Generación	40
Iniciando una Re-ingeniería de Software	40
Análisis del Sistema	42
Determinación de funciones en común	42
Determinación de datos comunes	43
Areas de Aplicación	44
Migración de datos	44
Problemas de datos	45

Re-ingeniería de datos vs Ingeniería de datos	47
Re-ingeniería de datos: Pasos y Herramientas	47
Análisis del Código	48
Análisis de datos	56
Síntesis de los metadatos	56
Rediseño	56
Revisión	57
Definición de la Exportación	58
Migración a bases datos relacionados	58
Seguimiento para obtener la documentación del sistema actual	59

### **CAPITULO III**

#### **TECNOLOGIA Y EVALUACION DE LA RE-INGENIERIA DE SOFTWARE**

Tecnología para la Re-ingeniería de Software	60
Herramientas de Re-ingeniería de Software	61
LDA	65
LINC	65
Estrategia y Economía de la Re-ingeniería de Software	70
Avalúo de los beneficios de la Re-ingeniería	72
Costo-beneficio de la Re-ingeniería de Software	73
Experiencia y Evaluación de la Re-ingeniería de Software	74
Historia del Sistema	75
Historia del Mantenimiento de los datos	75
Análisis de puntos por función	76
Métricas	77

### **CAPITULO IV**

#### **APLICACION DE LA RE-INGENIERIA DE SOFTWARE 78**

#### **CONCLUSIONES**

#### **GLOSARIO**

#### **BIBLIOGRAFIA**

#### **APENDICE 1 WFL**

#### **APENDICE 2. Reportes del sistema actual**

#### **APENDICE 3 Reportes en LDA**

## INDICE DE TABLAS Y FIGURAS

<b>FIGURA</b>		<b>Página</b>
No. 1	Gráfica de la evolución de los lenguajes	8
No. 2	Costos de software en proporción a costos de hardware	9
No. 3	Costos de desarrollo de software	17
No. 4	Etapas de Re-ingeniería de Procesos	30
No. 5	Re-ingeniería y términos relacionados	34
No. 6	Proceso de la ingeniería reversiva	39
No. 7	Evaluación de un programa	41
No. 8	Modelo del proceso usua	51
No. 9	Modelo del proceso nuevo	52
No. 10	LDA como una herramienta CASE	66
No. 11	Desarrollo de una aplicación	67
No. 12	Proceso de LDA	68
No. 13	Ambiente CASE de LINC	69
No. 14	Diagrama general del flujo de programas	82
No. 15	Proceso general del nuevo sistema	95

**TABLAS****Página**

Tabla No. 1 Comparativo de lenguajes de 3a. y 4a. generación	9
Tabla No. 2 Re-ingeniería de datos vs. Ingeniería de datos	47
Tabla No. 3 Comparativo entre pasos y herramientas	48
Tabla No. 4 Herramientas para el análisis del código fuente	55
Tabla No. 5 Herramientas representativas de la Re-ingeniería de software	63-64
Tabla No. 6 Consumo de recursos del S136 actual	91
Tabla No. 7 Volumen de operaciones del S136 actual	92
Tabla No. 8 Comparativo Sistema actual vs. sistema nuevo con Re-ingeniería.	157



**INTRODUCCION**

Por diversas circunstancias el aumento de la **COMPETENCIA** es evidente, se ha convertido en el factor comercial más importante; para ser **COMPETITIVO** se necesita mejorar, y toda mejora implica un **CAMBIO**, siendo necesario que las empresas **CAMBIEN** su forma de trabajo para triunfar o al menos garantizar su supervivencia. Para hacer frente a esto se tiene la **RE-INGENIERIA**, cuya meta principal es:

**AUMENTAR LA CAPACIDAD PARA COMPETIR EN EL MERCADO MEDIANTE LA REDUCCION DE COSTOS, O SEA LA VENTAJA COMPETITIVA.**

Antes de iniciar el proceso de **RE-INGENIERIA**, es necesario **POSICIONAR** a la empresa, aquí se determina el área donde se debe aplicar la **RE-INGENIERIA** e inicia las actividades necesarias para llevarla a cabo, implementando métodos para apoyar un **CAMBIO RAPIDO Y EFICAZ**.

La **RE-INGENIERIA** es un enfoque para planear y controlar el **CAMBIO**. Significa rediseñar los procesos de los negocios y luego implementarlos.

La **RE-INGENIERIA** no implica sólo los cambios procedimentales en los procesos de la empresa, sino además otros **CAMBIOS** asociados de igual importancia, esto es ayudar a las empresa a integrar sus actividades de negocios y el apoyo de computación dentro de un todo coherente, pero también se puede aplicar en toda la empresa o solamente en una parte de ésta.

La contribución más importante de la **RE-INGENIERIA** es la obtención de un **NUEVO AMBIENTE EMPRESARIAL, EL CAMBIO A UN NUEVO PARADIGMA**, que una vez alcanzado **CAMBIAR** ya no será difícil, sino que se convertirá en el camino del **EXITO**.

La **RE-INGENIERIA DE PROCESOS DE NEGOCIOS** considera tres componentes significativos:

- el Personal
- LA **TECNOLOGIA**
- el proceso mismo.

Que están interrelacionados y por lo tanto trabajan en forma paralela.

Con respecto a la **TECNOLOGIA** se refiere a la **TECNOLOGIA DE LA INFORMACION**, ya que cualquier empresa depende en gran medida de las computadoras, determinando que ésta es un factor importante para alcanzar ese **CAMBIO**.

## INTRODUCCION

La RE-INGENIERIA debe aplicarse cuando se presente cualquier oportunidad significativa, cuando se van a producir nuevos productos, cuando se pueden abrir nuevos mercados, cuando hay NUEVAS TECNOLOGIAS que pueden reducir los REDUCIR LOS COSTOS siendo este último el que justifica nuestro SEMINARIO DE INVESTIGACION INFORMATICA, en donde se expone y aplica la RE-INGENIERIA DE SOFTWARE para dar solución a la problemática que enfrentan las empresas debida a tanto cambio tecnológico. Tal solución se puede alcanzar solamente con una metodología, con un enfoque original y dinámico que permita controlar el cambio, mejorar la respuesta operacional y ante todo garantizar la competitividad de la empresa en esta nueva era; ¡ESTO ES LA RE-INGENIERIA DE SOFTWARE!

Si, la situación que viven diversas empresas es que muchas de sus aplicaciones críticas(aquellas cuyo procesamiento y análisis es base para la TOMA DE DECISIONES) están centralizadas en MAINFRAMES o minicomputadoras, sin embargo, no se puede estar ajeno de los recursos tecnológicos existentes en el mercado actual, tales como: microcomputadoras, redes de área local, estaciones de trabajo, etc., etc., también conocidos como SISTEMAS DE ESCRITORIO, y que a menudo nos los ofrecen como más costeables y que además incrementan la eficiencia de dichas aplicaciones.

Como ya se sabe tanto los MAINFRAMES como los SISTEMAS DE ESCRITORIO cuentan con ventajas y desventajas por lo que cabe comentar que cada año los mainframes mejoran su relación costo/rendimiento en un 10%, mientras que los sistemas de escritorio lo hacen en un 50%, significando que por menos dinero se puede tener más capacidad de procesamiento. En cuanto al costo de la expansión en disco se tiene una ventaja de 10 a 1 sobre los discos de mainframe, además por qué pagar \$4 millones de dólares por un mainframe que corre a 30 MIPS, pudiendo obtener por \$20,000 dólares un sistema de escritorio a 100 MIPS. Otras de las características propias de los sistemas de escritorio es su gran capacidad de trabajar con las GUIs(Graphic User Interface, Interfaces de Usuario Gráficas), monitor a color de alta resolución, capacidades de multimedia.

Por otro lado es cierto que desde siempre en el MAINFRAME se han procesado aplicaciones con grandes volúmenes de transacciones, que provienen de múltiples usuarios trabajando en forma concurrente, por lo que a la fecha la empresa continúa tratando de finalizar o iniciar el desarrollo de aplicaciones que llevan varios años de retraso, cargando con la planta de personal que cada vez se incrementa considerablemente.

Todo esto ha hecho que los conceptos y tendencia del cómputo se modifiquen y ha motivado que la empresa tenga que determinar la ESTRATEGIA a seguir para optimizar sus operaciones.

Entre estas ESTRATEGIAS están: el DOWNSIZING, UPSIZING y RIGHTSIZING. El DOWNSIZING, consiste en migrar aplicaciones(completas o funciones de éstas) de MAINFRAMES centralizados a redes o sistemas centralizados menores, con el fin de acercar la aplicación al usuario final. El DOWNSIZING tiene como principio elemental el que "todo lo que no da valor agregado a la producción debe eliminarse".

## INTRODUCCION

En el UPSIZING las aplicaciones abandonan el ambiente de escritorio o de redes locales para pasar a MAINFRAMES, aunque cabe comentar que casi no ocurre este fenómeno.

El RIGHTSIZING, consiste en migrar aplicaciones (completas o funciones de éstas) a los SISTEMAS QUE RESPONDAN MEJOR A SUS NECESIDADES. Es decir, se pueden combinar las ventajas de los MAINFRAMES con las de los SISTEMAS DE ESCRITORIO, y es aquí donde el MAINFRAME puede ser utilizado como SERVIDOR DE ARCHIVOS o de BASES DE DATOS y en cuanto al sistema de escritorio aprovechar su potencialidad ya mencionada.

Por todo esto, podemos ver que hay empresas que han instalado una variedad de plataformas entrando a lo que se conoce como SISTEMAS ABIERTOS, siendo su gran reto el que éstas trabajen de manera conjunta. SISTEMAS ABIERTOS es sinónimo de escalabilidad, portabilidad, interoperatividad, basándose en estándares.

Todo esto repercute rotundamente en toda la empresa..

Obviamente cualquier ESTRATEGIA de selección de hardware que se escoja, afecta ampliamente el SOFTWARE, pues de la misma manera hay que hacer una selección cuidadosa y a fondo de éste. Pero, hay que aclarar que para Tomar decisiones al respecto, es necesario hacerlo con una metodología, con un enfoque original y dinámico, y es aquí en donde entra la RE-INGENIERIA DE SOFTWARE, que a diferencia de la RE-INGENIERIA DE PROCESOS DE NEGOCIOS, no se basa necesariamente en la implantación de un nuevo sistema de tecnología de información.

En RE-INGENIERIA DE SOFTWARE, todo lo que se sabe que funciona, se aprovecha en el Sistema nuevo, y todo lo que NO, se ignora.

Antes de iniciar el PROCESO DE RE-INGENIERIA DE SOFTWARE, se debe POSICIONAR a la empresa, esto es, describir el esfuerzo encaminado para satisfacer requerimientos, fijar metas, determinar una nueva infraestructura, reubicar a la empresa para las nuevas formas de desarrollar el trabajo, pues con el logro de esto se incrementa el poder de la RE-INGENIERIA.

El alcance de este trabajo sobre la RE-INGENIERIA DE SOFTWARE comprende una parte de investigación teórica, y otra, que es su aplicación práctica.

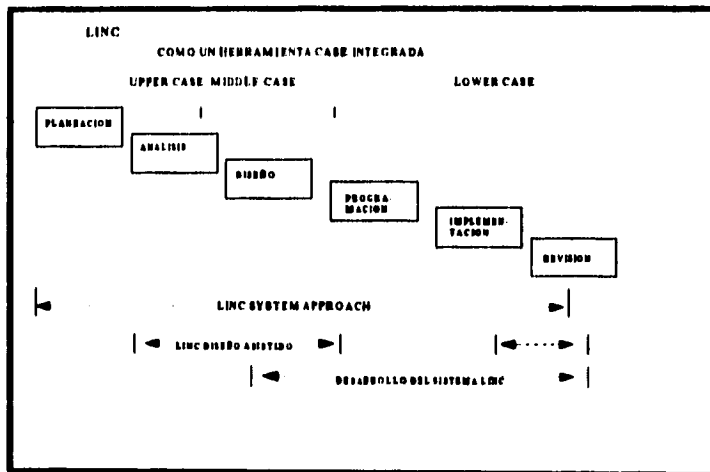
Agradecemos al Banco Nacional de México, S.A., BANAMEX por todo su apoyo para llevar a cabo la presentación de este trabajo.

La implementación de la RE-INGENIERIA DE SOFTWARE, se realizó en uno de la gran cantidad de SISTEMAS DE APLICACION con que cuenta la INSTITUCION. El SISTEMA en cuestión es el SISTEMA DE RESPONSABILIDADES DE CARTERA.

## INTRODUCCION

Se trata de un sistema de captura vía terminal, de actualización de saldos tipo batch, que da información a sucursales y departamentos de la DIRECCION, también permite registrar operaciones, mantener actualizado el archivo de datos y generar las salidas en cinta para dar cumplimiento al informe que desde 1981 exige el BANCO DE MEXICO, el cual debe de incluir una relación de los clientes que pagan puntualmente(cartera vigente) y también una de los clientes con problemas de pago y cuya deuda es mayor a los doscientos mil nuevos pesos(cartera vencida).

Para la RE-INGENIERIA de dicho SISTEMA se utilizó a LINC como una herramienta CASE INTEGRADA, en donde LDA, actúa en un nivel UPPER CASE y LINC a un nivel LOWER CASE, según se muestra en la siguiente Figura:

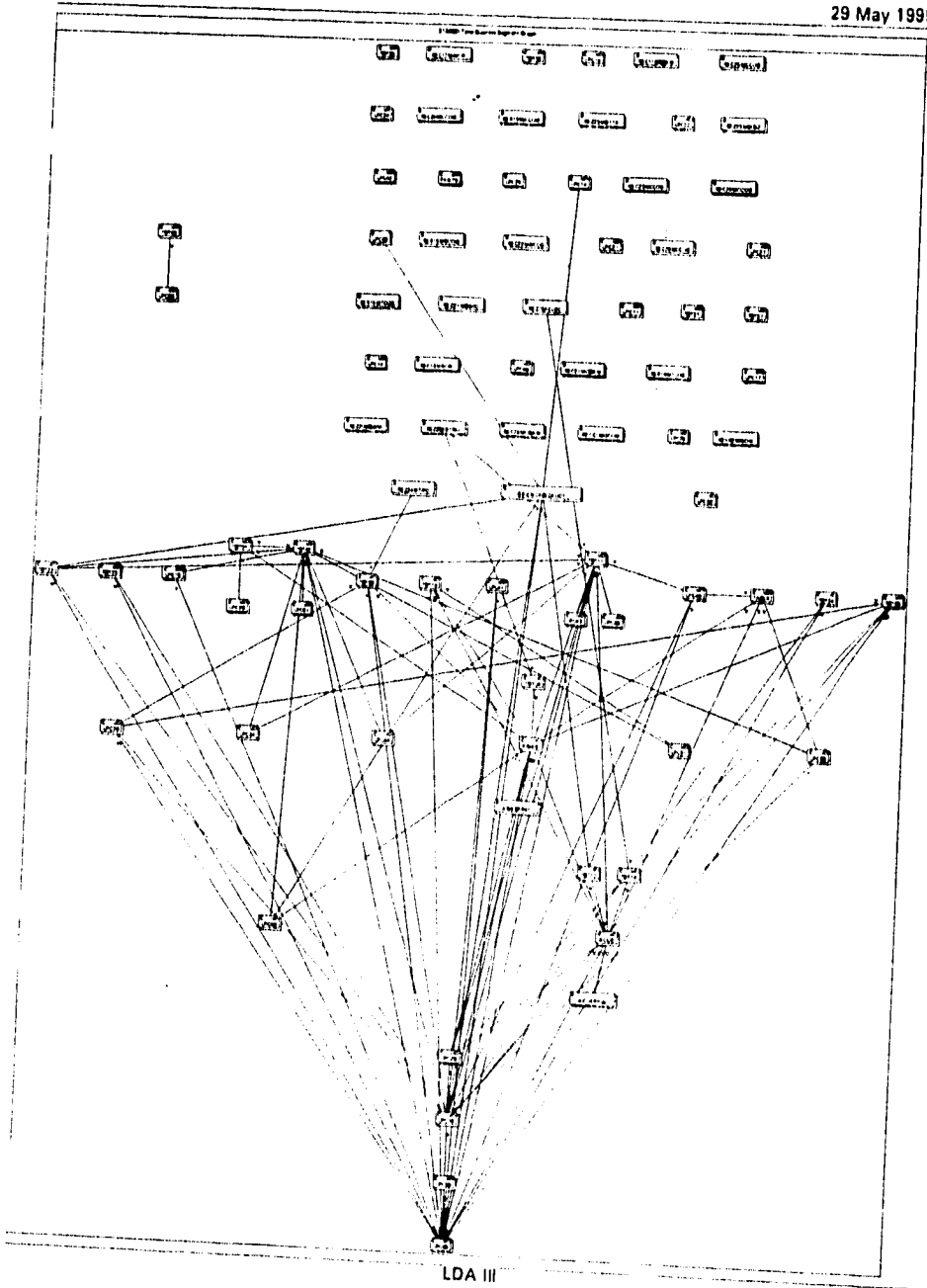


*LDA como una herramienta CASE integrada*

La siguiente gráfica que se obtuvo con LDA, muestra la importancia de tal sistema a nivel institucional.

S1368D Total Business Segment Graph

29 May 1995 Page 1



## INTRODUCCION

A continuación se comparan las características tanto del SISTEMA ACTUAL, así como las del SISTEMA NUEVO o sea con RE-INGENIERIA, en donde se pueden observar todos los beneficios de ésta.

<b>SISTEMA ACTUAL</b>	<b>SISTEMA NUEVO (Con Re-ingeniería)</b>
Utiliza archivos	Utiliza el concepto de Base de Datos
Programado en COBOL ANSI 74 (3a. generación)	Programado en LINC (4a. generación)
Hay una gran intervención del usuario (captura y corrección de errores)	Disminuye la intervención del usuario siendo todo más transparente para él.
Emite 55 reportes (13,000 páginas)	Emite sólo 16 (Reducción de un poco más del doble de páginas aproximadamente)
El usuario consulta en listados para después mediante un proceso de línea hacer algún cambio en la información.	El usuario ahora puede consultar y realizar cambios en la información mediante pantallas (23 pantallas).
El tiempo de mantenimiento es muy alto (aproximadamente 120 horas al mes)	Reducción en los reportes por falla y de requerimientos lo que permite un considerable decremento en el tiempo de mantenimiento.
Tiempo de proceso del sistema en general es de 15 a 20 días	Tiempo de proceso del sistema en general es de 3 a 5 días.
Costo N\$ 6000.00 aproximadamente	Costo de N\$ 3000.00 aproximadamente
Hay una cantidad considerable de errores en la información lo que en ocasiones se refleja en diferencias en la contabilidad.	Disminución en el número de errores por tener un mayor control de la información.
	Permite un mayor control de los registros con error para su integración al proceso.

*Comparativo Sistema Actual vs. Sistema Nuevo con Re-ingeniería*

## **INTRODUCCION**

**Para el logro de esto la organización del trabajo se encuentra de la siguiente manera:**

**CAPITULO I . La Ingeniería de Software como Base para la Re-ingeniería de Software**  
**CAPITULO II. La Re-ingeniería de Procesos de Negocios y la Re-ingeniería de Software**  
**CAPITULO III. Tecnología y Evaluación de la Re-ingeniería de Software**  
**CAPITULO IV. Aplicación de la Re-ingenier**

**CONCLUSIONES** como resultado de este estudio y su implementación.

**También se tienen los APENDICES y el GLOSARIO de términos utilizados en el desarrollo de esta investigación.**

**Cada pie de página define un concepto en el momento.**

**En cuanto a la BIBLIOGRAFIA, se encuentra dividida en libros, revistas, etc., etc., para una mejor comprensión.**

## CAPITULO I

**LA INGENIERIA DE SOFTWARE COMO BASE PARA LA  
RE-INGENIERIA DE SOFTWARE**

Para poder comprender el tema de Re-ingeniería de Software vamos a dar una introducción de lo que es la Ingeniería de Software, así como algunas definiciones.

**• Antecedentes**

El término "Ingeniería de Software", se introdujo a finales de la década de los sesentas, en una conferencia realizada para analizar la llamada crisis del software, que fue el resultado directo de la aparición de las computadoras de la tercera generación, dichas máquinas eran de una capacidad y velocidad mucho mayores a las de la segunda generación, esta potencia fue lo que hizo posible el desarrollo de aplicaciones antes irrealizables. Se desarrollaron también en esta época nuevas técnicas de programación como la multiprogramación, permitiendo entonces el establecimiento de sistemas interactivos, de multiusuario, en línea y en tiempo real.

Las metodologías utilizadas hasta ese momento para la construcción de grandes aplicaciones eran inadecuadas, ya que los proyectos siempre se retrasaban, con sobrecostos, difíciles de mantener y poco confiables, existía pues, una gran crisis en el desarrollo del software, ya que los costos de hardware disminuían mientras los del software se incrementaban considerablemente, por lo que se necesitaban urgentemente nuevas técnicas y metodologías que permitieran controlar la complejidad de los grandes sistemas de software.

La ingeniería del software, en inglés Software Engineering se utiliza como "Ingeniería de productos de programación", surge como consecuencia de los males que existen en el desarrollo de software, de la necesidad de combinar los mejores métodos, mejores herramientas para automatizar estos métodos y la calidad del software, todo esto en una disciplina, que nos ayuda a realizar mejor las actividades de coordinación, control y gestión en el desarrollo de software.

**• Definiciones de Ingeniería de software de diferentes autores:**

"Es el establecimiento y uso de principios de ingeniería, orientados a obtener software económico que sea fiable y que funcione de manera eficiente sobre máquinas reales".<sup>1</sup>

"Ingeniería del software es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido".<sup>2</sup>

<sup>1</sup> Software Engineering, de Naur. P. p. 17.

<sup>2</sup> Ingeniería de Software, de Richard Fairley, p. 2.



"La ingeniería del software surge de la ingeniería de sistemas y de hardware, abarca un conjunto de tres elementos clave: métodos, herramientas y procedimientos que facilitan al gestor, controlar el proceso del desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva".<sup>3</sup>

Para construir el software, la ingeniería del software, utiliza métodos que abarcan tareas como:

- Planificación y estimación de proyectos
- Análisis de requerimientos del sistema
- Estructura de los datos
- Arquitectura de procedimientos y programas
- Codificación, prueba y mantenimiento.

Para llevar a cabo cada uno de los métodos, se necesitan herramientas que les den soporte automático, actualmente las herramientas CASE (Ingeniería del software asistida por computadora), permiten que varias herramientas se integren de tal forma que la información creada por una, pueda ser usada por otra herramienta, las herramientas CASE combinan software, hardware y bases de datos sobre ingeniería del software.

Los métodos y las herramientas se unen mediante la utilización de procedimientos que nos proporcionen la secuencia en la que se van a aplicar los métodos, la entrega de reportes e informes y los controles que van a garantizar la calidad del software.

La ingeniería del software debe abarcar no sólo los programas de computación, sino también la documentación para instalar, usar, desarrollar y mantener estos programas y tiene como principales objetivos mejorar la calidad de los productos de software y aumentar la productividad y satisfacción profesional, de quienes desarrollan las aplicaciones, para conseguir lo anterior se requiere de la aplicación adecuada de herramientas y técnicas, así como de los principios básicos en los aspectos económicos y sociales que se toman en cuenta al desarrollar productos tecnológicos.

Por último la ingeniería del software se diferencia de la programación tradicional, en que utiliza técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro de un tiempo y presupuesto establecidos con anticipación.

#### **OBJETIVOS DE LA INGENIERIA DE SOFTWARE**

Una vez que las aplicaciones han sido seleccionadas, se pueden tomar varias decisiones.

La primera de éstas, es que después de que los problemas del negocio han sido resueltos, el ambiente puede ser seleccionado, y los administradores de sistemas desarrollarían y ejecutarían la aplicación construyendo la mejor arquitectura productiva posible. Esto puede

---

<sup>3</sup> Ingeniería de Software, de Roger S. Pressman, p. 25

ser un ambiente Host, distribuido, cliente/servidor o una combinación de desarrollos que deben ser también considerados.

Si todos los ambientes son considerados o no, es importante revisar las interfaces con los usuarios y los problemas más frecuentes de comunicaciones.

Una vez que estas decisiones han sido tomadas, el administrador de sistemas debe tomar en cuenta el futuro que la tecnología está cambiando cada vez más y más a pasos agigantados. Se debe considerar si realmente los sistemas operativos, interfaces, comunicaciones y tecnología de datos seleccionados actualmente, tienen el poder requerido en el futuro.

El software puede ser también desarrollado de forma portable y flexible en la medida como sea posible. El software es expansivo en su desarrollo y puede originar impactos críticos en la vitalidad de los negocios. Pero una aplicación puede ser eventualmente requerida para poder operar en cualquier parte, los principios de sistemas abiertos, aislamiento de código e independencia de plataforma, deben ser importantes.

En suma, se pueden hacer planes para la experimentación y mantenimiento de plataformas independientes y la potencialidad de las aplicaciones distribuidas. Un método automatizado de liberación e implementación del software puede ser también requerido. Si la aplicación es distribuida un método para obtener mejoras y arreglos debe ser considerado.

Los ambientes abiertos ofrecen una variedad de beneficios, pero se requiere de una buena planeación, una buena metodología y un grupo integrado de herramientas de diseño, desarrollo y mantenimiento.

Un pequeño grupo de código fuente que pueda trabajar en múltiples ambientes es la última respuesta a muchos de estos problemas. Esto resuelve los problemas de colocaciones múltiples de código fuente para ambientes específicos e integridad del código.

Actualmente, la mayoría de las aplicaciones con trabajos de volver a desarrollar, pueden ser implementadas utilizando COBOL. Esto es cierto, ya que en todos los ambientes de operación y todos los aspectos del código, incluyendo comunicaciones e interfaces Interface Gráfica de Usuario (GUI) (Graphic User Interface) es utilizado.

Los departamentos de sistemas de información, entienden que el código COBOL y otras inversiones son más factibles por las consideraciones estratégicas de la década actual. Esto no excluye que otras tecnologías o tecnologías mezcladas sean la solución, en realidad en ciertas instancias esto tiene sentido ya que podemos entender, que la mayoría del código fuente de aplicaciones viejas puede ser usado provechosamente a través de una gran variedad de plataformas y ambientes actuales.

• **Las causas principales que dan origen a que el software sea obsoleto son :**

- Planificación y estimación de costos imprecisos.
- La productividad no satisface la demanda de software.
- La calidad del software no es aceptable en la mayoría de los casos.

Los problemas anteriores existen debido a la propia naturaleza del software y por errores de las personas que lo desarrollan. Los responsables del desarrollo deben tener una comunicación eficiente con todos y cada uno de los componentes implicados en el desarrollo del software, para así poder comprender mejor las características especiales que éste debe tener y evitar éstos.

Para evitar los problemas mencionados, en el desarrollo de software es necesario comprender las causas de los mismos y buscar las soluciones mediante la combinación adecuada de métodos, herramientas y técnicas en cada paso del desarrollo y poder así garantizar la calidad del software y lograr una cultura informática que permita al personal encargado del desarrollo, realizar en forma eficiente la coordinación, el control y la gestión de éste.

Al conjunto de métodos, herramientas, técnicas y procedimientos que se utilizan en el desarrollo de software, se les conoce como "Paradigmas de la Ingeniería de Software" y la elección de éste se debe realizar de acuerdo con los objetivos y los controles requeridos de la aplicación a desarrollar.

## ENFOQUES DE LA INGENIERIA DE SOFTWARE

Durante la historia de la informática, se han utilizado diferentes enfoques en el desarrollo de software, estos son :

### 1) CICLO DE VIDA CLASICO DE LOS SISTEMAS

El ciclo de vida clásico, presenta grandes inconvenientes ya que difícilmente se sigue el flujo secuencial de las actividades; la dificultad que implica el definir los requisitos desde el principio del proyecto ocasiona incertidumbre por parte de los clientes, además de que las versiones operativas se tienen sólo hasta etapas finales del desarrollo, con lo cual un error arrastrado durante todo el desarrollo, resulta desastroso.

### 2) CONSTRUCCION DE PROTOTIPOS.

Cuando no se definen claramente los requisitos de entrada, proceso y salida o cuando no se está seguro de la eficiencia de los algoritmos, sistemas operativos e interacción hombre-máquina, el mejor método para desarrollar software nos lo da la creación de prototipos.

El prototipo nos sirve como mecanismo para identificar los requisitos al desarrollar el software real, pero se tienen desventajas ya que a veces es demasiado lento, demasiado grande o difícil de operar, cuando se construye el prototipo se descuidan los aspectos de calidad y mantenimiento a largo plazo, y se utilizan recursos técnicos (sistemas operativos y lenguajes de programación) inapropiados,

### 3) TECNICAS DE CUARTA GENERACION.

Las técnicas de cuarta generación proporcionan herramientas para facilitar, a quienes desarrollan software, la especificación de todas las características del software a alto nivel, generando automáticamente el código fuente. Estas técnicas se orientan hacia la posibilidad de especificar el software lo más parecido posible al lenguaje natural.

Hoy en día las técnicas de cuarta generación pueden soportar todas o algunas de las siguientes herramientas:

- Lenguajes no procedurales para consultas a bases de datos.
- Generación de informes o reportes
- Manipulación de datos
- Interacción y definición de pantallas
- Generación de código
- Facilidades de hoja de cálculo

Las técnicas de cuarta generación, aplican las siguientes actividades para el desarrollo de software:

- Recolección de requisitos
- Estrategia de diseño

- Implementación en un lenguaje de cuarta generación
- Prueba

En aplicaciones pequeñas, se puede ir desde la recolección de requisitos , hasta la implementación , pero se pueden ocasionar problemas de calidad y mala aceptación por el cliente.

Utilizando técnicas de cuarta generación se reduce el tiempo de desarrollo y se incrementa la productividad, aunque se tienen desventajas en cuanto a la eficiencia del código fuente que se genera y al mantenimiento. El uso de técnicas de cuarta generación para grandes aplicaciones implica pérdida de tiempo , ya que es necesario realizar detalladamente cada una de las actividades, perdiendo el tiempo que se ahorra al eliminar la codificación.

La aplicación de las técnicas de cuarta generación se limita sólo a sistemas de información de gestión, aunque se ha logrado abarcar los campos de ingeniería y de tiempo real utilizando herramientas CASE.

El diseño del software se traduce a un lenguaje, que nos dice la estructura de datos del software, los atributos, procedimientos y los requerimientos relacionados.

### **Lenguajes de programación**

Uno de los aspectos básicos que se debe tomar en cuenta en un sistema es el de la programación, por lo que es indispensable conocer las características de los lenguajes de programación.

Todos los lenguajes de programación son artificiales, tienen un vocabulario limitado, una gramática definida y reglas de sintáxis y semántica.

Existen dos formas fundamentales de clasificación de los lenguajes de programación, por su nivel y por sus aplicaciones.

**Por su nivel los lenguajes de programación pueden ser :**

- **Declarativos o de 4a. generación.** Son los más parecidos al inglés o al castellano en su potencia expresiva y funcionalidad y están en el nivel más alto respecto a los otros.

- **De alto nivel o de 3era. Generación.** Permiten que los algoritmos se expresen en un nivel y estilo de escritura fácilmente legible y comprensible para otros programadores. Tienen como principal ventaja la transportabilidad ya que están implementados sobre varias máquinas, de tal forma que un programa puede ser fácilmente transportado o transferido de una máquina a otra sin ser revisado sustancialmente.

- **Ensambladores y Máquina.** Son dependientes de la máquina, ya que cada tipo de máquina tiene su propio lenguaje. El ensamblador es una representación simbólica del lenguaje máquina asociado, lo que permite realizar una programación menos tediosa, pero es necesario conocer la arquitectura mecánica para realizar la programación en estos niveles de lenguajes.

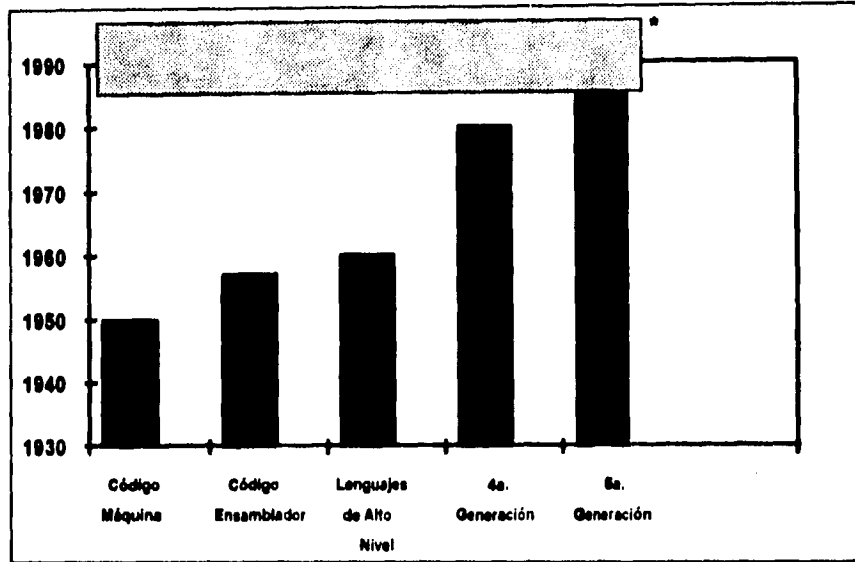
Los lenguajes máquina, ensambladores y de alto nivel, se consideran como las tres primeras generaciones, con cualquiera de estos la persona que programa debe especificar la estructura de la información, y la del control de los programas, razón por la cual a estos lenguajes se les conoce como **lenguajes procedurales**.

Los lenguajes no procedurales o de cuarta generación, no requieren por parte del que desarrolla el software de la especificación de los detalles procedurales. Por lo anterior un programa desarrollado utilizando un lenguaje de cuarta generación es la "especificación del resultado deseado en vez de la especificación de la acción requerida para conseguir el resultado"<sup>6</sup>

A continuación se presenta una gráfica de la evolución de los lenguajes de programación:

---

<sup>6</sup> In *Prise of 4GLS Datamation*, de Cobb, R.H.



*Fig. No. 1*  
Gráfica de la Evolución de los lenguajes

\* En esta etapa se desarrollan nuevas aplicaciones:

Lenguajes no procedurales El poder de las estaciones de trabajo.  
 Lenguajes específicos.  
 Desarrollo automatizado Software para DAC.  
 Probables especificaciones de técnicas de Prototipos.  
 Programación de Bases de datos Sistemas de Inteligencia Artificial.  
 Información de Ingeniería Sistemas Expertos.  
 Centros de Información Programación Orientada a Objetos.  
 Computarización de usuarios finales.

A continuación presentamos un comparación de los lenguajes de programación de 3a. y 4a. generación en la Tabla No. 1

<b>3era. Generación</b>	<b>4a. Generación</b>
Escritura legible y comprensible para otros programadores	Escritura parecida al inglés o castellano programación no estructural
Programación estructurada (Estructuras básicas secuencial, alternativa y repetitiva).	Programación no estructural
Transportabilidad a través de diferentes sistemas informáticos	Portabilidad a través de diferentes sistemas informáticos
Mantenimiento de programas deficiente y gran cantidad de código fuente	Mantenimiento eficiente y efectivo de los programas, gracias a que se genera poco código fuente
	Generación de código fuente en lenguajes de tercera generación como COBOL o C
	Disponibilidad para trabajar con micros y mainframes
	Soporte de procesamiento centralizado y descentralizado
	Permite la migración de una plataforma de hardware a otra
Acceso a archivos planos, deficiente para consultas	Acceso a archivos de bases de datos interactivos y programados
	Las operaciones o procesos en paralelo se pueden ejecutar en diferentes equipos para agilizar tiempo de proceso
Procesos en Batch o Lotes	
Consideran la integridad de los datos	Fundamento de Bases de Datos Relacional
	Accesos multiusuario a las bases de datos

*TABLA No. 1  
Comparativo de lenguajes de 3a. y 4a. generación*



**DESARROLLO DE SISTEMAS UTILIZANDO TÉCNICAS DE 3A. GENERACIÓN.**

Lenguajes como Cobol, Fortran (3a. Generación) siguen utilizándose ampliamente después de casi 30 años de su introducción en aplicaciones de procesamiento de datos, la programación en un lenguaje de 3a. generación es estructurada, con un diseño que va de lo general a lo particular (Top Down), permite utilizar estructuras que siguen utilizándose como son la secuencial, la alternativa y la repetitiva.

**Secuencial.-** Se utiliza cuando se quiere indicar que unas determinadas acciones se deben realizar en secuencia. Se escriben unas debajo de otras.

**Alternativa.-** Para indicar que según el resultado de una pregunta hay que hacer unas acciones u otras.

**Repetitiva.-** Cuando se desea que un conjunto de instrucciones realice un número finito de veces. En Cobol es un bucle o conjunto de sentencias que se repiten mientras se cumple una determinada condición.

Las aplicaciones de procesamiento de datos programados en Cobol, tienen como principal objetivo la creación, mantenimiento y compendio de datos en registros y archivos. El volumen de los datos que se encuentra en estos archivos es grande, por lo regular varios miles de registros por archivo y varios cientos de caracteres de información por registro.

Un programa típico de procesamiento de datos de 3era. generación gasta la mayor parte del tiempo de cálculo haciendo operaciones de entrada-salida.

La mayoría de las aplicaciones clásicas de procesamiento de datos se realizan por lotes (batch) en vez de interactivo, es decir pueden planificarse sobre una base regular, predecible y los requerimientos son generalmente conocidos de antemano.

La programación en un lenguaje de alto nivel o 3era. generación considera más la integridad de los datos, esto implica no solo la precisión y fiabilidad sino también la seguridad de éstos. Los programas protegen a los archivos de posibles contaminaciones con datos erróneos y asegura que datos delicados sean accedidos sólo por las personas que necesiten hacer uso de estos.

Cobol es el lenguaje más ampliamente usado para aplicaciones de procesamiento de datos. Fue diseñado y refinado como un lenguaje que hiciera que los programas y técnicas de programación fueran fácilmente compartidas y transferidas de unas máquinas a otras, contiene poderosos elementos funcionales incorporados, tales como una característica de escrituras uniforme, una función de búsqueda en tabla y una facilidad para la ordenación, todo esto se ha diseñado para ayudar al programador a organizar, accesar, actualizar, reordenar y sacar informes de sus archivos, tiene muchas facilidades para la lectura-escritura de datos durante las ejecuciones de programas. Un archivo en Cobol es una colección de registros. Un archivo puede ser accedido secuencialmente mediante un programa para

entrada o salida o entrada/salida o en forma aleatoria (es decir, los registros pueden ser leídos y escritos).

El acceso secuencial accesa a los registros del archivo en el mismo orden en el que están almacenados .

El acceso aleatorio no accesa los registros del archivo en ningún orden particular, sino según convenga al programa. Entonces cualquier registro en un archivo accedido aleatoriamente puede ser leído cero, una o más veces en una ejecución de un programa.

Cobol, se puede considerar en la actualidad como un mal lenguaje de programación, anticuado y engorroso, cuando en realidad el problema estriba en que la preparación de sus adeptos puede ser deficiente. Desde hace muchos años se ha augurado la desaparición de éste lenguaje, como lenguaje base en muchas organizaciones, pero esto no sucede sobre todo en el entorno de las grandes computadoras y grandes centros de cómputo. No obstante, existe excelente bibliografía sobre programación estructurada y diseño modular y en términos de productividad sería sumamente provechoso elevar el nivel de conceptualización computacional de un conjunto de analistas y programadores de Cobol.

Cuando se desarrolla una aplicación de procesamiento de datos utilizando un lenguaje de 3a. generación una de las principales desventajas con respecto a un lenguaje de 4a. generación como por ejemplo, LINC o SQL. es el manejo de bases de datos, ya que en el primero el acceso a los archivos a través de los programas requiere de bastante código, lo cual en determinados momentos puede hacer tedioso el trabajo del programador.

En contraste las bases de datos tradicionales proporcionan un conjunto de herramientas para acceso mediante programas y una facilidad de consulta aparte para peticiones ad hoc, sin ninguna sinergia entre los dos modos de acceso.

**DESARROLLO DE SISTEMAS UTILIZANDO TÉCNICAS DE 4A. GENERACIÓN.**

Los lenguajes de 4a. generación se han utilizado en aplicaciones de bases de datos y áreas de procesamiento de datos comerciales, por ejemplo, LINC es un ambiente total para desarrollo y operación de los sistemas informáticos que modelan sistemas de negocios, este lenguaje reduce los tiempos de generación de sistemas en línea, asiste a los diseñadores en el desarrollo de sistemas, con un editor de sintáxis en línea, pantalla para dibujar, sistema de seguridad, capacidad de configuración para red, diccionarios de datos y ayuda en línea. Es un compilador interactivo para línea y red y genera bases de datos y código de 3a. generación.

El advenimiento de los lenguajes de 4a. generación en la forma de planeación y modelación de lenguajes, consultas a bases de datos, reportes, y hojas de cálculo en software de aplicación, presenta nuevas oportunidades en el diseño y codificación de sistemas, ya que se tiene un enfoque no procedural y los problemas están orientados a lenguajes.

Los lenguajes de 4a. generación se utilizan frecuentemente en ambientes de centros de información en donde se desarrollan sistemas para dar apoyo a la toma de decisiones a nivel ejecutivo. Una aplicación desarrollada utilizando un lenguaje de 4a. generación proporciona un mantenimiento más eficiente y efectivo.

Los lenguajes de cuarta generación fueron creados en base a problemas reales, para cubrir los siguientes objetivos:

- Mejorar la velocidad de los procesos de construcción de aplicaciones.
- Crear aplicaciones sencillas para consultas y cambios, así como para reducir los costos de mantenimiento.
- Minimizar los problemas de depuración.
- Generación de código con expresiones de requerimientos de alto nivel.

En la actualidad muchos sistemas desarrollados con lenguajes de 4a. generación son frecuentemente utilizados por gerentes o ejecutivos que no tienen experiencia en el procesamiento de información, aquí las decisiones individuales se hacen utilizando simplemente la información obtenida.

Los usuarios entienden las operaciones matemáticas y la manipulación de los datos en las consultas que hacen, también pueden conocer, dónde se encuentran los datos que quieren que aparezcan en un reporte de salida, pero tienen problemas en la determinación de la secuencia propia de la lógica de los programas. Muchos estudiantes se enfrentan a problemas similares, en sus clases iniciales de programación aprenden rápidamente las sintáxis para varios tipos de instrucciones, pero tienen gran dificultad en aprender las combinaciones propias y las instrucciones necesarias para resolver problemas específicos.

Los lenguajes de 4a. generación proporcionan un futuro funcional con la generación de código fuente en un mínimo de tiempo, en lenguajes de 3a. generación como Cobol, PL/I, C, Fortran, Basic y Pascal.

Los lenguajes de consulta a bases de datos se enfocan a la extracción y reporte de datos desde la base de datos, el mantenimiento a las bases de datos es realizado por el departamento de sistemas de información. El acceso a las bases de datos requiere de mayor coordinación entre los usuarios y el departamento de sistemas de información que es el que modela las aplicaciones.

Hoy en día la tecnología en informática ofrece una arquitectura para la consistencia de los desarrollos, la integración de aplicaciones desarrolladas en diferentes ambientes de computación y el uso más efectivo de los lenguajes de 4a. generación, esto trae consigo el poder crear sistemas con las siguientes características:

- Consistencia en las tareas del desarrollo de las aplicaciones
- Facilidad de acceso y distribución de aplicaciones
- Consistencia en el uso de las aplicaciones

Los lenguajes de 4a. generación están disponibles para trabajar con microcomputadoras y mainframes o se puede emigrar de una plataforma de hardware a otra y soporta procesamiento descentralizado y centralizado. La migración de la aplicación desde una microcomputadora a un mainframe se realiza con un revisión mínima requerida incrementándose con ésta la productividad. Algunos segmentos del desarrollo de la aplicación como las operaciones en paralelo, pueden ser descentralizadas en microcomputadoras individuales para agilizar el tiempo de proceso.

En muchas organizaciones el departamento de sistemas determina cuál software de lenguajes de 4a. generación va a ser instalado en sus mainframes o microcomputadoras. En el ambiente de los mainframes la oportunidad de adquirir herramientas de lenguajes de 4a generación es remoto, mientras que la alternativa de obtener software para ambiente de microcomputadora es más real.

Para cada aplicación el departamento de sistemas, debe evaluar en base a toda una metodología y utilizando diversos criterios de selección, cuál es la herramienta más apropiada a utilizar, de entre varias disponibles.

- **Las características principales de los lenguajes de 4a. generación son:**

- **Portabilidad a través de sistemas informáticos.** Todos los productos de 4a. generación están diseñados para trabajar con computadoras personales, estaciones de trabajo y redes de área local, minicomputadoras y macrocomputadoras. Las aplicaciones monousuarios pueden ser transferidas a sistemas mayores de mini o macrocomputadoras cuando crecen. Los datos de las bases de datos corporativas pueden ser extraídos y remitidos a bases de datos departamentales o personales.

- **Tienen fundamento relacional** ya que son lenguajes para trabajar con bases de datos relacionales en su mayoría. La estructura tabular de filas y columnas de una base de datos relacional, es intuitiva para los usuarios y hace que los lenguajes se mantengan simples y fáciles de entender.

- **Tienen estructuras de alto nivel en inglés,** que son sentencias sencillas por lo que son fáciles de aprender y entender. Se describen los datos a recuperar. La mayoría de las sentencias dicen lo que significan y pueden ser leídas como frases claras y naturales.

- **Son lenguajes de consultas interactivas** que proporcionan a los usuarios acceso ad hoc a los datos almacenados, un usuario puede por lo tanto obtener respuesta a cuestiones complejas en minutos o segundos, en fuerte contraste con los días o semanas que le llevaría a un programador escribir un programa para dar un informe a la medida. Los datos son más accesibles y pueden ser utilizados para ayudar a las organizaciones a tomar decisiones adecuadas.

- **El creador de una base de datos en lenguajes de 4a. generación puede dar a los usuarios diferentes vistas de sus estructuras y contenidos,** mejorando la seguridad de los datos.

- **Las estructuras de las bases de datos pueden ser modificadas constantemente,** adaptándose a exigencias cambiantes mientras se continúa con las aplicaciones en línea.

Quando se desarrolla una aplicación de procesamiento de datos utilizando un lenguaje de 4a. generación, se definen la estructura y organización de los datos almacenados y las relaciones entre éstos, permitiendo también la recuperación, actualización y compartición de datos por parte de usuarios concurrentes. Se tienen restricciones de integridad y se limita la capacidad de los usuarios para actualizar las Bases de datos utilizadas.

- **Límites de funcionalidad.**

Los lenguajes de cuarta generación varían gradualmente en su poder y capacidades.

Algunos lenguajes son meramente de consulta; otros son generadores de reportes o generadores de gráficas; otros pueden generar aplicaciones completas; algunos son de alto nivel para lenguajes de programación, otros pueden ser utilizados por usuarios finales, o por

analistas de sistemas directamente. Algunos lenguajes son diseñados sólo para una clase específica de aplicaciones, otros son altamente restringidos en su rango; otros pueden ser utilizados en una diversidad de aplicaciones.

En la cuarta generación, mucho más que en la tercera, se puede seleccionar el lenguaje apropiado para cada aplicación.

- **Principios básicos en el diseño de 4GLs.**

Los lenguajes de cuarta generación 4GLs representan un incremento en el poder de las computadoras al trabajar más eficientemente. Una vista a los lenguajes de tercera generación nos muestra que el diseño concierne más a los costos de personal que a los costos de hardware. Estos son algunos principios importantes :

- minimizar trabajo
- minimizar tiempo
- minimizar errores
- minimizar mantenimiento
- maximizar resultados

- **Propiedades de los lenguajes de cuarta generación.**

Para que un lenguaje pueda ser llamado lenguaje de cuarta generación , es necesario que cubra las siguientes características:

- Que sea amigable con el usuario
- Que no requiera de un programador profesional para obtener resultados satisfactorios con éste.
- Que emplee directamente un administrador de base de datos.
- Que los programas para la mayoría de las aplicaciones, puedan ser creadas con un orden de magnitud inferior que con Cobol.
- Utilizar código no procedural, cuando sea posible.
- Que sea diseñado para operación en línea.
- Que proporcione la facilidad de entender y mantener el código por otra persona.
- Que esté diseñado para fácil depuración.
- Que los prototipos puedan ser creados y modificados rápidamente.
- Que los resultados puedan obtenerse en una orden de magnitud de tiempo menor que con COBOL o PL/I para la mayoría de las aplicaciones.

## SISTEMAS DE CUARTA-GENERACIÓN

- **Alcance de los Sistemas de Cuarta-Generación.  
Aplicación y Métodos de Evaluación<sup>5</sup>**

Muchas organizaciones utilizan la administración de Sistemas de Información, contruidos en lenguajes de tercera generación como COBOL y Fortran, puede ser más efectivo producirlos y mantenerlos utilizando herramientas modernas. Esas herramientas han tenido que utilizar varios nombres, incluyendo lenguajes de Cuarta-Generación (4GLs), generadores de aplicaciones o más recientemente Sistemas de Cuarta Generación(4GSs). El término cuarta generación está abierto para una gran variedad de significados.

Es verdad que 4GL implica todas esas fases de diseño y desarrollo de aplicaciones, no es una fase de código, la cual es después de todo una pequeña proporción relativa de el total de esfuerzo involucrado en el desarrollo y mantenimiento de un sistema de aplicación más grande.

Esto es, como siempre, un tema central común de todas las definiciones de un ambiente de 4G, y eso es significativo en las mejoras de la productividad convencional de métodos de aplicación de software escritos (otra vez "métodos convencionales" , está abierto a diferentes interpretaciones dependiendo del uso individual, pero generalmente es tomado como un lenguaje de alto nivel como COBOL o PL/I, junto con archivos manuales para uno u otro archivo de estructura secuencial o bases de datos jerárquicas en red).

- **Proceso de Desarrollo de Software**

En los últimos 20 años se ha visto un estable y continuo cambio relativo en el costo de software y componentes de hardware de los sistemas de cómputo.

La figura No. 2 muestra una gráfica proporcional de los cambios. Es muy difícil saber con precisión donde estamos en la curva, en particular, cuál ha sido el incremento en los costos de software.

---

<sup>5</sup> Fourth -Generation Systems, de Simon Holloway

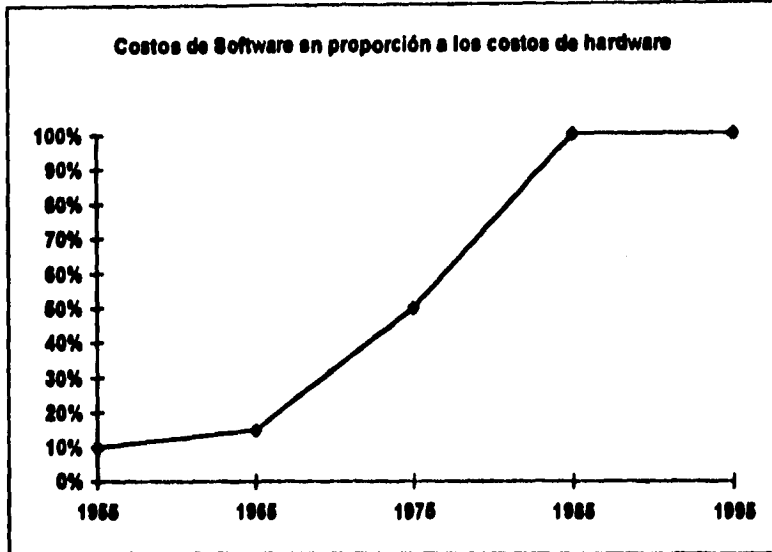


Fig. No. 2  
Costos de Software en proporción a costos de hardware

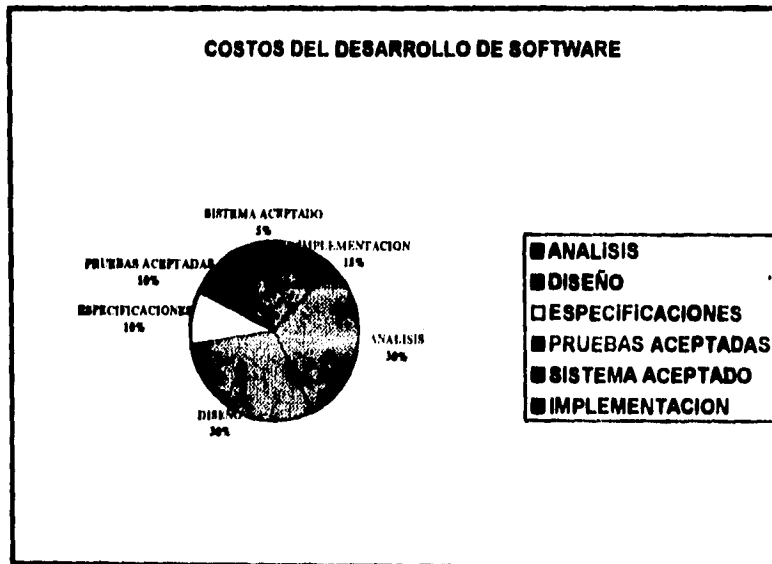


Fig. No. 3  
Costos del desarrollo de software



En la Fig. No. 3, se muestra claramente que las mayores áreas en las cuales están concentrados los requerimientos es en el análisis y diseño de sistemas. Algunas instalaciones de desarrollo de proyectos (DP) están por debajo de reducir el tiempo de escala del ciclo de desarrollo de sistemas. Consecuentemente esto es apropiado para distinguir las herramientas que reducirán el tiempo de desarrollo de una pieza de software de esos que sólo reducen un total de horas-hombre de un proyecto completo.

En suma, es necesario atacar los diferentes niveles de complejidad del total del proceso de desarrollo de software. Como siempre, las herramientas de aplicación, en algunas fases necesitarán más inversión de tiempo que otros.

- **Características de 4GLs:**

El término 4GL es usado para describir un paquete de software, el cual está dirigido a la simplificación de procesos de desarrollo de software de aplicaciones.

Los 4GLs es la mejor serie de aplicaciones de datos-intensivos, donde relativamente, simples operaciones están cargando por fuera pequeños volúmenes de datos, una larga suma de datos un poco complejos de procesar. En los 4GLs está implícita la interacción para vencer los problemas de procesamiento de archivos, relevando a los programadores de las tareas propensas a errores, controlando la secuencia de lecturas y escrituras durante la mezcla y actualización de archivos. Similarmente muchas de las decisiones acerca de la necesidad de definición de datos en un programa COBOL están igualmente hechas por el uso implícito de mecanismos de escritura.

Existen limitaciones en las interfaces de los lenguajes procedurales. Esto el programador lo puede evitar con un lenguaje convencional y con las facilidades de 4GL.

- **Componentes de 4GLs.**

En el mercado existen muchos paquetes que contienen un gran número de componentes, que proveen al usuario, de las facilidades descritas previamente.

- **Sistema de Diccionario de Datos**

El diccionario de datos es una de las partes más importantes de un sistema, es toda la información acerca de la construcción de procedimientos, más y más detalles acerca de la aplicación puede ser almacenada- En adición al 4GL, salva a los usuarios de pérdidas de tiempo y esfuerzos. Cuando el diálogo es completo, el diccionario de datos contiene una especificación completa de la aplicación. El diccionario también es la fuente de la documentación de un sistema.

**• Herramientas de desarrollo de Aplicaciones:**

La herramienta de desarrollo de Aplicaciones utiliza la información que está en el diccionario de datos, para poder correr la aplicación existen dos formas; que son las siguientes:

**1a.** Es posible usar compilador. Este puede ser obtenido de la información que existe en el diccionario de datos, generando un programa en 3GL, como COBOL, y usando un compilador convencional, o creando un código interno y ejecutarlo.

**2a.** Usar un intérprete. Este es un programa que utiliza valores en el diccionario de datos para un control, es un run-time. Para acceder el diccionario de datos el intérprete utiliza la secuencia de I/O y otras funciones para ser cargadas fuera en una secuencia correcta. La segunda alternativa tiene diferentes beneficios. El compilador puede producir eficiencia de código run-time, mientras el intérprete puede responder inmediatamente a un cambio en el diccionario de datos, sin la necesidad de recompilar.

**-Manejador de un Sistema de Base de Datos (DBMS, Data Base Management System)**

Algunos sistemas de DBMS tienen diccionario de datos, donde especifican información que protege a la Base de Datos.

**• Interfaces de comunicaciones de datos:**

Es esencial que un paquete 4GL tenga interfaces de comunicación. La interface puede ser usada para un intercambio de datos entre mainframes y estaciones de trabajo (workstations), para soportar el uso de micro desk-top para un intensivo análisis de pequeñas partes incorporadas en una Base de Datos.

**• Diseño de pantallas:**

La parte principal de un sistema es la pantalla que se le presenta al usuario de una aplicación. Muchos 4GLs tiene facilidades para que cada usuario puede diseñar formatos de pantallas directamente en la terminal o en una estación de trabajo. Los detalles de la posición y contenido de cada campo creado son registrados en el diccionario de datos.

**• Lenguajes de consulta.**

Muchas empresas necesitan consultar sus datos, consecuentemente los sistemas 4GLs tienen alguna forma de lenguajes de consulta, en este caso el estándar es SQL.

**• Gráficas de negocios:**

Con el desarrollo de más estaciones de trabajo sofisticadas y terminales se ha dado la necesidad de presentar los datos en otras formas como tablas o figuras.

- **Contribución de 4GLS para desarrollo de software.**

4GL ayuda a la disminución de errores en código, para acceder archivos y crear pantallas, ésta es una de las mayores fuentes de error. Con un 4GL la responsabilidad de codificación de un programador es mínima. 4GL provee una buena documentación en un estilo uniforme que disminuye el mantenimiento de código.

## **CONSTRUCCIÓN DE BLOQUE DE UN SISTEMA DE CUARTA GENERACIÓN**

### **Sección General de Criterios:**

Con un lenguaje de 4a. generación puede obtener resultados en un décimo de tiempo y no como COBOL. El término más frecuentemente asociado con 4GS (Sistema de 4a. generación) es "de uso fácil, "fácil de aprender" y "amigable". Porque el usuario no necesita de una sintaxis o experiencia en el uso del sistema en orden para hacerlo efectivo. En contraste los lenguaje de 3a. generación como COBOL, el cual no puede ser llamado "amigable", porque es fundamentalmente procedural y demanda de detalles de sintaxis.

### **Factores en la Selección y Evaluación de Productos de Sistemas de 4a. Generación:**

Existen muchos factores en la selección de productos 4GS para algunas aplicaciones algunas de las más importes son:

- Tipo de aplicación
- Rapidez
- Tamaño
- Costo
- Portabilidad de datos
- Portabilidad de Código
- Reputación del vendedor
- Fácil aprendizaje
- Fácil mantenimiento
- Competencia

### **Aplicaciones:**

Es importante considerar cuáles aplicaciones pueden ser beneficiadas al utilizar 4GLs.

La corriente de 4GS son herramientas las cuales mejoran la productividad en ciertas áreas de aplicación, y no son capaces de resolver fácilmente problemas de aplicaciones.

Como siempre muchas fallas de aplicaciones en categorías bien definidas, se solucionan utilizando equipo moderno y métodos de diseño .

**Ejemplos de Desarrollo de Sistemas de Aplicación:**

- Consultas de directorios
- Sistemas de Contabilidad
- Sistemas de Citas
- Sistemas Personales
- Controles de Plantas
- Paquetes de Dibujo para arquitectos
- Paquetes de Diseño de Circuitos Electrónicos
- Paquetes de Circuitos Electrónicos
- Sistemas de Comunicaciones de Oficinas
- Sistemas de Diagnósticos Médicos
- Sistemas de control de Tráfico Aéreo
- Sistemas de Publicidad
- Sistemas de Enciclopedias computarizadas
- Sistemas de misiles guías
- Sistemas de Admon. de misiles
- Sistemas de Registros criminales
- De registros fotográficos, etc.

**Propiedades de las aplicaciones:****Velocidad:**

Muchas aplicaciones de 4GS son lo suficientemente rápidas para aplicaciones comerciales.

Pero hay aplicaciones que necesitan un mayor nivel de velocidad, esto no es una desventaja usando productos 4GS. Va a ser esto de acuerdo al estudio que se realice para seleccionar la herramienta. Los productos 4GS ayudan al desarrollo y mantenimiento de sistemas implementándose para aplicaciones mucho más rápidas.

**Tamaño:**

Existe hardware que es suficiente para soportar aplicaciones sofisticadas y tienen suficiente memoria para correr sistemas. Si un sistema es muy grande hay métodos que pueden reducir el tamaño. Un método para simplificar, es utilizar un algoritmo de compresión/expansión de los datos. Otro método sería hacer un análisis detallado de la aplicación utilizando código para reducir el tamaño de los datos.

**Costo:**

El costo para el desarrollo de sistemas usando 4GS ha bajado en los últimos años

Algunos sistemas 4GS corren en hardware similar. El costo de software 4GS es relativamente bajo.

Cuando un producto 4GS ha sido seleccionado para una aplicación, si ésta requiere correr en mainframes, la evaluación basada en PC puede ser usada para ser desarrollada en un sistema el cual subsecuentemente correrá en un mainframe.

**Portabilidad:**

Es importante que una aplicación puede migrarse de una máquina a otra.

Algunas versiones de productos 4GS son hechos para un uso estándar de lenguajes de consulta, el caso de SQL o INGRES y algunos productos de código son desarrollados en lenguajes estándar como C, PL/1, FORTRAN, COBOL. El uso de lenguajes de consulta estándar, es posible transferirlo a otro sistema 4GS.

Algunos sistemas producidos en código de lenguajes estándar como C o COBOL, pueden ser procesados en un nuevo ambiente, usando 4GS para generar un nuevo sistema.

**Reputación del Vendedor:**

Hay muchas empresas grandes que venden sistemas 4GS los cuales no pueden dar soporte a sus clientes en comparación con pequeñas empresas. Para saber elegir se tiene que hacer un análisis a fondo.

**Fácil de Aprender:**

Existen dos tipos de usuarios de los productos 4GS. Algunos usuarios desarrollan aplicaciones usando sistemas 4GS, otros utilizan las aplicaciones desarrolladas con 4GS.

Existen diferencias considerables entre el experto de computadores y esos dos tipos de usuarios. Idealmente el uso de un sistema puede ser fácil, para el desarrollador de la aplicación y para el usuario final. Eso es como ser comerciante entre usuario final y desarrollador de los requerimientos.

Usuarios finales más sofisticados necesitan conocer el uso de lenguajes de consulta, edición de textos, preparación de documentación, hojas de cálculo y simples lenguajes procedurales.

**Lo que tienen que saber los desarrolladores de aplicaciones 4GS:**

- Concepto de bases de datos relacionales, semejanzas de las entidades, atributos, relacionales
- El concepto de vistas con bases de datos
- Para algunas aplicaciones, conocer bases de datos
- Sistemas de diccionario de datos
- Lenguajes procedurales como C, Pascal o COBOL
- Software de multitareas
- Hardware moderno
- Sistema administrador de Windows

**De uso fácil .**

Un usuario final necesita software de uso fácil.

Esto muestra la necesidad de seleccionar una herramienta de uso fácil para los que no son especialistas en computadoras.

**Mantenimiento Fácil:**

Los sistemas 4GS necesitan mantenimiento.

El mantenimiento es por muchas razones como el detector de errores y el que ve las necesidades de los usuarios. Un ejemplo de mantenimiento es que los usuarios requieren de adicionar campos a un registro de una base de datos, párrafos, copiar nuevas bases de datos y otras modificaciones.

**Competencia:**

Algunos sistemas de 4GS proveen un ambiente de usuario amigable en los cuales un gran número de tareas es cargado por fuera y éstas algunas veces dificultan el uso de algunas aplicaciones y otras no. En este ambiente tecnológico existe mucha competencia, por lo que el cliente va a obtener el que más beneficios le dé y menos trabajo le cueste.

## CAPITULO II

**LA RE-INGENIERIA DE PROCESOS DE NEGOCIOS  
Y LA RE-INGENIERIA DE SOFTWARE**

El presente capítulo introduce al lector a conocer las diferentes definiciones de la re-ingeniería de software y la tecnología de software existente.

En la introducción de este trabajo se muestra como la RE-INGENIERIA DE SOFTWARE, es sólo una parte de la RE-INGENIERIA DE PROCESOS DE NEGOCIOS, pero dada la importancia de las dos, a continuación se explican con más detalle cada una de éstas.

**RE-INGENIERIA DE PROCESOS DE NEGOCIOS.**

Hablar de RE-INGENIERIA DE PROCESOS DE NEGOCIOS, significa un cambio radical, esto es, empezar de cero, dejando a un lado todos los conocimientos que por más de doscientos años se han tenido acerca de la ADMINISTRACION, adoptando esta nueva corriente administrativa con el fin desde luego, de optimizar las funciones de las empresas.

La RE-INGENIERIA DE PROCESOS DE NEGOCIOS es una alternativa para eliminar todos aquellos métodos y principios organizacionales obsoletos y/o ineficaces con los que se manejan las empresas y que son los que a veces las llevan a su desaparición.

El avance tecnológico, el libre comercio y nuevas expectativas de los clientes al contar con mayor oferta de productos han obligado a las empresas a buscar un cambio; un cambio en su forma de trabajo para mejorar su rendimiento, basándose en qué es lo que se tiene que hacer, y cómo hacerlo, pero sin tomar en cuenta cómo se están haciendo las cosas en este momento. Pero no, no se trata de utilizar tecnología u otras herramientas para determinar un nuevo modelo de desarrollo de la empresa, sino que su importancia radica en que ayuda a las empresas a organizar su forma de trabajo de acuerdo a las exigencias del ya tan cercano siglo XXI.

Los problemas a que se enfrentan las grandes organizaciones se deben a la competitividad global en los mercados, además de que ya nada es constante ni previsible. Lo que está impulsando a las compañías a penetrar cada vez más en los territorios que antes no ocupaban son los clientes, la competencia y el cambio.

Los clientes consumidores y corporaciones, exigen productos y servicios que satisfagan sus necesidades particulares y específicas.

La competencia se intensifica, ya que al venirse abajo las barreras comerciales, ninguna organización está protegida sobre la competencia extranjera.

Con lo anterior se ha formado un mundo nuevo para los negocios y cada día se hace más notorio que organizaciones diseñadas para que funcionen en un ambiente no se pueden adaptar para que funcionen en otro.

Algunas personas atribuyen los problemas de las organizaciones a las deficiencias en la administración de éstas, pero ningún estilo de administración : a lo largo de los últimos veinte años ha logrado detener el deterioro en el desempeño competitivo de las organizaciones, algunas otras personas piensan que la automatización es la solución a todos los problemas de los negocios, no cabe duda que la automatización permite realizar algunas tareas más rápidamente, pero se están haciendo los mismos trabajos y entonces realmente no existen mejoras en el rendimiento.

Las diferencias entre una organización ganadora y una perdedora, es que las primeras saben hacer su trabajo mejor y organizarse en torno a los procesos, pero el problema radica en que estamos entrando al siglo XXI con organizaciones diseñadas en el XIX para que funcionaran en el XX.

La re-ingeniería de procesos de negocios propone abandonar procedimientos que se establecieron hace muchos años, y llevar a cabo una revisión del trabajo que se requiere para llegar a un producto o servicio final. Al rediseñar se tienen que hacer a un lado los sistemas viejos y empezar de nuevo, inventando una manera mejor de hacer el trabajo y utilizando técnicas o herramientas previstas para este fin.<sup>1</sup>

#### **Definición formal de Re-ingeniería de Procesos de Negocios:**

"Re-ingeniería es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras espectaculares en medidas críticas y contemporáneas de rendimiento, tales como costos, calidad, servicio y rapidez".

Con la definición anterior observamos que la re-ingeniería debe determinar primero qué debe hacer una compañía y luego cómo debe hacerlo, olvidándose por completo de lo que es y tomando en cuenta lo que debe ser.

Se debe llegar a la raíz de las cosas, no hacer cambios de manera superficial, no arreglar lo que ya está, descartando todas las estructuras y los procedimientos existentes e inventar nuevas formas de realizar el trabajo.

Se deben de dar saltos gigantescos con la reingeniería , en cuanto a rendimiento y se debe de utilizar solamente cuando haya necesidad de rediseñar todo.

A pesar de la importancia que tiene la informática en la re-ingeniería de negocios, como herramienta para llevarla a cabo, debe quedar claro que re-ingeniería no es automatización, ya que esta última sólo nos ofrece maneras más eficientes de hacer el trabajo.

---

<sup>1</sup> Conferencia del UNITE en Nashville, Tennesse, 1994



"No debemos confundir la re-ingeniería de procesos de negocios con la llamada reingeniería de software, que significa reconstruir sistemas obsoletos de información con tecnología más moderna. La re-ingeniería de software no produce otra cosa que no sean sofisticados sistemas computarizados que automatizan sistemas obsoletos".<sup>2</sup>

**Reconstrucción de los Procesos:**

La re-ingeniería de procesos de negocios afirma que para hacer frente a las demandas de calidad, servicio, flexibilidad y bajo costo, los procesos deben ser sencillos, algunas de las características de los procesos rediseñados son:

- Varios oficios se combinan en uno
- Los trabajadores pueden tomar decisiones
- Los pasos del proceso se ejecutan en orden natural
- Los procesos tienen múltiples versiones, no hay estandarización.
- El trabajo se realiza en el sitio razonable
- Se reducen las verificaciones y los controles son diferidos
- La conciliación se minimiza.

---

<sup>2</sup> Michael Hammer and James Champy "Re-ingeniería" Pág. 50

## **METODO PARA REALIZAR LA RE-INGENIERIA DE PROCESOS DE NEGOCIOS**

**En forma general la Re-ingeniería de procesos de negocios se realiza de la siguiente forma:**

- Evaluación de las condiciones de la empresa
- Mobilización enfocada en los recursos
- Proceso de reingeniería
- Desarrollo de sistemas de soporte

**Evaluación de las condiciones de la empresa.** En este paso se determinan los procesos primordiales de la empresa, realizando mediciones de los recursos empleados para su realización (máquinas, hombres, tiempo, inversión)

**Mobilización enfocada en los recursos.** En este paso se realizan cruces entre las funciones y las tareas que realiza la empresa para llegar a entender los procesos presentes en ésta.

**Proceso de Re-ingeniería.** Se definen los objetivos de la empresa y los medios para lograrlos.

**Desarrollo de sistemas de soporte.** Se denota la arquitectura sobre la cual se van a desarrollar los sistemas de soporte, se realiza un diseño y se especifican los requerimientos necesarios.

**Implementación operacional de los procesos.** A partir de la descripción de los procesos determinados en el proceso de la reingeniería y con la infraestructura definida en los sistemas de soporte se realiza la implementación operacional, definiendo prioridades para cada una de las etapas.

Para asegurar el éxito de la Re-ingeniería de procesos en negocios se requiere:

**Compromiso de la alta Dirección.** Lo cual permite que toda la empresa participe y se encuentre con la apertura necesaria para tener acceso a toda la información que involucra este proceso.

**Involucrar a expertos en el proceso del negocio.** Con la finalidad de facilitar el análisis de las funciones y de contar con ideas actualizadas de los diversas formas de realizar las tareas.

**Acceso a todos los Gerentes de Areas funcionales.** Para poder estar en posibilidades de evaluar las condiciones de la empresa y entender los procesos actuales.

**Metas y objetivos del negocio claros y aceptados.** Permite que todo el personal participe de forma positiva en el proceso de re-ingeniería.

**Comprensión completa de los procesos del negocio actuales y propuestos.** Se asegura que todo el personal se encuentre ubicado conciente de su participación.

**Comunicación continua para informar claramente la situación de la re-ingeniería.** Hace abierto y del conocimiento de todos como se está desarrollando el proceso de re-ingeniería y no lo etiqueta como un proyecto secreto y que pueda dañar los intereses personales.

**Compromiso de todos los miembros de la organización para implementar y continuar con los cambios.** Si todos no se encuentran comprometidos eso indica que existen puntos de resistencia y rechazo al cambio, lo cual incrementa en forma considerable los riesgos.

El equipo de re-ingeniería debe estar integrado por:

**Equipo de la alta Dirección.** Permite redefinir en forma dinámica los objetivos, y llevar a cabo la toma de decisiones en forma más clara, así mismo también indica de una forma abierta el interés de ésta en el proceso de la re-ingeniería, dando como resultado una alta participación.

**Directivos de todas las Areas del negocio.** Asegura el tener acceso a toda la información de la empresa de forma rápida al no tener que hacer memorándum para la solicitud de datos o que estas solicitudes sean relegadas a prioridades menores. Durante el diseño de los procesos propuestos se garantiza no transgredir los intereses, deberes y obligaciones de cada una de las Areas, permitiendo llegar a un consenso uniforme y sólido, del cual se disminuyen considerablemente los cambios.

**Grupo de trabajo interdisciplinario.** Revisa los procesos desde enfoques muy diversos permitiendo generar una solución real desde diferentes aspectos.

**Analista del negocio.** Son las personas que recopilan la información de los procesos actuales y en etapas posteriores transcriben los resultados y recomendaciones de los diversos grupos de trabajo.

El método para realizar la re-ingeniería de procesos de negocios es:

**Identificar las metas y objetivos estratégicos del negocio.** Permite canalizar los recursos y esfuerzos en lo primordial, disminuyendo las desviaciones.

**Identificar las áreas funcionales y los procesos propios del negocio.** Enfoca directamente la atención hacia lo fundamental de una empresa.

**Identificar responsabilidades.**

**El modelado del proceso debe contemplar:**

**Múltiples niveles de complejidad y simplicidad del modelo.** Permite realizar análisis más rápida y eficientemente al tener diversos tipos de detalle de acuerdo al nivel en que se esté trabajando.

**Identificación del ámbito.** Al tener claramente definido el alcance de la re-ingeniería se ahorra tiempo al eliminar labores fuera de ese ámbito, y permite dejar claramente establecido desde un inicio las perspectivas que se tienen con este proceso.

**Para planear el cambio se tienen que hacer diversas preguntas como son:**

- ¿La funcionalidad actual es todavía requerida?, ¿Qué se tiene que hacer?
- ¿Los beneficios del proceso justifican sus costos?

**Las etapas para lograr una re-ingeniería de procesos de negocios satisfactoria son dos:**

**Primera :** Consiste en determinar con mucha precisión los requerimientos y necesidades de sus clientes. Lo anterior implica, definir, junto con ellos, la calidad del diseño.

**Segunda :** Consiste en ver cómo se va a crear una herramienta tecnológicamente poderosa que haga posible satisfacer, de manera diferente y con excelencia, las necesidades y requerimientos de sus clientes. Esto permite a la organización obtener una ventaja sobre sus competidores y, por lo tanto, hacerse de más y mejores clientes y así permanecer en el mercado.

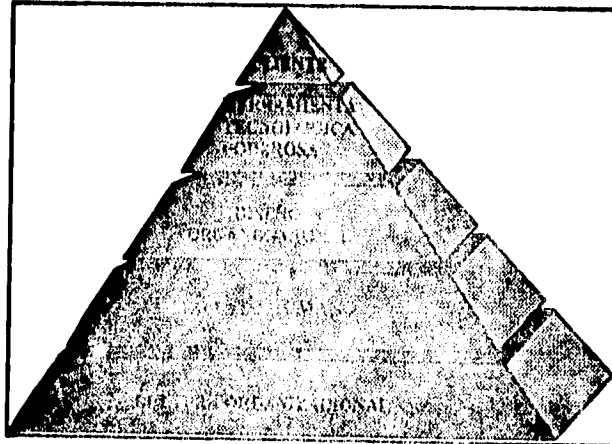
A partir de esta tecnología, se diseñará la nueva estructura, que deberá estar orientada al proceso innovado, es decir quedarán eliminadas las organizaciones diseñadas en forma divisional o departamental.

El proceso requiere de la elaboración de una cadena de clientes y proveedores internos en la que cada eslabón se planteará las siguientes preguntas:

- ¿Por qué estoy haciendo esto?
- ¿Quién es mi cliente?
- ¿Por qué lo estoy haciendo de esta forma y no de otra?
- ¿Cómo lo realiza la competencia?
- ¿Qué puedo hacer por mejorarlo?

La respuesta a estas preguntas darán como resultado que las empresas se vuelvan a orientar hacia la satisfacción de las necesidades de sus clientes.

A continuación se presenta una pirámide con las etapas de proceso de la Re-ingeniería de procesos de negocios.



*Fig. No. 4*  
*Etapas de Re-ingeniería de Procesos*

## RE-INGENIERIA DE SOFTWARE

### Antecedentes de Re-ingeniería de Software:

En los pasados 25 años, se tenían provistos sistemas profesionales con una interminable variedad de tecnologías de desarrollo para ayudar a incrementar la productividad. Estas tecnologías avanzadas incluyen lenguajes de 3a. generación en los años 60's, ciclo de vida de las metodologías y técnicas estructurales en los 70's e ingeniería de la información y herramientas CASE en los 80's. Ahora orientación a objetos, reusabilidad, generación de código y re-ingeniería (junto con las próximas generaciones de herramientas CASE) tienen las mejores oportunidades de proveer significantes ventajas en los Sistemas de Información.

En los negocios profesionales, están de acuerdo que la tecnología de información es una de las llaves para tener éxito y ser competitivo. Asimismo esta competitividad requiere el uso eficiente de los recursos existentes. Dos puntos claves en los recursos de Sistemas de Información son: La producción de aplicaciones comunes y los empleados quienes desarrollaron sistemas y ahora les dan mantenimiento.

Muchas organizaciones tienen inversiones significantes en el software de moda. Aplicaciones base, llamadas también sistemas legados, usados para operar negocios y proveer competitividad, pero son recursos costosos. En suma, los administradores de sistemas han adquirido una riqueza de información acerca de esos sistemas, subrayando las principales organizaciones.

Inicialmente, una aplicación es seleccionada, por que se requiere modernización en la empresa en cuanto a técnicas y términos funcionales. Si la re-ingeniería de una aplicación, es de primera vez en una organización, la selección del sistema es crítica. Un sistema es seleccionado sólo si es importante para la organización y puede ser rediseñado en pasos definidos. Cada paso tiene un deliberado objetivo que puede ser implementado y proveer beneficios inmediatos al usuario.

**El principal objetivo de la re-ingeniería de software** es obtener el control lógico de los programas y así realizar la reestructuración del flujo de control y del flujo de datos, y hacer a los programas entendibles para su fácil mantenimiento y uso. El tiempo de vida del software se extiende y las posibilidades para que el software se reutilice pueden ser identificadas. La re-ingeniería ofrece mayor rendimiento, una mejor comprensión, un mejor mantenimiento y un mejor desarrollo del software.

**DEFINICIONES DE RE-INGENIERIA DE SOFTWARE**

Para empezar, debemos comprender lo que significa el término "Re-ingeniería de Software". Todos estamos de acuerdo en que re-ingeniería de software es la examinación y alteración del código existente con el propósito de reorganizar dentro de una nueva forma, lo ya existente<sup>3</sup>.

La re-ingeniería de software es una actividad que provee el entendimiento del software o que proporciona mejoras al software para así incrementar su mantenibilidad y reusabilidad.

En la anterior definición, el término software incluye código fuente, documentación, gráficos y análisis acerca del código fuente, diseño, especificación de datos y otra documentación soportada directamente por el desarrollo y mantenimiento del software.

La definición incluye actividades diseñadas para mejorar la calidad del software.

Existen diferentes definiciones de re-ingeniería de software: <sup>4</sup>

"El proceso de modificar los mecanismos internos de un sistema o programa, o de las estructuras de datos de un sistema y/o programas con cambios funcionales."

"La examinación y alteración de un sistema para reconstituirlo en una nueva forma y subsecuentemente hacer la implementación."<sup>5</sup>

Existen una gran variedad de términos relacionados con las diferentes definiciones de la re-ingeniería de software, pero que sin embargo no son universalmente aceptados, dichos términos son:

- Mejoramiento
- Renovación
- Modernización
- Ingeniería de redesarrollo
- Reanudación

Los términos mejoramiento, reanudación y redesarrollo son similares y proveen al software de evolución y usos futuros.

La modernización incluye mejoras de software, pero muchas veces afecta las actividades de desarrollo y mantenimiento de software.

---

<sup>3</sup> Reverse Engineering and Design Recovery: a taxonomy, de Eliot Chikofsky, IEEE software Magazine, 1990.

<sup>4</sup> Proyecto GUIDE

<sup>5</sup> Chikofski y Cross, lo definen así

Se dice que un buen camino para predecir el futuro es estudiar el pasado. La terminología es el obstáculo número uno. Es importante primero que nada clarificar algunos términos. Vamos a explicar algunos conceptos en donde puede existir alguna confusión.

**Re-ingeniería de software** tiene el objetivo de crear sistemas más factibles de cambiar. Ejemplos de re-ingeniería de software incluye racionalización de nombres de datos, reestructuración de código y lenguajes de alto nivel .

El término **ingeniería reversiva** se refiere a la existencia de sistemas variando sus niveles de especificación.

El término **ingeniería hacia adelante** significa ir de una especificación a un sistema operacional.

La terminología global de las actividades de la ingeniería es llamada **redesarrollo de ingeniería**.

La necesidad de redefinir estos términos es para discutir el impacto organizacional en las industrias para entender la aceptación de la misma.

Los principios de la re-ingeniería nos proveen la manera de utilizar ventajas, y nos proporciona una estructura para poder modernizarnos además de una revisión de las especificaciones de la re-ingeniería para una plataforma independiente , esto es importante a considerar en el impacto que se tiene en las organizaciones., ya que los cambios implican desde el principio, que las decisiones afecten algunas áreas del negocio y para esto los administradores de sistemas deben dar el soporte requerido.

La tecnología de la re-ingeniería del software soporta tres temas:

- 1) Entendimiento del software
- 2) Mejoramiento del software y
- 3) Obtención, preservación y extensión del conocimiento acerca del software.



## EL CONTEXTO DE LA RE-INGENIERIA DE SOFTWARE

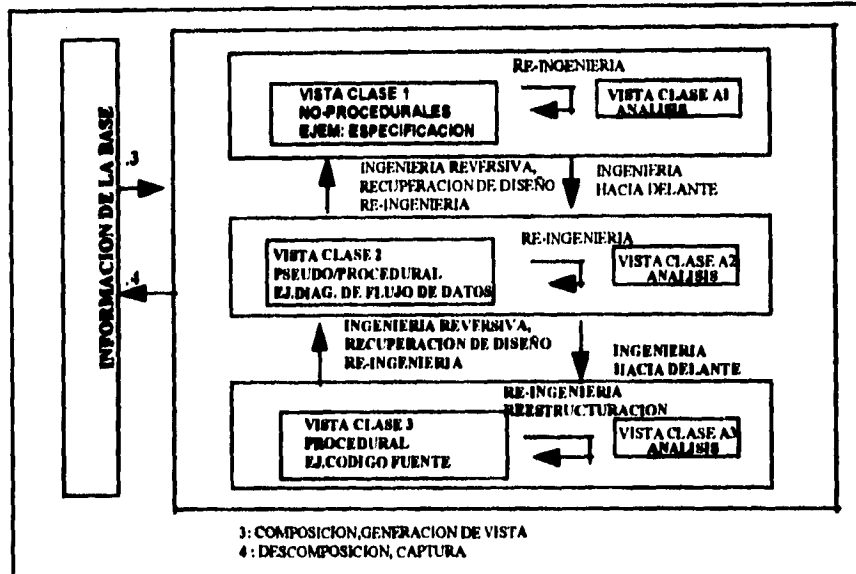


Fig. No. 5

*Re-ingeniería y términos relacionados*

En la figura anterior son cinco los conceptos o ideas que se muestran:

- 1.- vistas de software
- 2.- información de la bases
- 3.- composición
- 4.- descomposición
- 5.- transformación

Una **vista del software** es una representación del mismo o un reporte acerca del mismo, estas vistas pueden ser o no contempladas por el usuario, pero típicamente se refiere a una representación interna. La palabra vista se refiere a la forma en que puede verse un sistema por ejemplo un diagrama de flujo de datos.

Como ejemplos de vistas de software, se tienen las especificaciones de datos, código fuente, reportes derivados desde el análisis del código fuente estático y estadísticas de datos utilizados en las características del funcionamiento del software.

La Figura 5 puede ser agrupada en 4 diferentes clases o tipos de vistas:

- 1.-Clase de vistas no procedurales
- 2.-Clase de vistas pseudoprocedurales
- 3.-Clase de vistas altamente procedurales
- 4.-Clase de vistas de análisis que pueden acompañarse de otras vistas

La clase 1 contiene vistas que son no procedurales tales como esquemas o tablas de decisión.

La clase 2 se refiere a la descripción de diseños de lenguajes de programas y diagramas de arquitectura de software como por ejemplo los diagramas jerárquicos o diagramas de flujo de datos.

La clase 3 incluye información asociada o derivada de otra información representativa; código fuente, programas, datos, definición de datos en código fuente, objetos y relaciones descompuestas desde diferentes vistas y árboles de sintaxis se incluyen en esta clase.

La clase 4 contiene vistas de análisis derivadas de otras vistas o clases, por ejemplo las métricas de software son derivadas de analizar el software.

La base de información es el lugar donde se encuentra todo acerca del software. Esto es en tres pasos:

- (1) Descomposición del software en objetos y pequeñas relaciones
- (2) Incrementos en la construcción de objetos y relaciones
- (3) Información importante de otra información base

La descomposición es el proceso de transformación de una vista en objetos y relaciones en la base de la información.

La composición genera vistas de información de la base de información. Esto se hace ensamblando vistas de información en objetos y relaciones adicionando vistas formateadas igual a las necesidades que se muestran en las vistas de información.

La transformación es un concepto central. En la figura 5, la transformación de la re-ingeniería, es la información de una vista de software dentro de otra vista de información, al igual que una clase de vista prematura.

Ejemplos de transformación son: Reestructuración del código fuente, correcciones a especificaciones, etc.

**EL CICLO DE LA RE-INGENIERÍA DE SOFTWARE:**

Pasos que constituyen el ciclo de la re-ingeniería de software:<sup>6</sup>

1).- **Captura.**- Toma el código fuente y lo categoriza dentro de la definición de datos, toma el flujo de datos y de procesamiento de datos.

2).- **Ingeniería Reversiva.** - Son cambios de la información, desde un nivel de código al próximo nivel más alto. Este cambio no es simplemente uno por uno, sino que los sistemas se modifican desde la representación de un modelo de tecnología, a un modelo de empresa, produciendo una descripción unificada de las reglas del negocio.

La Ingeniería Reversiva es el proceso de recapturar lo esencial del diseño, estructura y contenido de un SISTEMA DE APLICACION, que sí corre y del cual se obtienen buenos resultados. Cuando el SISTEMA DE APLICACION ACTUAL es definido y entendido la RE-INGENIERIA consiste en el proceso de reescribirlo, aprovechando las ventajas de la última tecnología como son: las bases de datos relacionales, distribuidas, orientadas a objetos, redes, sistemas distribuidos,arquitecturas cliente/servidor, entre otras; o bien se le da mantenimiento a todas las funciones destacando el resultado que es de utilidad en el SISTEMA DE APLICACION ACTUAL. Todo esto se integra, por lo que se puede decir que para que exista RE-INGENIERIA DE SOFTWARE primero tiene que haber Ingeniería Reversiva.

Los avances de la Ingeniería Reversiva han sido significativos, a tal grado que han dado como resultado muchos métodos que están disponibles y que se aplican a las diferentes fases de la RE-INGENIERIA DE SOFTWARE.

En el problema de la reimplantación de un SISTEMA DE APLICACION programado en COBOL y que se tiene que reimplantar a un lenguaje de 4a. generación y que por lo tanto se tiene que generar nueva documentación, una solución típica es que a través de una herramienta se haga el traslado directo del código fuente en Cobol a un lenguaje de 4a. generación, obteniendo en forma automática la documentación del SISTEMA DE APLICACION NUEVO.

Otra alternativa puede ser aplicar la Ingeniería Reversiva al código fuente en Cobol y a la documentación existente, obteniendo el diseño de la información. Este diseño recuperado se usa para implementar una versión nueva del SISTEMA DE APLICACION, pero escrito en lenguaje de 4a. generación y desde luego que también se genera la documentación de éste.

El proceso de recuperación de diseños de SISTEMAS DE APLICACION es un elemento de la Ingeniería Reversiva del software.

La Ingeniería Reversiva de Software es definida como el proceso de análisis de un sistema para identificar los componentes del mismo y sus interrelaciones con otros sistemas

---

<sup>6</sup> Charles Bachman, fundador de Bachman Information Systems, Inc.

y crear representaciones de éste en otra forma o en un alto nivel de abstracción. Hay algunos tipos de información de SISTEMAS DE APLICACION que pueden ser abstraídos y examinados, resultando de utilidad al abstraer registros del SISTEMA DE APLICACION ACTUAL.

**Cómo usar la Ingeniería Reversiva.**

El problema de la reimplementación de un SISTEMA DE APLICACION ACTUAL en un lenguaje de programación existente se puede solucionar tomando en cuenta los siguientes puntos:

- Escribir manualmente el SISTEMA DE APLICACION ACTUAL.
- Usar un traductor automático de lenguajes
- Rediseñar y reimplementar el SISTEMA DE APLICACIÓN.

**Procedimiento de la Ingeniería Reversiva.**

El procedimiento de la Ingeniería Reversiva comienza extrayendo el diseño detallado, así como la abstracción de alto nivel de éste. El diseño detallado es extraído del código fuente y de la documentación que se tiene, esta información incluye: la estructura de diagramas, la descripción de datos y la de los procesos.

**Recolección de la Información.** Identificar el personal que tiene experiencia en el software para que recolecte toda la información posible, acerca del SISTEMA DE APLICACION. La información incluye código fuente, documentación del diseño, documentación de las interfaces del sistema y de rutinas externas.

**Examinar la información.** Revisar la información recolectada, para familiarizar al recolector con el SISTEMA DE APLICACION EN CUESTION.

**Extracción de la Estructura.** Identificar la estructura del SISTEMA DE APLICACION y utilizarla para crear diagramas de la misma. En el diagrama estructurado que se obtiene cada nodo corresponde a una rutina en el programa.

**Funcionalidad de los registros.** Determinarla en el diagrama estructurado por cada nodo.

**Flujo de Datos de registros.** La recuperación de la estructura de los SISTEMA DE APLICACION debe ser analizada para identificar la transformación de los datos. Los pasos de la transformación, muestran el proceso de los datos en el programa.

**Control del Flujo de los Registros.** Identificar el nivel más alto de control de la estructura del programa y sus registros, utilizando diagramas de control de flujo.

**Revisión de Diseño Recuperado.** La revisión del diseño recuperado se hace para tener una consistencia de la información útil y consistente y para encontrar las posibles

correcciones. Además se pueden identificar campos extraños de la información y verificar que el diseño sea una representación correcta del programa.

Generación de la documentación. El paso final es generar la documentación del diseño, explicando el propósito del programa, su historia, etc., etc., ya que necesita que sea registrado. Esta información probablemente no está contenida en el código fuente, por lo que debe de ser obtenida de otras fuentes.

En la siguiente figura se muestra un diagrama del proceso de Ingeniería Reversiva.

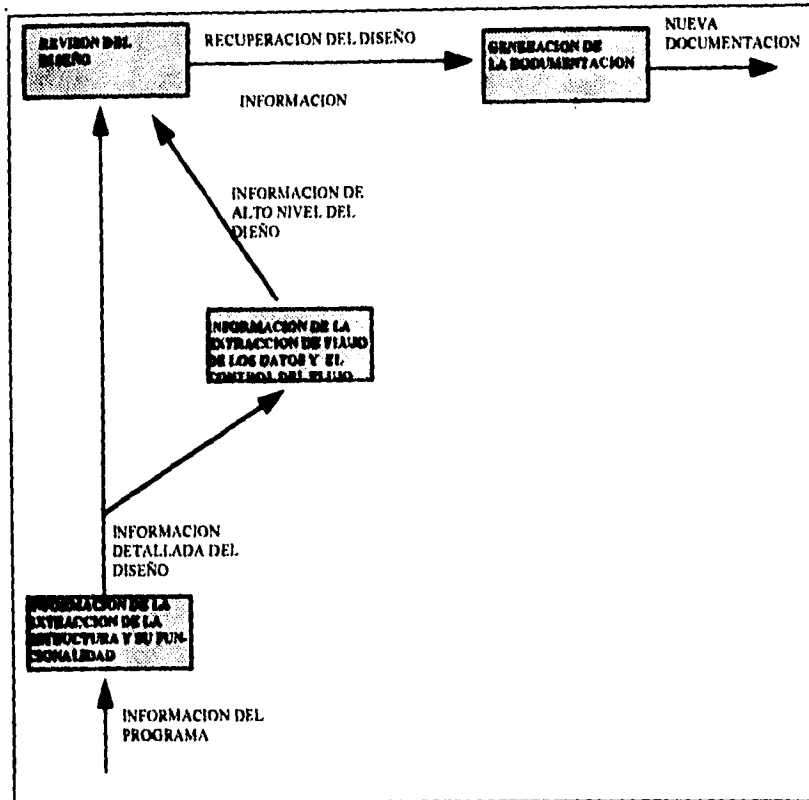


Fig. No. 6  
Proceso de la Ingeniería Reversa

La ingeniería reversiva es necesaria para crear un software flexible, incrementar la productividad, reducir costos y documentar sistemas. Esto ayuda a entender mejor el negocio y los datos, así mismo los pequeños procesos, dando la oportunidad de revisar la calidad del sistema al mismo tiempo que se hacen correcciones de posibles errores existentes. Se requiere de expertos en sistemas, siendo un desafío para ellos. Una de las grandes ventajas de la ingeniería reversiva es que los usuarios involucrados tienen una gran participación en el proceso del desarrollo del sistema.

En la Ingeniería reversiva se debe ser más cuidadoso en el ciclo de desarrollo. Esto significa, que es más efectivo cuando se ha definido bien la ingeniería hacia adelante y se puede comparar la ingeniería reversiva hacia fuera con el generador de código. El final de la re-ingeniería llega a ser un componente en un ambiente de proyectos. Tres principales usos de la ingeniería reversiva son : proveer una mejor metodología para el mantenimiento que

permita una adaptación rápida entre los requerimientos señalados, y la implementación para tomar ventaja de los cambios de tecnología y de arquitecturas de sistemas.

En empresas que tienen sistemas heredados, que están funcionando fuera de tiempo, pobremente documentados y con tecnología obligada y obsoleta, utilizando herramientas CASE se enfatiza la necesidad de la re-ingeniería.

**3).- Expansión.** Comprende tareas de salida de pasos de la ingeniería reversiva, así como las modificaciones de las reglas del negocio de acuerdo a nuevos requerimientos.

**4).- Ingeniería hacia adelante.** Abarca reglas de negocios de una empresa adheriendo los detalles tecnológicos. Esta es la inversa de los pasos de la ingeniería reversiva.

**5).- Optimización.** Se logra usando métricas, para obtener mejoras en el sistema, aplicando métodos apropiados de procesamiento.

**6).- Generación.** Son tareas del formato interno y creación de código legible-compileable. En este punto el sistema está abierto otra vez para la re-ingeniería.

### INICIANDO UNA RE-INGENIERIA

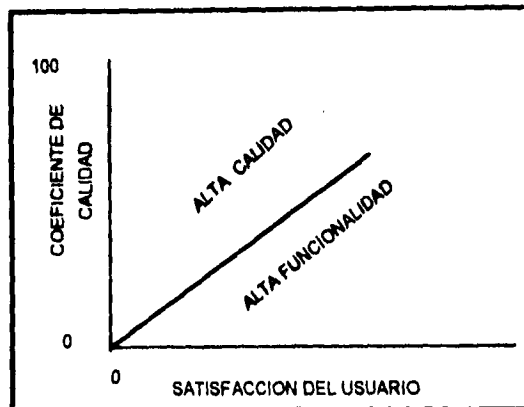
**Re-ingeniería de Software:** La estrategia para llevar a cabo un re-ingeniería del software existente; consiste en:

- 1.- Seleccionar los sistemas que se están utilizando y que se utilizarán en los próximos 5 o 10 años.
- 2.- Determinar el costo anual de mantenimiento de los sistemas, incluyendo corrección de errores y adaptaciones.
- 3.- Asignar prioridades a los sistemas
- 4.- Hacer una estimación del costo de la re-ingeniería de los sistemas seleccionados en el punto 1.
- 5.- Comparar el costo de mantenimiento contra el costo de re-ingeniería
- 6.- Calcular en qué tiempo, se va a recuperar la inversión de la re-ingeniería
- 7.- Considerar todos los aspectos intangibles, como son mejoras en rendimiento e interfaces con el usuario.
- 8.- Conseguir la aprobación de la Dirección.

#### Criterios para seleccionar sistemas aptos para la Re-ingeniería de Software:

Para seleccionar un sistema al que se le quiere hacer re-ingeniería se deberán tomar en cuenta diversos factores.

Una selección correcta sería evaluando el sistema de acuerdo a la calidad técnica y a la satisfacción del usuario: Esto lo podemos ver representado en la sig. figura No. 7:



*Fig. No. 7*  
*Evaluación del sistema*

La re-ingeniería también denominada renovación, no sólo nos ofrece el poder de recuperar la información del diseño del sistema existente, si no que a través del uso de ésta ayuda a modificarlo o reconstruirlo para mejorar su calidad.

El resultado de la re-ingeniería de software ayuda a reimplementar la función del sistema existente, añadiendo funciones y mejorando su rendimiento.

En las empresas, el no poder aplicar la re-ingeniería a todos los sistemas, se debe a que:

- Algunos sistemas no se usan con frecuencia
- Las herramientas de re-ingeniería aún se encuentran en su infancia
- Las herramientas pueden realizar re-ingeniería sólo a una clase limitada de aplicaciones
- El costo puede ser muy grande

La re-ingeniería de software aprovecha la parte de extracción de análisis y de diseño de la ingeniería inversa, además tiene la capacidad de reestructuración de los datos, la arquitectura y la lógica del sistema. La reestructuración que es una técnica propia de la re-ingeniería, sirve para obtener un diseño que produzca la misma función que tenía, pero con una mayor calidad que la del sistema original.

Muchos de los sistemas existentes están en mainframes y desarrollados en COBOL, en ambiente host, por ejemplo, CICS, IMS o DB2. Para los profesionales esto puede ser productivo, ya que es importante el uso de su talento y experiencia.



### ANALISIS DEL SISTEMA

Una vez seleccionado el sistema, se lleva a cabo un análisis del mismo. El resultado de ese análisis determina los 4 objetivos de la re-ingeniería de software:

- 1) Estabilizar la aplicación para prevenir los deterioros futuros.
- 2) Solidificar las aplicaciones en términos técnicos y funcionales.
- 3) Utilizar métodos estructurados de transformación (nuevas tecnologías y arquitectura).
- 4) Proveer la documentación detallada de la aplicación, así como las estructuras de datos y procesos.

El análisis es ejecutado, utilizando una combinación de análisis automatizado y trabajo de escritorio.

El inicio de los procesos de re-ingeniería de software es la transformación metodológica de cómo combinar la visión de la empresa y la tecnología dentro de un todo coherente. Este método es actualmente usado como TOP-DOWN y BOTTOM-UP (visión técnica), resultando una transformación global de las organizaciones.

La entrada estratégica a la re-ingeniería de software, requiere de entender la visión de la empresa negocios. Es necesario conocer el sistema actual y qué es lo que se va a revisar, cuáles son las perspectivas que tiene el usuario con el nuevo sistema y cuál es el ambiente bajo el cual va a ser utilizado.

La entrada táctica, se requiere para analizar la composición del sistema existente, incluyendo datos, programas, procesos y la asociación de los requerimientos de hardware y software. En suma se deben conocer las estrategias de transformación a largo y a corto plazo.

### DETERMINACION DE FUNCIONES EN COMUN

Una vez que el plan ha sido formalizado y dado el plazo, el redesarrollo del sistema puede iniciarse. El análisis del sistema existente es para identificar las funciones que los sistemas viejo y actual o nuevo tienen en común

Debido a que estas funciones, pueden ser usadas repetidamente durante el proceso de desarrollo, con esto se determina la funcionalidad del código existente, el analista de sistemas identificará y catalogará todas las demás funciones, ya que algunas pueden ser excelentes para la generación de código, si es que están disponibles.

Una vez que el análisis funcional ha sido terminado, una copia de la información acerca del código existente y del sistema está disponible.

Debido a que uno de los objetivos de la re-ingeniería de software es conocer los procesos y el código, esto es fundamental, por lo tanto esta información es importante y crítica.

El análisis, nos revela las áreas de la aplicación que necesitan mejoras. Los problemas comunes incluyen código no estructurado, inconsistencia o redundancia de funciones, mal uso del lenguaje, tecnología obsoleta e interfaces malas.

#### **DETERMINACION DE DATOS COMUNES**

También se define la relación entre los datos, esto es datos comunes entre el sistema viejo y el sistema nuevo, pueden ser catalogados y analizados. Un banco de datos nos provee de un excelente medio para conocer los requerimientos de los datos del sistema, y nos dice cuáles datos están o no duplicados.

El análisis de la representación de los datos en el sistema indica qué mejoras pueden ser hechas. Los problemas comunes incluyen redundancia de los datos, inconsistencia en el uso de los datos y el acceso a tecnologías obsoletas y además la mayoría de las veces no existe documentación sobre éstos.

Por otro lado el modernizar la representación de los datos, nos facilita la cuidadosa revisión del código. Además, la llave de la estructura de los datos puede ser identificada, para saber cuándo y cómo pueden ser manipulados. También, los nombres de los datos y su definición pueden ser estandarizados. Finalmente los métodos de acceso a los datos pueden ser revisados así como el medio en el cual son obtenidos.

Una vez que las tareas anteriores, han sido terminadas, el código de la aplicación puede ser reensamblado. Pero para que la plataforma del código pueda ser ensamblada, hay que aislar los procesos.

Uno de los mejores usos de codificación para diversos ambientes son los programas de aplicación de interfaces que se encuentran actualmente. Esto es particularmente cierto para adquisición de datos y protocolos de comunicación. La estandarización alrededor de SQL y bases de datos que soportan acceso de datos distribuidos proveen una solución a los problemas de los datos.

Una solución a los problemas de acceso de los datos es la estandarización bajo Bases de Datos distribuidas y SQL, y para su adquisición en diversos ambientes se tienen muchos productos de aplicación de interfaces.

## AREAS DE APLICACION DE LA RE-INGENIERIA

### MIGRACION DE DATOS

La calidad y rendimiento de los sistemas de información son una estrategia importante en las empresas

El bajo rendimiento y mantenimiento del software no estructurado ha creado una crisis en diversas empresas. Se hace necesario un método efectivo para realizar la re-ingeniería del software existente y así conseguir los beneficios del diseño estructurado.

Muchas de éstas están en crisis debido a que sus sistemas de información tienen un bajo rendimiento y no se pueden conseguir las metas por las cambiantes necesidades del negocio. Estos problemas son ocasionados por el software no estructurado, en el cual los sistemas de información están basados.

La migración de datos es una de las partes más importantes de la re-ingeniería de software, y nos permite entender la importancia de los siguientes aspectos:

- a).- Acceso a la re-ingeniería de datos y mejoras en las bases de datos.
- b).- Experiencia en uso de herramientas CASE, para extraer modelos de datos desde el código fuente.
- c).- Acceso para la migración de datos desde un DBMS a otro DBMS.
- d).- Herramientas específicas que ayuden a la re-ingeniería de los datos y a la migración.

La re-ingeniería y migración de los datos es una parte importante de los sistemas de aplicación existentes que pueden ser de nuevo rediseñados con la ayuda de métodos y herramientas de software. La re-ingeniería y migración de datos alarga la vida de los sistemas existentes gracias a la estandarización de la definición de los datos y facilita la simplificación del código fuente. También provee un correcto modelo de datos y parte de un excelente punto en la modelación de datos, migración de la tecnología de base de datos y es un paso de preparación para la ingeniería reversiva.

Mucha información de los departamentos de sistemas se atasca en el mantenimiento de sistemas de aplicación antiguos, que son demasiado importantes para desecharlos, demasiado costosos para redesarrollarlos y muy difícil de ignorarlos.

El futuro de la re-ingeniería y migración de datos continúa en expansión, y los procedimientos de ésta pueden ser automatizados durante los años siguientes, permitiendo su evolución. La re-ingeniería de datos es más reciente y extiende a la re-ingeniería a otros enfoques.

La re-ingeniería y migración de datos es importante porque los problemas de los datos son muy extensos a través de las aplicaciones y muy arraigados con los sistemas. La historia

del desarrollo y mantenimiento de la definición de datos para sistemas de aplicación típicos abarca los siguientes aspectos :

- Cuando se crean por separado los elementos de datos, registros y archivos para cada sistema y no para cada programa , la integridad de los datos se ve forzada.
- La validación de los datos puede hacerse en procedimientos no compartidos.
- Docenas de programadores modifican programas sin conocimientos y sin la certeza deque otros programas y archivos son afectados.

En consecuencia los problemas de los datos se extienden. Portafolios de aplicaciones típicas contienen documentación de programas, con años de no actualizarse, otros programas contienen miles de elementos de datos, algunos otros redundancia, inconsistencia o son demasiado incomprensibles y además tienen millones de líneas de código, algunos son obsoletos y otros erróneos.

Utilizando herramientas CASE, los problemas de los datos se pueden evitar en los nuevos sistemas, pero la mayoría de las herramientas tienen poco poder para detectar o para corregir los problemas de los datos mencionados anteriormente, en los sistemas antiguos.

Reorganizando la definición de los datos desde el código fuente al diccionario de datos o a la modelación de los datos, las herramientas CASE proporcionan grandes facilidades para hacer algunas correcciones manuales, pero esto también permite la existencia de problemas graves.

La re-ingeniería de datos, provee un mantenimiento fácil de los sistemas existentes.

A continuación se discuten los métodos y herramientas para la re-ingeniería de los datos, con énfasis en la re-ingeniería de la definición de datos. Se hacen comparaciones entre procedimientos y re-ingeniería de datos, entre definición de datos y re-ingeniería de valores de datos y entre re-ingeniería de datos e Ingeniería de datos.

### **PROBLEMAS DE DATOS**

Los problemas de datos pueden ser divididos en dos clases:

- a) Problemas de definición de los datos
- b) Problemas de valores de los datos

Estos dos tipos de problemas raramente ocurren simultáneamente y las soluciones son diferentes. El enfoque en este apartado es mencionar las herramientas relevantes que están disponibles para la re-ingeniería de la definición de datos.

La re-ingeniería del valor de los datos es incluida para perfeccionar y es un indicador de investigaciones futuras.

a) Los problemas de la definición de datos son :

- Inconsistencia en el nombre de los elementos
- Longitud inflexible de los campos
- Inconsistencia en la extensión de los registros
- Diccionario de datos inexistente, incompleto o incorrecto.

Además de estos problemas, se tienen otros que afectan el desarrollo y mantenimiento de los sistemas tales como :

- Al adicionar nuevos campos o expandir el tamaño de los ya existentes, ocasiona que se requiera la modificación de los programas y convertir o alterar decenas de archivos. Identificar estos programas y archivos es difícil, pero implementar los cambios puede llevar muchos años, convirtiendo a los proyectos en multimillonarios. Parte de estas dificultades se ven desde el flujo de los datos, al utilizar sinónimos, alias y homónimos dentro del flujo de datos y entre programas.

- Deficiencias de auditoría que son causas de riesgo para las organizaciones, con los empleados y clientes, la inadecuada alteración de los datos, accesos no autorizados o inadvertidos, exponen a daños potenciales.

b) Los problemas con los valores de los datos son:

- Inconsistencia en los valores asignados
- Deficiencias para distinguir entre validación y valores perdidos.
- Valores negativos en campos que no siempre son negativos.
- Valores negativos o positivos cuando los signos se omiten.
- Truncamiento de dígitos significativos.
- Inconsistencia en el manejo de unidades numéricas.
- Cambios en las validaciones de los datos que no son hechos en los archivos estáticos o históricos.

**RE-INGENIERIA DE DATOS VS. INGENIERIA DE DATOS**

La re-ingeniería de datos y la ingeniería de datos, están relacionadas, pero son diferentes, tal como se muestra en la tabla No. 3, en la cual se compara a éstas en términos generales.

	<b>RE-INGENIERIA DE DATOS</b>	<b>INGENIERIA DE DATOS</b>
<b>ORIENTACION</b>	Sistemas antiguos	Sistemas nuevos
<b>FRECUENCIA</b>	Una vez para c/sistema	Proceso continuos
<b>ALCANCE</b>	Definición de datos Valores de datos	Definiciones de datos
<b>ALCANCE</b>	Completamente manual Semi automático	Comienzo en frío Comienzo en cálido
<b>EFFECTOS</b>	Los sistemas antiguos son más fáciles de mantener	Los nuevos sistemas proveen mejores clasificaciones de datos y son más fáciles de mantener

*Tabla No. 2*

*Re-ingeniería de datos vs Ingeniería de datos*

Mientras que la re-ingeniería de datos se refiere a los sistemas existentes, la ingeniería de datos se refiere a los sistemas planeados a futuro. La Ingeniería de datos puede estar basada en información reunida durante la re-ingeniería de datos, entonces la transformación del modelo de datos puede propagarse por todas las partes de el sistema antiguo. Después de eso un modelo de unificación del modelo de datos es compartido por el sistema viejo y por el sistema nuevo.

El acceso manual de la re-ingeniería está típicamente limitado a pocos archivos maestros. En contraste el semiautomático puede aprovechar las direcciones de sistemas completos, rápidamente y más a fondo que los accesos manuales. Completando un factor crítico que es el pasar por alto las definiciones de los datos siendo esto una causa de los problemas que se originan en los datos.

**RE-INGENIERIA DE DATOS: PASOS Y HERRAMIENTAS**

La tabla número 3, compara los pasos y herramientas que siguen utilizando, tanto en la Ingeniería como en la Re-ingeniería de Software. Sin embargo como se ve en la tabla, ambas son viables.

RE-INGENIERIA DE DATOS	INGENIERIA DE DATOS
PASOS: ANALISIS DEL CODIGO ANALISIS DE DATOS SINTESIS DE LOS METADATOS REDISEÑO REVISION DEFINICION DE LA EXPORTAC.	MODELACION DEL NEGOCIO MODELACION CONCEPTUAL MODELACION FISICA ESPECIFIC. PROCEDIMIENTOS GENERACION DE CODIGO
HERRAMIENTAS: DATEC	SILVER RUN
.ANALIZADOR DEL MEDIO AMBIENTE	.MODELACION DE LA ENTIDAD
.DEFINICION DE ESTANDARES	.MODELAR DATOS LOGICOS
.ANALIZADOR DE METRICAS	.DIAGRAMA DE FLUJO DE ESPECIFICACION DE LA EXPORTACION
.DEFINICION DE LA EXPORTACION	.GENERACION DE CODIGO

Tabla No. 3  
Comparativo entre pasos y herramientas

### ANALISIS DEL CODIGO.

El análisis del código es el descubrimiento de las definiciones, flujo y reglas de los datos, que existen en cualquier código fuente. Por ejemplo en un programa hecho en COBOL, las declaraciones de archivos y registros se encuentran en la ENVIRONMENT DIVISION y en la DATA DIVISION, el flujo de datos de los programas y archivos se encuentra en la PROCEDURE DIVISION, el flujo de datos entre programas se encuentra en la LINKAGE SECTION .

La validación de las reglas es mucho más difícil de definir o descubrir, por que éstas se encuentran por todas partes en la PROCEDURE DIVISION.

El análisis del flujo de datos para programas en lenguaje COBOL, es más difícil que para cada uno de los otros lenguajes existentes, porque :

- 1.- Las acciones en los elementos de datos elementales afectan a los grupos de campos.
- 2.- Las acciones en los grupos de campos afectan a los elementos básicos.
- 3.-Se crean múltiples conflictos en las definiciones para la localización del almacenamiento físico.

Si el código es analizado, un almacenamiento de metadatos es también imprescindible. El nivel de análisis puede depender del programador, ya que estos usualmente piensan en términos de datos que afectan a los programas.

Para poder llevar a cabo el análisis del código, se debe conocer previamente cuales son los programas que afectan a determinados archivos, así como también se requiere conocer con precisión la asociación o relación entre los programas y puntos de entrada, entre los

puntos de entrada y los módulos que son llamados, entre los módulos que son llamados y los trabajos (jobs) y entre los jobs y los archivos en cuestión.

Uno de los más graves problemas durante el desarrollo de sistemas se debe a la extensión o expansión de los errores en el código fuente, que se pueden originar al modificar una sola línea de código, ya que con el tiempo estos cambios resultan catastróficos. La importancia de lo anterior radica en que una de las razones por las cuales se dificulta el mantenimiento de software, es no conocer de qué manera los cambios en el código fuente pueden afectar algunas otras piezas del código y en general al sistema.

Hoy en día el mantenimiento del software tiene mucha demanda, es por eso que se hace necesario el conocimiento de herramientas o métodos que nos ayuden a comprender mejor los impactos que se tiene al realizar dichas modificaciones y facilitar así todos aquellos problemas asociados con la revalidación de datos y buscar la manera de reducir los recursos para las actividades de mantenimiento de software.

Al realizar un análisis del código fuente podemos conocer o saber si un programa no estructurado puede seguir siendo utilizado en los sistemas antiguos. Además de que nos ayuda a determinar el tipo de mantenimiento a aplicar (correctivo, adaptativo, perfectivo o preventivo).

El análisis del código fuente se puede realizar utilizando herramientas y métodos que nos ayuden a:

- Descomposición de los programas en dos o más componentes, por variables, por sentencias, o por código no utilizado.
- Aplicación de las modificaciones en el código, y conocer o saber como el manejo de las variables afecta las sentencias y saber si son independientes o no.
- La forma en que se van a realizar las modificaciones: adición, borrado y cambios, de tal forma que no se afecte el flujo de datos y el flujo de control.
- Probar los cambios
- Descubrir nuevos modelos de procesos de mantenimiento del software, para minimizar defectos y costos, mientras se maximiza la satisfacción de los usuarios.
- Conocer las direcciones futuras de las formas de descomposición de los sistemas de software, como un método a usar el personal de mantenimiento de los mismos.

En general el proceso de mantenimiento de software es:

- Identificación del problema.- Identificando el funcionamiento específico es más fácil adicionar o modificar el código.



- Entender los programas, identificando cada uno de los elementos de los programas que son importantes funcionalmente, para ser modificados y su papel dentro del sistema. Entender los programas incluye también entender como las especificaciones del diseño y el código fuente tiene relación entre si.

- Modificación de los programas.- Diseñar o implementar cambios a varias partes del código fuente, ocasiona que se vean afectadas otras funciones de los programas, cada vez que una modificación en el software es requerida el personal de mantenimiento de software debe identificar y evaluar diferentes alternativas de diseño, así como también examinar los impactos de la modificación.

- Re-validación de los programas.- Comprobar que los cambios están de acuerdo a lo especificado y que no se están cambiando o modificando otros programas que no sean los especificados.

- Re-documentación.- Cambiando la documentación relacionada a los programas, incluyendo todas las especificaciones y diseño al reflejar los cambios en los programas.

El proceso usual del mantenimiento del software, se muestra gráficamente en la figura. No. 12 PROCESO USUAL. y consiste en que el software es cambiado subjetivamente de acuerdo a prioridades pendientes. Los cambios están compuestos de dos partes: Entendiendo el código, lo cual requiere de documentación, lectura de código y ejecución. Entonces el programa es modificado; el personal de mantenimiento de software primero debe diseñar los cambios y después alterar el código, tratando siempre de minimizar los efectos. El cambio es válido y la alteración del código es verificada de acuerdo a las especificaciones requeridas; entonces el nuevo código es integrado con el sistema existente al estar de acuerdo con las especificaciones del sistema. Todo lo anterior requiere tareas de pruebas de regresión.

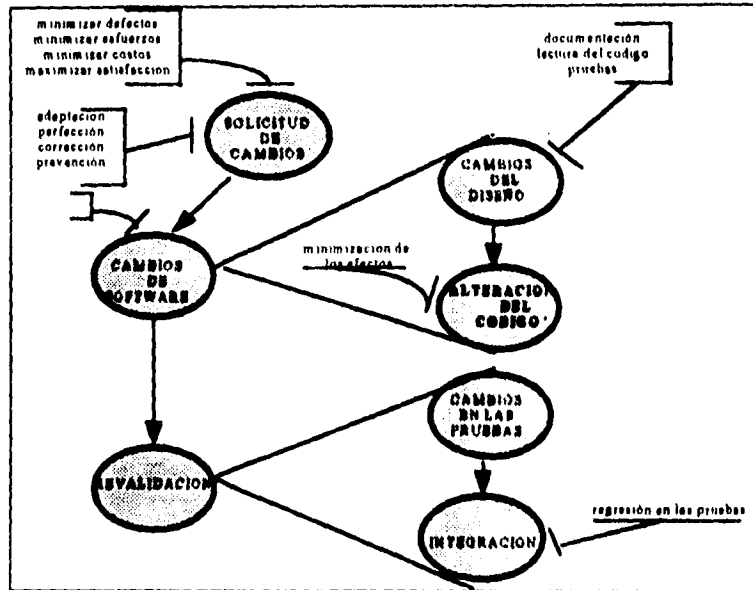


Fig. No. 8  
Modelo del Proceso Usual

Un nuevo modelo para el proceso del mantenimiento del software, se muestra gráficamente en la figura No. 9 NUEVO MODELO PARA EL PROCESO DEL MANTENIMIENTO DE SOFTWARE, el software es modificado sujeto a prioridades pendientes, los cambios se componen también de dos partes. Entender el código nuevo que requiere de documentación, lectura y ejecución y el uso de la descomposición. El código es entonces modificado, sujeto a las estructuras de lineamientos, los cambios son probados en la descomposición y la prueba de regresión es innecesaria. El personal de mantenimiento solo necesita verificar si los cambios son correctos; después aplicando la mezcla de algoritmos, el cambio del código fuente está completo.

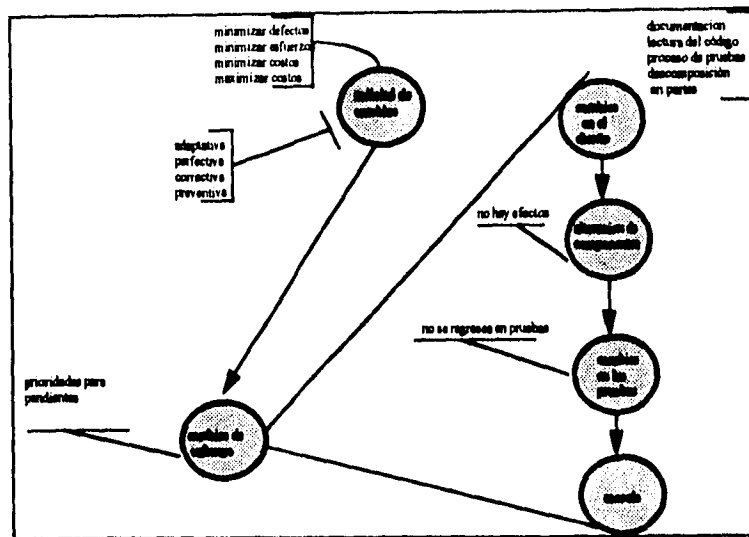


Fig. No. 9  
Modelo del Proceso Nuevo

La descomposición de los programas ayudan al personal de mantenimiento del software a entender mejor los siguientes tres pasos:

- Compresión.
- Depuración.
- Modificación.

La **compresión** del código fuente puede ser asistida por la documentación de las especificaciones del sistema, pero si esta es poca o esta incorrecta el problema es grave y el mantenimiento se hace lento.

La productividad del mantenimiento del software puede incrementarse utilizando enfoques desde el diseño y una efectiva técnica de descomposición. Estas técnicas pueden estar incorporadas en un programa interactivo analizador que provee de información acerca de la semántica estática y dinámica de los programas o código fuente.

La semántica estática corresponde a la información pertinente de la estructura y sintaxis de los programas; por ejemplo un programa puede descomponerse en módulos, dependiendo del lenguaje de programación cada módulo puede ser un procedimiento o una colección de procedimientos. Otros campos de interés incluyen a las variables estructuradas y tipos de variables en este contexto, la descomposición corresponde a la localización y despliegue de procedimientos tipos específica de variables.

La semántica dinámica se refiere a qué un programa puede visualizarse como una función abstracta que genera los valores de salida de las variables basadas en los valores de entrada especificados. La semántica dinámica de un programa corresponde con el comportamiento funcional de el mismo, esto es posible al analizar un programa y determinar el comportamiento dinámico mientras se ejecuta el programa.

Un programa puede también ser monitoreado mientras es ejecutado y la información puede ser recolectada. Si utilizamos el análisis dinámico, el programa actual y su ejecución puede ser utilizados para generar una precisa relación de dependencia.

La semántica dinámica es utilizada para:

- Identificar todas aquellas sentencias que son ejecutadas al producir el valor de salida de una variable.
- Seleccionar todas las sentencias que utilizan el valor de entrada de una variable específica.
- Encontrar todas las variables cuyo valor de entrada es usado para determinar el valor de salida de una variable específica.

El análisis del código fuente es fundamental en muchas de las herramientas utilizadas para la re-ingeniería de software. Analizar la estructura de los programas es una de las más frustrantes y tardadas de las partes del mantenimiento de software. Existen sistemas para la abstracción de programas diseñados para proveer los diferentes niveles de abstracción que pueden descubrirse a lo largo de los procesos del mantenimiento de software. Estos sistemas sumarizan la información de la estructura de los programas en la base de datos relacional. Los programadores pueden invocar consultas relacionales para analizar todos los aspectos del software. Sofisticadas herramientas ayudan a automatizar el desarrollo de software y las funciones de mantenimiento, utilizando las vistas relacionales.

La construcción de los sistemas para la abstracción de programas incluyen los tres siguientes pasos:

- Formar un módulo conceptual: Aquí se definen los objetos o partes de software y las relaciones, en un nivel de abstracción selectivo, esto nos sirve como una especificación de requerimientos para el abstractor de la información y determina la extensión de conocimientos disponibles en la base de datos.

- Extracción de vistas relacionales: Consiste en la construcción de análisis del texto del programa y extraer la información relacional de acuerdo al modelo conceptual. En este almacenamiento, la representación textual de los programas es convertido a una base de datos relacional.

- Construcción de vistas abstractas: Un conjunto de herramientas proveen procesos para construir las vistas relacionales, así como diferentes niveles de vistas abstractas. Una vista

abstracta muestra un aspecto de la estructura de los programas y elimina los detalles innecesarios.

Estos enfoques pueden ser aplicados en muchos lenguajes de programación y documentos estructurados.

El almacenamiento de la información de la estructura de los programas en una forma relacional ofrece al usuario el sofisticado poder a las consultas de los sistemas de bases de datos relacionales, además de que la combinación de los programas de bases de datos con las bases de datos relacionales para otras formas de documentación de software, pueden tener una natural extensión en el futuro..

Los sistemas de abstracción de programas se utilizan para conocer y estudiar los siguientes aspectos:

- Subsistema:- Identificación de los componentes del sistema.
- Diagramación:- Topológicamente ordena las diferentes combinaciones de relaciones.
- Código no utilizado.- Detecta el código fuente que no es utilizado en el sistema.

Existe una gran cantidad de sistemas para la abstracción de código fuente. La Tabla No. 5, muestra una breve descripción de algunos de éstos.

Un aspecto importante es la re-escritura y documentación de los programas existentes, que se pueden realizar utilizando diagramas gráficos, descripciones matemáticas y documentos textuales.

Los diferentes problemas acerca del mantenimiento de software y la importancia de técnicas efectivas para éste, han sido tratadas por diferentes autores (Heninger 1980, Lehman 1980, Martín 1983, Pressman 1987, Hazzah 1989, etc.), incluso un gran porcentaje de ellos coinciden en que los procesos de mantenimiento pueden resultar una gran ventaja para las organizaciones que utilizan grandes aplicaciones. Mejores ventajas en los procesos de mantenimiento de software pueden tener impactos importantes al reducir la inversión requerida para darle soporte al software, extendiendo la vida útil de los sistemas existentes y expandiendo el rango de problemas que pueden ser realmente identificados.

HERRAMIENTAS	DESCRIPCION
MASTER SCOPE	Utilizada en este ambiente de interlisp, origina desde un simple programa la impresión de la estructura del mismo, así como el análisis de la definición de las funciones que se llaman.
FAST (FORTRAN ANALISIS SYSTEM)	Puede re-analizar las partes modificadas de un programa y dar mantenimiento a la base de datos. Provee un conjunto de capacidades de análisis de programas en lenguaje Fortran. utiliza un sistema manejador general de base de datos.
OMEGA	El modelo conceptual de esta herramienta consiste de atributos a módulos, sentencias, nombres y relaciones entre otras partes u objetos del lenguaje Incorpora el diseño de un esquema relacional para un lenguaje como Pascal, utiliza INGRESS para administrar la base de datos y se pueden aplicar todos los operadores relacionales. Reconstruye el software objeto desde el programa de bases de datos. Almacena todos los detalles de la información de los programas tales como variables, expresiones y sentencias
CSCOPE	Esta diseñado para análisis del código fuente en lenguaje C
LESSONS	Es un sistema de abstracción de sistemas que ofrece una separación de extracción y presentación, modelo conceptual conciso que define como la información está almacenada en una base de datos, separación de bases de datos y códigos fuente y construcción de bases de datos incremental.

*Tablas No. 4  
Herramientas para el análisis del código fuente*

### **ANALISIS DE DATOS**

El análisis de datos, se refiere a las propiedades de los datos, se examinan los datos actuales y se comparan dichas propiedades con las definiciones de datos. En el curso del análisis de los datos se descubre que la definición puede ser incorrecta o imprecisa, así como los valores de los mismos.

Por instancia, si una definición permite valores numéricos negativos, pero en el contexto de la aplicación sólo tienen sentido valores no negativos, la definición debe ser revisada, y deben quedar igual los valores en el archivo y en la definición, permitiéndose también la conversión de datos.

En la práctica, es muy común encontrar definiciones que se contradicen con el valor real de los datos. Si un programa inicializa un campo con ceros y otro programa lo inicializa con espacios, un tercer programa que lo utilice puede producir resultados erróneos, dependiendo de cual de los dos programas se haya ejecutado más recientemente.

Comparando las definiciones y los valores actuales se encuentran imprecisiones y definiciones de datos contradictorias. Si un sistema continúa con su operación bajo esas condiciones, los pasos siguientes de la re-ingeniería no tolerarían dichas disparidades.

### **SINTESIS DE LOS METADATOS**

El objetivo es presentar el modelo de datos de la mejor manera posible, para esto se reúne la información recolectada en los pasos anteriores. La síntesis de los metadatos se realiza para determinar mejor la relación entre los datos y los programas, cuya relación se logra a través de la examinación de los sistemas en su totalidad.

Después de la síntesis, la revisión del almacenamiento de los metadatos nos proporciona un inventario de los datos fuentes y el flujo de éstos, así como las referencias entre programas, módulos llamados, puntos de entrada y archivos.

### **REDISEÑO**

El rediseño convierte la definición de los datos de una forma a otra. El objetivo primordial es la purificación o filtro de la definición de los datos, cambiando el modelo de datos. Algunas actividades del rediseño clarifican las definiciones de los datos, algunas veces se requiere de cambios en los datos existentes, otras veces no se requieren y otras solo se realizan cambios sustanciales.

El nivel de rediseño más simple es el que se refiere a la estandarización de los registros de datos, que consisten en crear las comparaciones de las definiciones de los registros, en donde se incluye el más alto nivel de detalles de las diferentes definiciones y se utilizan agregados de datos.

La estandarización reconcilia los formatos físicos diferentes y crea copias compartidas de estos en ese espacio.

El diseño de registros compuestos es muy común y lo realizan gentes que conocen a detalle la organización, y los datos, pero aún así es necesaria para ciertas tareas, la utilización de herramientas de re-ingeniería que examinan a mayor detalle los sistemas y sus datos.

Un nivel más complejo del rediseño, se refiere a la racionalización de los nombres de los datos, que consiste en crear nombres consistentes con todas las copias de los registros. Al hacer la racionalización de los nombres de los datos, el flujo de los datos puede ser trazado más fácilmente, tanto en el código fuente como entre programas.

La reestructuración de los datos, es uno de los niveles más complejos e importantes del rediseño, ya que tiene implicaciones significativas para la existencia de los datos porque estos cambios trascienden en el modelo físico. Por instancia los grupos repetidos pueden ser convertidos en tablas durante el proceso de normalización.

Cuando se realiza la migración a bases de datos la normalización es esencial, pero la reestructuración no está limitada a bases de datos relacionales, esta se puede aplicar también a modelos de datos jerárquicos. Si el rediseño se realiza, ya sea durante la re-ingeniería o durante la ingeniería de datos, se depende de varios factores como urgencia, disponibilidad de herramientas o gente, y de la extensión de cambios en las expectativas del sistema.

### REVISIÓN

La revisión implementa el rediseño del código fuente y algunas veces en archivos existentes. El objetivo es hacer los sistemas existentes de acuerdo a el rediseño. Resulta obvio encontrar herramientas para la re-ingeniería de definición de datos que actualmente revisan el código fuente, pero no por eso deja de ser importante el nivel de revisión manual que asiste a estos procesos.

La revisión del código fuente incluye lo siguiente:

- Reemplazo de las definiciones viejas con copias de registros estandarizadas y racionalizadas, con esto se hacen externas dichas definiciones desde el código.
- Cambios en las referencias anteriores en el código a los nuevos nombres
- Reemplazos literales con los nombres de las constantes, creando copias de registros o libros con esas constantes que son comunes a través de los programas.
- Cambios en los procedimientos conforme a la nueva definición de datos.



La revisión de los valores de los datos, también llamada conversión de datos, puede no ser requerida por dos razones:

- La estandarización de los nombres, no afecta en nada los valores de los datos y
- La revisión en los valores de los datos puede ser tediosa y larga por la cantidad de programas que tiene el sistema.

Cuando la revisión de los datos se hace, se incluye una conversión del formato físico y una validación de datos.

### **DEFINICION DE LA EXPORTACION**

La definición de la exportación hace a las definiciones de datos disponibles a otras herramientas CASE, para realizar diccionarios de datos, modelación de datos y generadores de aplicaciones.

### **MIGRACION A BASES DE DATOS RELACIONALES**

Los usuarios contemplan las conversiones de bases de datos generalmente desde una de las siguientes dos situaciones:

- Realizando la conversión desde sistemas que utilizan archivos a un administrador de bases de datos del mismo tipo.
- Realizando la conversión desde un administrador de base de datos, especialmente jerárquico o sistemas de red a un RDBMS.

Muchas de las conversiones que se llevan a cabo tomando en cuenta la segunda situación pueden ser divididas en dos subgrupos:

- Usuarios que cambian desde sistemas NON-IBM a DB2 y a bases de datos relacionales de IBM.
- Usuarios que cambian desde IMS y bases de datos IBM a DB2.

Las ventajas de las conversiones a bases de datos son:

- Reescribir la estructura de las bases de datos
- Utilización de la re-ingeniería de software
- Ingeniería reversiva
- Uso de nueva tecnología

**SEGUIMIENTO PARA OBTENER LA DOCUMENTACION DEL SISTEMA ACTUAL**

Para obtener la documentación del sistema actual es necesario utilizar la ingeniería reversiva, nuestra aproximación se encuentra basada en las siguientes teorías:

(1) La representación explícita de varios elementos estructurales y funcionales del código y múltiples relaciones entre estos ayuda a comprender el programa y de este modo al mantenimiento del software.

(2) Las representaciones requeridas pueden ser clasificadas dentro de dos componentes: estructural y funcional. El modelo estructural es una colección de entidades jerárquicas de los elementos estructurales tal como rutinas y bloques de código modelando el código. La vista funcional provee un perfil de la jerarquía, describiendo las funcionalidades y subfuncionalidades

## CAPITULO III

## TECNOLOGIA Y EVALUACION DE LA RE-INGENIERIA DE SOFTWARE

Hoy en día, los clientes requieren de herramientas capaces de ayudar en el manejo de sistemas antiguos, de tal forma que la información contenida en estos sistemas antiguos, pueda ser usada en proyectos de desarrollos futuros. El medio para realizar lo anterior radica en el almacenamiento.

Debemos saber qué tan real es lo anterior, si son suficientes las herramientas disponibles, para aprovechar eficientemente los objetivos que tiene el realizar almacenamiento automáticamente.

Los problemas con los sistemas heredados o legados, se resuelven tomado en cuenta las siguientes razones, para así justificar el porqué de alterar el código existente :

- Corrección de errores existentes
- Adaptación de nuevos requerimientos del negocio
- Aprovechar mejor las ventajas de la nueva tecnología

**Corrección de errores existentes:** Por ejemplo en el caso práctico, la corrección de errores es muy frecuente por que existe mucho código que no se utiliza, pero que afecta en los procesos debido a que los programas son muy antiguos (Cobol).

**Adaptación de los nuevos requerimientos del negocio:** Las necesidades cambiantes del negocio, ocasionan que la cantidad de requerimientos sea diferente, lo que implica que se le debe dar mantenimiento al código fuente. Por ejemplo en el Caso práctico diferentes interfaces con otros sistemas, han ido creciendo, por lo que el sistema requiere nuevos formatos y validaciones de los datos.

Cuando el usuario tiene necesidad de utilizar la información en equipo diferente en el que fue procesada, es necesario especificarlo desde el código fuente.

La oportunidad de la información implica una revisión del código fuente para eliminar todos los elementos que retrasan la entrega de la información, por ejemplo en el caso práctico que se estudia en esta investigación, en el sistema actual existe información de clientes que no ha sido actualizada, ocasionando que los procesos se lleven demasiado tiempo.

El sistema emite una gran cantidad de reportes, que no son realmente utilizados, lo que ocasiona pérdida de recursos como son: tiempo de proceso y papel, entre otros.

La información que contienen algunos de los reportes emitidos por el sistema, es utilizada sólo en parte, lo que implica que el usuario solicite que en un nuevo reporte se concentre la

información que realmente le interesa y que se encuentra en los diferentes reportes ya emitidos.

**Aprovechar mejor las ventajas de la nueva tecnología.** En el caso que tomamos es cambiar de un lenguaje de 3a. generación, como Cobol, a un lenguaje de 4a. generación, como LINC, aprovechando las ventajas de trabajar con Bases de Datos.

Por otra parte Dave Sharon <sup>1</sup> afirma que los problemas son multiplicados por el gran número de gente que interviene en la creación y modificación del código, ya que es prácticamente imposible poder conocer los requerimientos originales. Las tareas de automatización requieren a la larga de un trabajo humano intensivo.

Es un requerimiento absoluto entre usuarios, que las soluciones de re-ingeniería sean una parte de las herramientas CASE. Ellos deben ser capaces de hacer la migración de sistemas existente a nuevos ambientes.

Otro paso importante en un proyecto de re-ingeniería de software es realizar un análisis acerca de las herramientas disponibles para llevarla a cabo.

Los clientes de las herramientas de re-ingeniería, la mayoría de las veces, define cuáles son las dificultades que tienen con la re-ingeniería. Estas pueden ser varias cosas tal como la reusabilidad o la traslación.

No todos los sistemas son candidatos para la re-ingeniería de software, ya que existen varios grupos de aplicaciones y cada una de éstas tienen diferentes características, además de que el tamaño se refleja en los métodos de programación y en los estilos usados por cada compañía para el procesamiento de los datos.

Existen diferentes enfoques, para el aprovechamiento de la re-ingeniería en una empresa donde se tienen sistemas de aplicación muy variados, dichos enfoques son: reemplazo, expansión, duración y salvamento.<sup>2</sup>

Estos enfoques de re-ingeniería son importantes al desarrollar y se deben aplicar selectivamente de acuerdo al criterio de desarrollo.

#### **Herramientas de Re-ingeniería de Software**

Muchos clientes están interesados en la idea de poder modificar o cambiar el código a un nivel más alto de abstracción, lo cual es aprovechado por los distribuidores de herramientas de re-ingeniería.

Actualmente, no existe una herramienta de re-ingeniería capaz de satisfacer las necesidades de cada una de las empresas que operan con sistemas de aplicación, ya que un

<sup>1</sup> Dave Sharon, Presidente de Case Associates Inc., en Oregon

<sup>2</sup> Stephen Errico de Cia. Price Waterhouse define estos enfoques.

sólo producto no puede proveer todas las soluciones que el sistema requiere. Demasiadas líneas de código con diferentes estilos de programación y una gran variedad de niveles de tecnología impiden que utilizando una sola herramienta se pueda llevar a cabo la re-ingeniería de software.

Debido a que existe una gran variedad de productos, la re-ingeniería de software puede ser un proceso complicado. El consejo que se da universalmente, pero que es frecuentemente ignorado es **planear, planear, planear**. Después es esencial contar con un buen grupo de capacitación.

Actualmente existen herramientas que están en uso y tienen calidad, pueden dividirse en:

**Reestructuración de código.-** Aceptan como entrada código fuente, sin que esté estructurado, realizan un análisis de ingeniería inversa y reestructuran el código para ajustarlo a los conceptos de programas modernos y se centran en el diseño procedural de los programas.

**De revisión de datos.-** Analizan las definiciones de los datos de una base de datos, traduciendo esta descripción a una notación gráfica. Con esto la persona que utiliza la herramienta puede modificar la estructura lógica de la base de datos, normalizar los archivos y generar de manera automática el nuevo diseño físico de la base de datos.

A continuación, en la Tabla No. 2 hacemos referencia a una lista de herramientas de reingeniería producidas por importantes empresas en el mundo, así como sus principales características:

HERRAMIENTAS REPRESENTATIVAS DE RE-INGENIERIA VENDEDOR	PROYECTO	REQUERIMIENTOS DE SISTEMA OPERATIVO	COMENTARIOS
ADPAC CORP.	PM/SS	MVS	ESTANDARIZACION DE NOMBRES DE DATOS Y ANALISIS DE CODIGO PARA COBOL,PL/1 Y JCL.
ADVANCED TECHNOLOGY INTERNATIONAL.	SUPERCASE	VAX/VMS	INTEGRACION DE HERRAMIENTAS DE DESARROLLO; INGENIERIA REVERSIVA
ADVANTADGE SYSTEMS GROUP	DATABASE MIGRATION TOOLS	MVS	MIGRACION DIMS
ALDON COMPUTER	ANALYZER	HP-3000	ANALIZADOR DE CODIGO
ANDERSEN CONSULTING	FOUNDATION	MVS,PC-DOS,OS-2,VAX/VMS	INTEGRACION, CASOS DE CICLO DE VIDA Y REING
APPLICATION PROGRAMMING	LOGIC CHAIN		DOCUMENTACION Y ANALIZADOR DE METRICAS
AUTOCASE TECHNOLOGY	AUTO FLOW-C	PC-DOS Y OS 2	GENERACION DE ESTRUCTURA DE CARACTERES Y FLUJO DESDE CODIGO C.
BACHMAN INFORMATION SYSTEMS	BACHMAN RE-ENGINEERING SEET	PC-DOS	RE-INGENIERIA, DE LA DESCRIPCION DE BASE DE DATOS Y CODIGO
CADRE TECHNOLOGIES	TEAMWORK,	VAX/VMS,UNIX,PC-DOS Y OS-DOS	INTEGRACION
	PATHMAP,	VAX/VMS, UNIX, PC-DOS	INGENIERIA REVERSIVA
	TEAMWORK 'C REV	VAX/VMS Y UNIX	FORMAS DE LAS ESTRUCTURAS DE CARAC. DESDE ARCHIVOS FUENTES EN C.
CARLETON CORP.	CQS	MVS	ANALISIS DE DATOS, HERRAMIENTAS DE CONVERSION Y MIGRACION
COMPUTER ASSOCIATES INT'L.	CA - LIBRARIAN	DOS/VSE,MVS Y VM	ADMINISTRADOR DE CODIGO FUENTE
COMPUTER DATA SYSTEMS,INC.	THE RE-ENGINEERING PLATAFORM	MVS	METODOLOGIAS DE RE-INGENIERIA PARA COBOL.
COMPUTER TASK GROUP	DATA CONVERSION TOOLS	MVS, DOS/VSE Y PC-DOS	HERRAMIENTAS DE MIGRACION, ANALISIS Y SERVICIOS DE CONVERS
D APPLETON Co.	IDEF/LOVERAGE	MVS, VAX/VMS	DISEÑO DE SISTEMAS REUSABLES, BASADOS EN LA METODOLOGIA IDEF.

*Tabla No. 5 Herramientas Representativas de la Re-ingeniería de Software*

HERRAMIENTAS REPRESENTATIVAS DE REINGENIERIA VENDEDOR	PRODUCTO	REQUERIMIENTOS DE SISTEMA OPERATIVO	COMENTARIOS
GLOBAL SOFTWARE	OLGA	MVS	CAPTURAS, MANTENIMIENTO DE ICL EN DICCIONARIO DE DATOS
	URMA	MVS	CAPTURAS, MANTENIMIENTO DE LA DOCUMENTACION DE LOS PROGRAMAS EN EL DICCIONARIO DE DATOS
GROUP OPERATIONS	SCAN COBOL	MVS, DOS/VSE	DOCUMENTOS LOGICOS E IDENTIFICACION DE PROBLEMAS POTENCIALES
	SUPERSTRUCTURE	MVS, DOS/VSE Y PC-DOS	COBOL ESTRUCTURADO
HEWLETT-PACKARD	HP C**/SOFTBENCH	HP/UX	DESARROLLOS ORIENTADOS A OBJETOS, ANALIZADOR PARA ANALISIS
HYPERSOFT CORP.	APPLICATION	VAX/VMS	INGENIERIA REVERSIVA PARA APLICACIONES EN COBOL
INFOSPAN CORP.	INFOSPAN REPOSITORY MANAGEMENT SYSTEM (IRMS)	PC-DOS, OS/2 Y UNIX	SOPORTA CASE, RE-INGENIERIA Y HERRAMIENTAS PARA DICCIONARIO DE DATOS
EDP MANAGEMENT	REFORMAT		HERRAMIENTAS DE REFORMATEO
FWM DIOITECH	COHXREF	WANG VS	DOCUMENTACION Y ANALIZADOR DE METRICAS
UNISYS	LINC	UNISYS	4GL PARA CASOS DE RE-INGENIERIA
UNISYS	LIDA	DOS 3.0 <OS2	DISEÑO ASISTIDO PARA LINC

Tablas No. 5a

*Herramientas Representativas de la Re-ingeniería de Software*

En el caso práctico se utilizará como lenguaje de programación a LINC y a LDA como la herramienta CASE.

LDA es la herramienta que se utilizó en el caso práctico para llegar hasta la etapa de diseño.

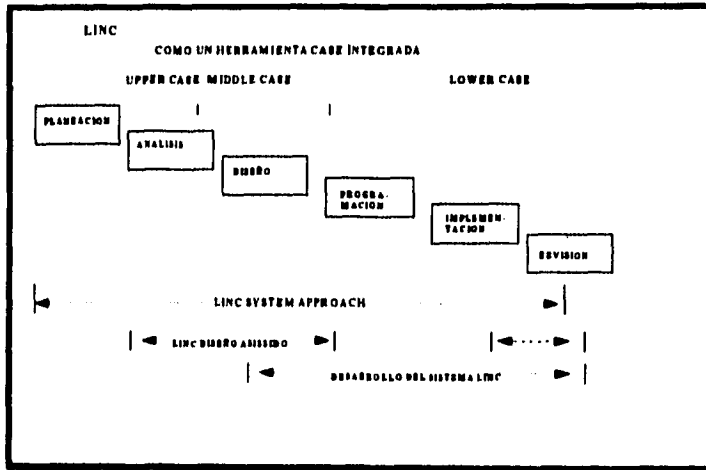
**EL ROL DE LINC:**

Es un ambiente total para desarrollo y operación de los Sistemas de Información que modelan sistemas de negocios.

LINC es un compilador interactivo para línea y red. Es un lenguaje de 4a. generación y genera Bases de Datos.

La relación que existe entre LDA y LINC en el desarrollo de sistemas durante la re-ingeniería de sistemas, se muestra en las siguientes figuras.



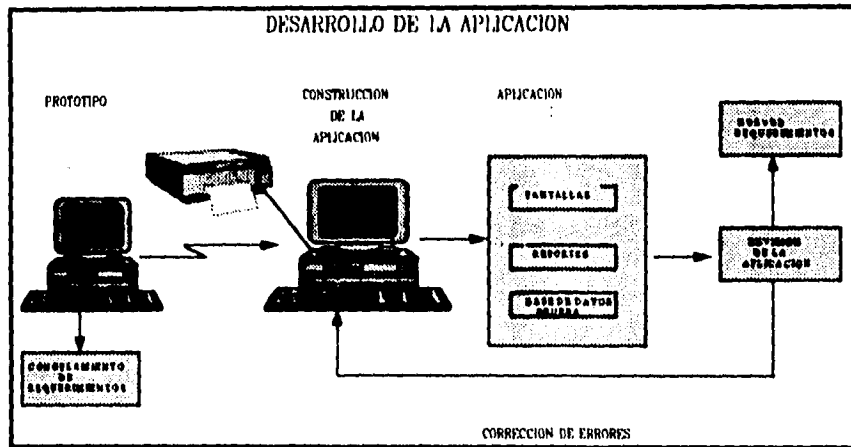


*Fig. No. 10*  
*LDA como una herramienta CASE integrada*

En esta figura existe una descripción general de la parte que abarca LDA y LINC en el desarrollo de un sistema

Podemos observar que LDA está presente en la parte de Análisis y Diseño y también lo podemos utilizar en la Revisión.

A partir de la programación veremos a LINC.



*Fig. No. 11*  
*Desarrollo de una aplicación*

Esta lámina muestra todo lo que implica una aplicación, de una forma general.

Todos los requerimientos que existen deben estar en LDA, después se van a transferir a LINC. En LINC podemos ver las pantallas, los reportes y la demostración de la Base de Datos.

En esta figura podemos ver que existe un ciclo el cual termina hasta que todos los requerimientos estén dentro.

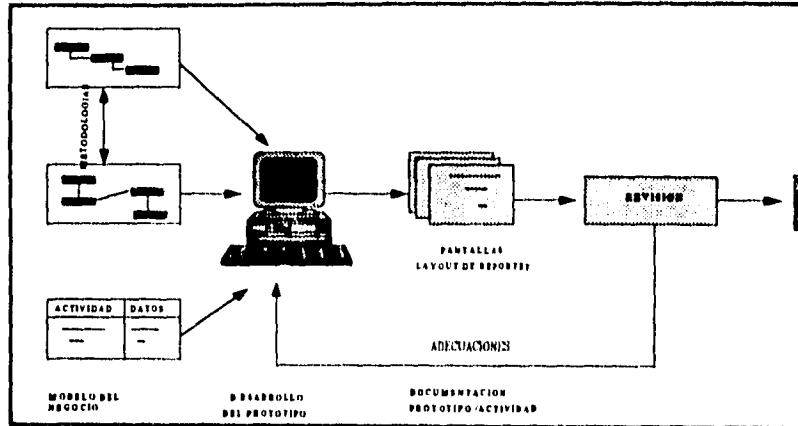


Fig. No. 12  
Proceso de LDA

Esta figura muestra todo el ciclo de LDA, desde la selección de la metodología, el análisis, el diseño y la revisión del diseño, para comenzar el desarrollo.

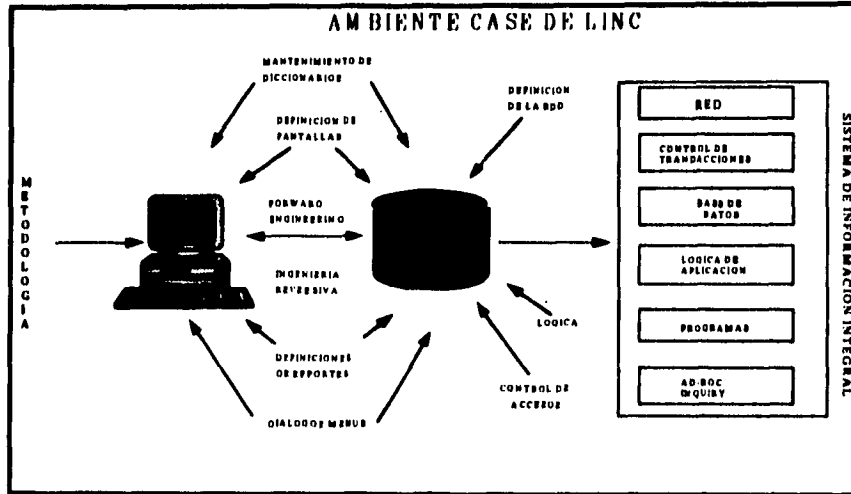


Fig. No. 13  
Ambiente CASE de LINC

Esta figura muestra la relación directa entre LDA y LINC El presente capítulo introduce al lector a conocer las diferentes definiciones de la re-ingeniería de software y la tecnología de software existente.

### **ESTRATEGIA Y ECONOMIA DE LA RE-INGENIERIA DE SOFTWARE**

Uno de los principales problemas existentes en la historia de la informática se refiere a la reducción de costos de procesamiento y almacenamiento de los datos en el área de hardware de computadoras, este problema se ha eliminado gracias a los avances tecnológicos obtenidos hasta la década de los 80's.

Actualmente el principal problema, existe en el área del software, ya que existe el desafío de mejorar la calidad y reducir los costos de todas las soluciones basadas en computadoras implementadas con el software.

El software es el elemento que marca la diferencia en cuanto a la suficiencia y oportunidad de la información en cualquier organización. Durante mucho años se le dio más importancia al desarrollo del hardware en cuanto a la reducción de costos de procesamiento y almacenamiento de los datos. En estos primeros años el hardware evolucionó continuamente y era de propósito general, mientras que el software se veía simplemente como un añadido y su desarrollo se realizaba sin ninguna planificación y se diseñaba a medida.

A continuación presentamos algunas estrategias de Reingeniería.<sup>3</sup>

#### **Economía de la Re-ingeniería de Software**

Es una necesidad importante saber calcular y justificar los costos de la re-ingeniería de software.

El objetivo de la re-ingeniería es reducir los costos de mantenimiento. Este objetivo es para incrementar la calidad y reducir la complejidad. Los costos de re-ingeniería son manejados por el tamaño y la interdependencia del software, esto es bueno por la utilidad de automatización.

Existen factores los cuales manejan los costos de mantenimiento como la calidad del producto, los cuales son:

- Calidad del ambiente
- Calidad del personal
- Calidad organizacional

**Calidad del ambiente:** Los desarrolladores y los que mantienen el software dependen del ambiente en el que trabajan. Su productividad esta determinada por las herramientas que tienen para trabajar; no es lo mismo si se tienen herramientas antiguas que se tenga lo más moderno en tecnología.

---

<sup>3</sup> CASE Trends magazine from Nov/Dec 1990 May/June 1991.

La calidad del personal es el factor que más influye en el manejo de los costos de desarrollo ya que depende del dominio de los casos que se tengan para solucionar cualquier problema y para ir avanzando más en el desarrollo.

Finalmente el mantenimiento de la organización. Una buena organización con una alta estructura es mucho más efectiva que cualquier tienda. Debe haber un control en los procesos para tener un buen mantenimiento de software.

Todo esto es para justificar la re-ingeniería de software basándose en la reducción de los costos del mantenimiento e incrementando beneficios en ésta.

La decisión de hacer re-ingeniería a un sistema es una difícil decisión y va a estar basada en la economía de la empresa. Idealmente antes de hacer re-ingeniería, se debería hacer conciencia del problema del mantenimiento, cuánto costaría hacerla, cuánto puede ser salvado en mantenimiento y cuándo veríamos la inversión.

Existen 11 criterios que pueden ayudar a saber decidir cuándo se debe comenzar a revisar un sistema existente, los cuales son:

1. Cuál es la frecuencia en que ocurren fallas
2. Cuando el código tiene más de 7 años
3. Cuando la lógica del programa y el flujo de la misma llega a ser muy compleja
4. Cuando el programa fue escrito en hardware de generaciones anteriores
5. Cuando el programa corre a través de un emulador
6. Cuando los módulos o subrutinas llegan a ser muy grandes
7. Cuando se requieren recursos excesivos para correr un sistema
8. Cuando los parámetros están sujetos a cambios
9. Cuando es muy difícil mantener los programas
10. Cuando la documentación está obsoleta
11. Cuando las especificaciones del diseño son extrañas, incompletas u obsoletas.

Es importante considerar estos puntos cuando se quiere hacer re-ingeniería.

Un análisis completo de costo-beneficio es inevitable. Una de las razones, de mayor frecuencia, para la re-ingeniería es para reducir los costos del mantenimiento, pero existen otras buenas razones, una es para bajar los niveles de error, otra es para convertir el software a una mejor plataforma y otra es para prolongar la vida de los sistemas.

Frecuentemente, la re-ingeniería, es considerada una alternativa viable, cuando el esfuerzo de la re-ingeniería no sobrepasa el 50% del esfuerzo.

Si el 80% del problema es causado por el 20% del software, entonces es lógico que sólo se haga re-ingeniería al 20% y dejar el resto como está.

### **Estimación de los costos de la Re-ingeniería de software**

El costo de la re-ingeniería se estima de acuerdo al tamaño y la complejidad del sistema al cual se va hacer la re-ingeniería.

El sistema analizado es descompuesto en partes. Para un negocio típico la aplicación debe ser:

- Programas
- Subrutinas
- Procedimientos
- Utilerías
- Reportes/Guías
- Archivos
- Estructura de datos
- Parámetros

El tamaño del software puede ser expresado por los números de procesos de los programas, subrutinas, jobs, etc. o por número de líneas del código. El tamaño de los datos es determinado por el número de datos, reportes, archivos, registros, parámetros o por el número de campos de los datos.

Es necesario conocer la cantidad de esfuerzo estimado para el desarrollo y así calcular la complejidad del sistema.

El nivel de tamaño del sistema es determinado por el número de relaciones entre los componentes de las partes, las llamadas a subrutinas, el acceso de archivos y las entradas/salidas (I/O) de operaciones entre los componentes de las partes.

Para estimar el esfuerzo de re-ingeniería, el número medido de unidades que se va a utilizar para hacer la re-ingeniería, necesita ser multiplicado por el esfuerzo requerido por cada tipo de unidad. Este esfuerzo por tipo de unidad, es la productividad de la re-ingeniería, la cual es derivada de un análisis empírico de la experiencia pasada.

### **Avalúo de los beneficios de la Re-ingeniería de Software**

Un avalúo de los beneficios de la re-ingeniería de software debe primero definir el contexto del problema. Esto es típicamente tres tipos de situaciones:

**A.** Cuando la aplicación del sistema existente ha llegado a ser técnicamente obsoleto y debe ser reemplazado.

**B.** Cuando hay problemas técnicos severos con el sistema existente.

**A.** Este caso sólo existe cuando una de la elecciones esta entre redesarrollo y re-ingeniería. Si las funciones del sistema son estables y sólo la implementación técnica es obsoleta. Es definitivo que es necesaria la re-ingeniería.

**B.** En este caso existe un alto costo con el mantenimiento. Por lo consiguiente hay cuatro posibilidades:

- Reemplazar el sistema con un paquete estándar
- Redesarrollar el sistema
- Hacerle re-ingeniería al sistema
- Continuar pagando el alto costo del mantenimiento

**C.** La calidad de la aplicación del sistema es adecuada pero el mantenimiento es muy costoso, así como el bajo ánimo de los programadores. El usuario tiene mucho personal y que en ocasiones tarda mucho tiempo en completar una tarea. La situación no es crítica pero sí una molestia.

**Los beneficios que se tendrían son los siguientes:**

- Bajos costos de mantenimiento en salvar tiempo y esfuerzo por mantener una tarea
- Bajos costos de mantenimiento al reemplazar personal senior por personal junior
- Bajos costos a través de una reducción de tiempo perdido.
- Bajos costos con el nuevo personal.

**Ya que los programas han sido seleccionados para la Re-ingeniería:**

- Cuando un programa ha sido convertido a otro ambiente
- Cuando un programa ha sido funcionalmente alterado o mejorado
- Cuando el programador responsable lo ha abandonado
- Cuando el sistema esta fuera de control

**Costo-Beneficio de la Re-ingeniería de Software:**

En el análisis de costo/beneficio, se debe considerar los costos, beneficios y riesgos del desarrollo de un nuevo sistema así como los costos y riesgos del mantenimiento del viejo sistema.

**Existen factores importantes que se deben considerar para hacer :**

- Re-ingeniería
- Re-desarrollo o
- Mantenimiento

Son:



Costos de Re-ingeniería relativos a los costos de re-desarrollo y los costos del mantenimiento

Adicionar valores de re-ingeniería relativa al valor del nuevo sistema y al valor del sistema que está presente.

Riesgos de Re-ingeniería de Software relativos a los riesgos del nuevo desarrollo y riesgos de lo no hecho.

Expectativa de vida en tiempo del sistema existente relativos al tiempo requerido para la re-ingeniería y el tiempo requerido para el re-desarrollo.

Finalmente la decisión que se tome, debe tener una calidad y funcionalidad significativa.

### **EXPERIENCIA Y EVALUACION DE LA RE-INGENIERIA DE SOFTWARE**

Determinar si en un tiempo específico, el sistema puede tener un costo de re-ingeniería de software efectivo sobre una plataforma comercial, incluyendo hardware, herramientas comerciales, lenguajes y sistemas operativos de software comerciales, mientras es definido el funcionamiento de la aplicación, como un bi-producto de la re-ingeniería, la arquitectura de las aplicaciones y de otras bases de datos pueden ser simplificadas al eliminar dependencias innecesarias, creando mientras una arquitectura visible. Esto es en armonía con los objetivos del negocio al eliminar los componentes utilizados del sistema y a desarrollar y utilizar los programas de aplicación que son fáciles de leer y modificar. Los componentes utilizados típicamente requieren adiestramiento y mantenimiento especializado y son vulnerables a cambios en una línea de producto o tecnología.

Existen características de los sistemas candidatos para el re-diseño, las cuales son:

- Código fuente con 7 años de antigüedad
- Estructura excesivamente compleja
- Código escrito para una generación de hardware anterior
- Uso de demasiados parámetros
- Módulos muy largos

En este apartado vamos a mencionar los pasos a seguir para conocer bien el procedimiento que se tiene que llevar a cabo para el sistema que se va a presentar en el caso práctico, estos pasos son para hacer re-ingeniería; para reducir el mantenimiento del sistema, como a continuación se describe:

- **HISTORIA DEL SISTEMA**

- Explicación del sistema
- Número de programas
- Lenguaje en que está el sistema
- Tiempo que tiene en uso
- Líneas de código
- Cambios realizados hasta el momento
- Documentación existente
- Subsistemas u otros sistemas utilizados
- Estandarización de los datos

- **HISTORIA DEL MANTENIMIENTO DE LOS DATOS**

- Tipo de mantenimiento que se le ha dado
- Mantenimiento a reportes
- Tiempo
- Cambios por año
- Análisis del mantenimiento a los datos
  - Tipo de problemas de datos
- Estrategia de revitalización del sistema
  - Soluciones para mejorar el sistema
  - Objetivos de la re-ingeniería
  - Desarrollar un mini análisis para conocer la factibilidad de la re-ingeniería:
    - Definición formal de los problemas y soluciones
    - Definición de los parámetros del sistema incluidos en ambos problemas y soluciones)
    - Determinar el número de componentes afectados del sistema
    - Documentación de los resultados del mini análisis en los trabajos de mantenimiento.
  - Aplicando el método revolucionario: Metas y objetivos del proyecto que se van cumpliendo con la re-ingeniería.
  - Implementando las soluciones en las áreas de procesamiento de transacciones, administración de datos y modelación de datos.
  - El papel de las herramientas CASE durante el redesarrollo.
  - Resultados que utilizan las herramientas CASE.
  - Conclusiones
    - Decremento en las solicitudes de mantenimiento
    - Reducción de tiempo dedicado al mantenimiento
    - Reducción en el número de reportes de fallas
    - Almacenamiento en línea es reducido significativamente
    - Almacenamiento en cinta menor
    - Procesamiento batch (beneficios)

El costo-beneficio de un proyecto de re-ingeniería depende de los objetivos de ésta.

El contexto para la re-ingeniería puede establecerse en términos de los objetivos del negocio.

La re-ingeniería de software es un proceso complejo y difícil, los sucesos de las organizaciones al utilizar esta tecnología puede ser determinada por el nivel de compromisos hechos por la organización.

**Para comenzar una re-ingeniería de software debemos conocer:**

- **Objetivo de la re-ingeniería** : Lo que se desea con el nuevo sistema
- **Elección y ventajas técnicas**: Selección de herramientas, el porqué y sus características.
- **Metodología para re-ingeniería**:
  - Basada en el sistema original
  - Extracción/Análisis de datos, código fuente y documentación
  - Generación de documentación
  - Generación de nuevo código
  - Pruebas y ejecución del código
- **Categorización de los programas de acuerdo al nivel de complejidad y la interfase entre programas.**
- **Dificultades encontradas durante el proceso de re-ingeniería**
- **Impacto de emigrar o rediseñar los datos**

Una forma de evaluar nuestra re-ingeniería de software es el análisis de puntos por función.

#### **ANÁLISIS DE PUNTOS POR FUNCION**

Para medir la productividad de los procesos de la re-ingeniería de software se utiliza el análisis de punto de función, que en muchos de los casos provee las métricas convencionales para medir la productividad del software, éste se basa principalmente en el código fuente (líneas de código contadas).

El análisis de punto por función tiene sus bases además en la medición de entradas, salidas, archivos maestros e interfases. Los posibles impactos influenciales son analizados para determinar el nivel de complejidad del sistema. Este análisis provee de un número de dimensiones que es un indicador de funcionalidad<sup>4</sup>.

---

<sup>4</sup>. El Grupo Internacional de Usuarios del Punto de Función (IFPUG) publica toda aquella información relacionada con definiciones y metodologías, en el boletín IFPUG90.

**Resultado de las Métricas de Análisis.**

El resultado del análisis enfocado a los procesos de re-ingeniería de software afectan el mantenimiento y la flexibilidad del código.

Los aspectos referentes a código fuente, que pueden ser contabilizados después de la re-ingeniería de software, se muestran a continuación.

**• METRICAS**

**Total de líneas de código ejecutable**  
**Número de programas del sistema**  
**Total de decisiones contadas**  
**Densidad de decisiones**  
**Total de funciones contadas antes y después de la re-ingeniería de software**  
**Total de entradas y salidas contadas**  
**Tamaño del sistema**  
**Funciones y programas**  
**Densidad de funciones**  
**Número total de archivos del sistema**  
**Número total de llamadas**  
**Densidad de comentarios**

Todas las métricas anteriores deben ser contabilizadas antes y después de realizar una re-ingeniería de software.

**CAPITULO IV**

**APLICACION DE LA RE-INGENIERIA DE SOFTWARE**

El proceso de re-ingeniería de software será aplicado al Sistema S136 "Responsabilidades de Cartera", el cual es propiedad de Banco Nacional de México S.A. (BANAMEX).

El análisis que se efectuó para llevar a cabo la re-ingeniería de software del S136, se basó en una metodología propia de BANAMEX que tiene por nombre "Nuevo Enfoque".

A continuación presentaremos una descripción general del sistema siguiendo los pasos expuestos en "Experiencia y evaluación de la Re-ingeniería" del Capítulo III

**RESPONSABILIDADES DE CARTERA****HISTORIA DEL SISTEMA****• ANTECEDENTES**

En Enero de 1981 el Banco de México S.A. en un comunicado oficial indicaba los cambios y adiciones necesarias para reportar la relación de Responsabilidades de Cartera, esto es que cada Banco tenía que presentarle una relación de sus clientes que pagan puntualmente (cartera vigente) y de sus clientes con problemas de pago y cuya deuda con el Banco es mayor a los doscientos mil nuevos pesos (cartera vencida).

Para cumplir con este requerimiento oficial en BANAMEX se establecieron las necesidades de información para reportar la Relación de Responsabilidades bajo un nuevo esquema; con el fin de garantizar la información en forma oportuna y confiable al Banco de México, S.A. y a los usuarios del Banco.

**• DESCRIPCION DEL SISTEMA**

El sistema de Responsabilidades de Cartera es un sistema automatizado que permite el registro de operaciones, mantiene actualizado el archivo de datos y elabora las salidas que pide Banco de México; así mismo, considera las fuentes para operarlo y los requerimientos de información de Banca Regional y Metropolitana.

Su diseño contempla procesos operativos y de información en el equipo Serie A de Unisys en su modelo A17.

**Tipo de Sistema :** Se trata de un sistema de captura vía terminal, de actualización mensual de saldos tipo batch, emisión de cinta para el Banco de México, Comisión Nacional Bancaria e información a sucursales y departamentos de dirección de BANAMEX.

Tiene como **Objetivos específicos :** Simplificar el trabajo en sucursal , producir la información de Relación de Responsabilidades con mayor exactitud, producir la cinta con información de clientes, emitir el reporte de la Relación de Responsabilidades el cual es elaborado en microficha, estandarización de estos reportes y lograr mayor calidad en la información producida.

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

**Procesos básicos del Sistema :****A) Proceso de Captación y Actualización**

Aquí se lleva a cabo precisamente la captación y actualización de los movimientos recibidos, la emisión de cifras de control de sucursales actualizadas y no actualizadas y la emisión de rechazos por movimientos y altas de clientes.

**B) Emisión de Reportes**

En este proceso se emite el reporte de saldos por cliente y una relación de clientes de cartera para captura de saldos del próximo mes. El sistema actualmente genera un total de 55 reportes, equivalentes a 13 000 páginas.

**C) Emisión de Información**

En esta parte se lleva a cabo la emisión de la cinta, y de la Relación de Responsabilidades (1500 Microfichas) ambas para el Banco de México, así mismo se emiten las carátulas gerenciales para normas y técnicas de crédito. Se realiza la integración de la información de SENICREB (Servicio Nacional de Información de Crédito Bancario) con información de usuarios mayores a doscientos mil nuevos pesos.

**Políticas Generales :**

- Los archivos de este sistema son controlados en forma automática por el sistema mediante fechas de proceso.
- Las cifras de control de cada una de las sucursales son reportadas, cada vez que existe un cambio para la sucursal.
- La información es proporcionada a las sucursales mediante listados que previamente les envía el departamento de Cartera Metropolitana
- La captura de información se lleva a cabo vía terminal.
- El sistema maneja moneda nacional y moneda extranjera.
- La información sólo se conservará durante seis meses.
- Los clientes que no tienen movimiento durante los últimos seis meses automáticamente son dados de baja.

- El sistema contempla las siguientes validaciones : "por movimientos", errores, en suma horizontal, error en número o clave de usuario, no existe cliente en archivo maestro, error en importe, error en clave de sector, error en clave de cuenta, error en clave de fondo, error tipo moneda.

"Por cliente": error en número o clave de usuario, nombre del cliente sin comas, nombre del cliente sin diagonal, error en clave de localidad, error en clave de actividad, error en clave de banca, error en clave de grupo económico, error en clave del avalista, ya existe cliente en el archivo maestro.

• **NUMERO DE PROGRAMAS.**

El Sistema de Responsabilidades de Cartera cuenta con un WFL (Apéndice I) y veintitrés programas divididos de acuerdo a la periodicidad de procesos, de la siguiente forma :

**Programas para proceso diario,**

**Programas para proceso mensual,**

**Programas para proceso eventual,**

**Programas para proceso de precierre mensual,**

**Programas para proceso de cierre mensual.**



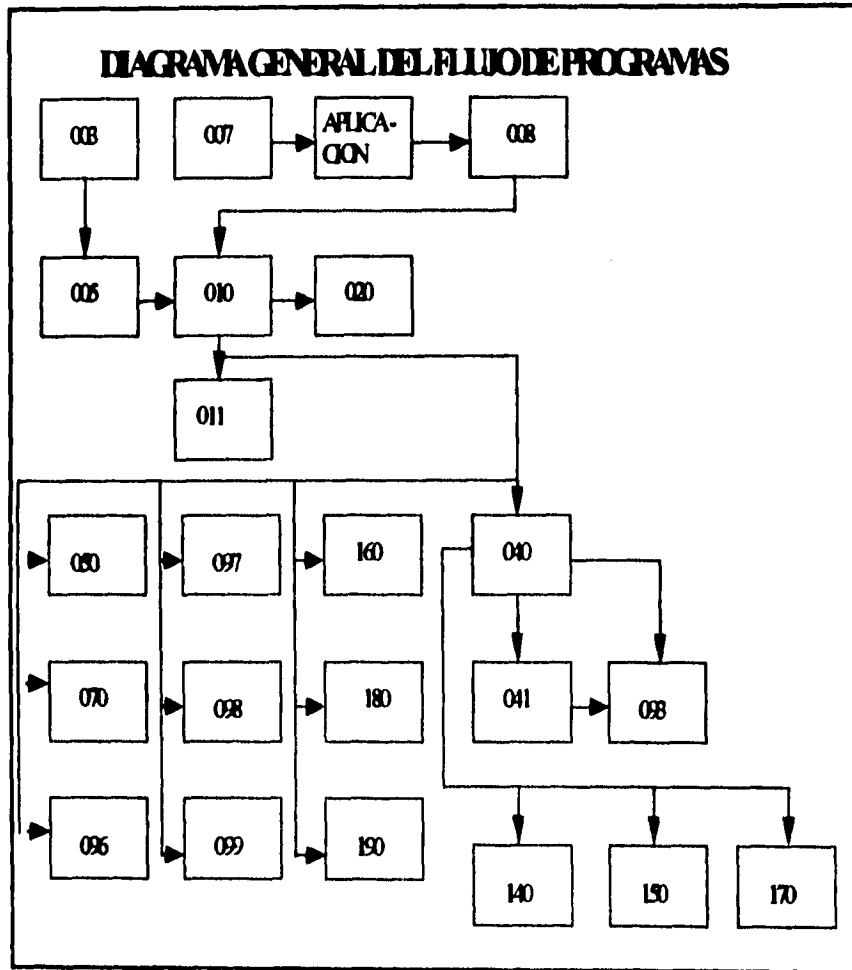
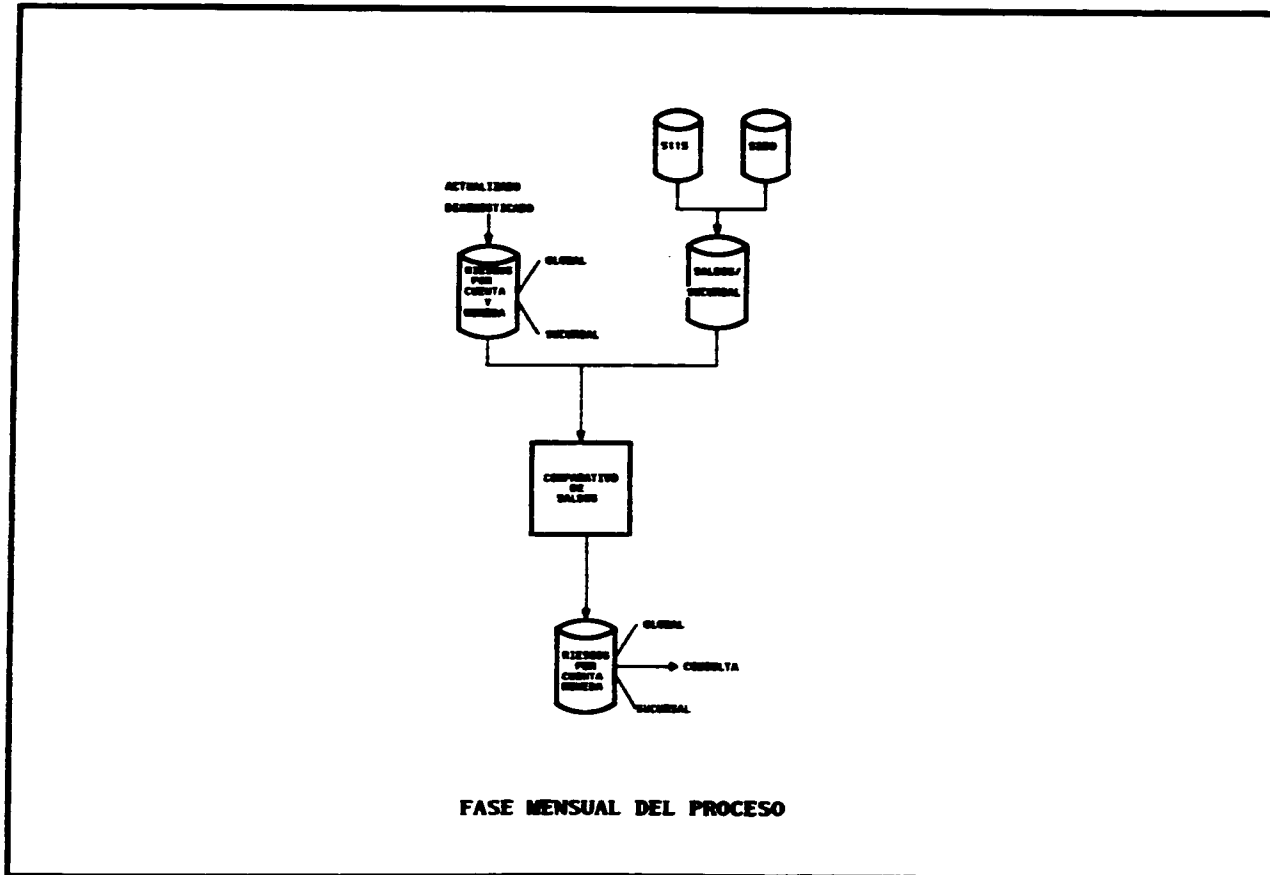


Fig. No. 14  
Diagrama General de Flujo de Programas

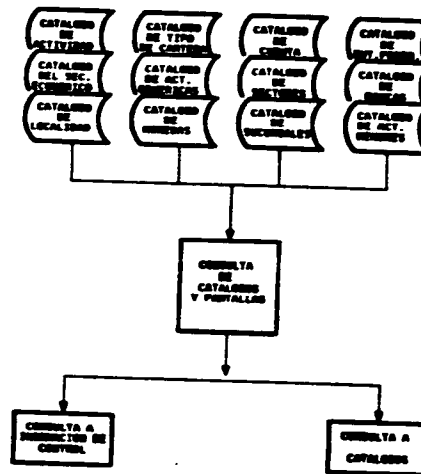
**TESIS SIN PAGINACION**

**COMPLETA LA INFORMACION**

**RESPONSABILIDADES DE CARTERA**

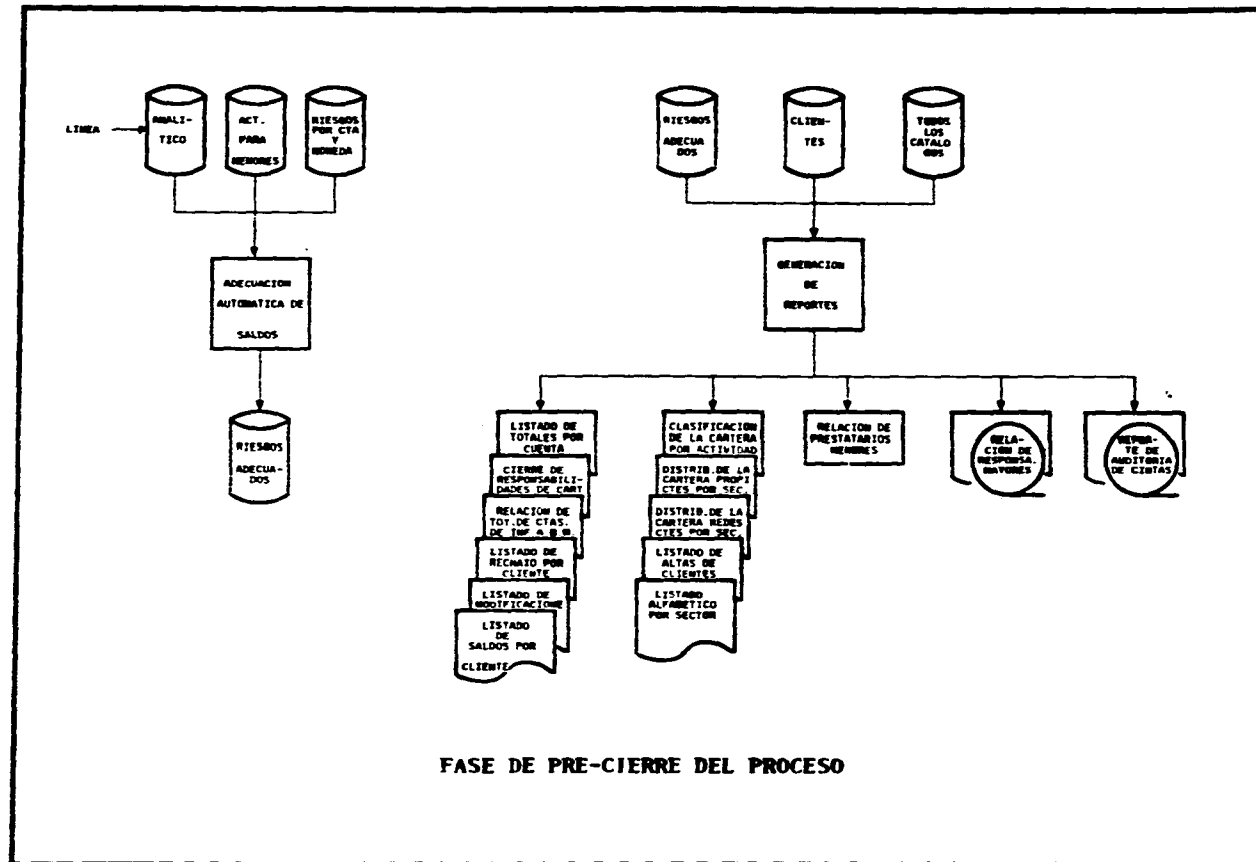


## RESPONSABILIDADES DE CARTERA

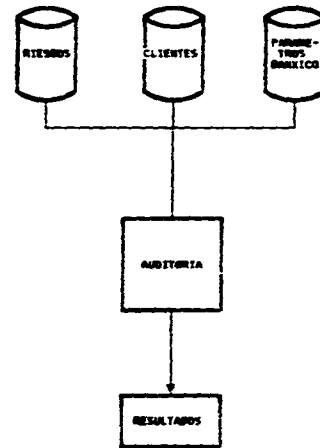


FASE EVENTUAL DEL PROCESO

## RESPONSABILIDADES DE CARTERA



**RESPONSABILIDADES DE CARTERA**



**FASE DE CIERRE DEL PROCESO**

Cada uno de los programas que integran el Sistema de Responsabilidades de Cartera se denomina **paso (P)** y se califica con un número (003, 007, 010, etc.). Cada paso tiene una o varias funciones dentro del sistema, las cuales expondremos a continuación :

**WFL** : Tiene como objetivo controlar el proceso del sistema S136 automáticamente.

**P003** : En base a la información enviada por Internet a las agencias en el extranjero extrae la información para completarla y complementarla para su ingreso al Sistema de Responsabilidades de Cartera.

**P005** : Revisa la información de archivos y consolida la información para ser procesada por el S136/P010.

**P007** : Genera un archivo para el proceso de línea.

**P008** : Genera el archivo de movimientos de la línea para su uso en el S136/P010 mensual.

**P010** : Con los movimientos, altas y modificaciones generadas por las sucursales actualiza el archivo maestro de la Relación de Responsabilidades, validando y clasificando la información; rechazando y generando un archivo de los movimientos con error .

**P011** : Extrae información del archivo de movimientos con error, acumulándola a registros globales por sucursal con lo que crea un archivo para ser reingresado al sistema.

**P020** : A partir del archivo maestro, y de acuerdo con la opción de proceso del P010, genera los reportes de clientes BANAMEX actualizados, y genera un comparativo de saldos de posición contra saldos captados.

**P040** : Basado en la información del archivo maestro de la Relación de Responsabilidades, genera las cintas con la información de clientes de cartera, efectúa también una depuración del archivo maestro, eliminando los clientes más antiguos que no hayan tenido riesgos.

**P041** : A partir del archivo maestro genera información de clientes calificándolos con un número llamado "Número Banxico".

**P050** : A partir del archivo maestro genera carátulas gerenciales, información de sectores y cinta de clientes mayores para el departamento de normas y técnicas de crédito.

**P070** : Genera un listado comparativo entre los saldos del Sistema de Responsabilidades (S136) y los del sistema de Contabilidad de Sistemas Automatizados (S250) para el ajuste de cifras para el Banco de México.

**P093** : Basado en la información de las cintas generadas por el S136/P040, desglosa los importes de Moneda Nacional y moneda extranjera de cada cuenta, así como, número de registros e importe total de cada cinta.

**P096** : Genera el listado de clientes mayores, personas físicas sin número Banxico para la asignación de dicho número.

**P097** : Es un proceso eventual que se lleva a cabo a solicitud del usuario para editar reportes con movimientos de las sucursales que aún no están automatizadas.

**P098** : Lleva a cabo la distribución de la cartera por entidad federativa y actividad económica.

**P099** : Edita los saldos de cartera redescontada por actividades económicas y localidades.

**P140** : Edita el detalle de las operaciones redescontadas por fondo de fomento.

**P150** : Genera un listado del archivo maestro de la Relación de Responsabilidades de Cartera.

**P160** : Genera los dígitos para las sucursales manuales y para las sucursales nuevas.

**P170** : Genera un listado del archivo maestro de la Relación de Responsabilidades, en el que clasifica las localidades y actividades con o sin error.

**P180** : Genera un archivo con información de clientes de cartera vencida con diez mil nuevos pesos ó más. Así como, un archivo de saldos por Sucursal-Actividad a partir del archivo maestro.

**P190** : Proporciona información de cartera vencida para el comité de crédito.



- **TIEMPO DE PROCESO :**

Para un proceso normal el tiempo de procesamiento es de 1 hora con 30 minutos aproximadamente si se lleva a cabo después de las 8 de la noche; y de 2 horas con 30 minutos si se realiza durante el día.

Para el proceso de cierre se requieren de aproximadamente 12 horas durante el día y de 5 horas con 30 minutos durante la noche.

- **LENGUAJE**

El sistema se encuentra programado en COBOL ANSI 74.

- **TIEMPO DE USO**

El sistema entró en vigencia en el año de 1981 siendo su tiempo de uso a la fecha 14 años.

- **LINEAS DE CODIGO**

El total aproximado de líneas de código COBOL por todo el sistema es de un millón de líneas.

- **CAMBIOS REALIZADOS**

Los cambios que se efectúan al sistema generalmente son por falla de alguno de los pasos, por algún requerimiento presentado por el usuario del sistema y por alguna optimización en los datos, reportes y tiempo de proceso.

- **DOCUMENTACION**

El sistema cuenta con un manual de operación el cual se encuentra dividido en :

Datos generales del sistema.

Descripción y diagramación de cada uno de los programas que integran al sistema.

Listados de compilación de cada programa.

WFL. (Apéndice 1).

Relación general y muestra de los listados que genera el sistema. (Apéndice 2).

Relación general y especificaciones de archivos de entrada y salida del sistema.

Catálogo de mensajes emitidos por el sistema.

- **OTROS SISTEMAS**

El sistema de Responsabilidades de Cartera interactua con los sistemas :

S080	_____	Estructura Jerárquica Institucional.
S115	_____	Sistema Central Contable.
S190	_____	Atlas (Internet)
S250	_____	Contabilidad de Sistemas Automatizados.
S400	_____	Cartera.
S404	_____	Sistema Integral de Crédito.
S680	_____	Créditos Comerciales (AMS).
PC	_____	Sistemas en PC (Agencias y sistemas extranjeros)

- **ESTANDARIZACION DE LOS DATOS**

En cuanto a las políticas de estandarización de la Institución tenemos:

Etiquetas de listados, nombre y número de programas, nombres de archivos y nombre y número de mensajes que puede emitir el programa.

**HISTORIA DEL MANTENIMIENTO DEL SISTEMA**

- **TIPO DE MANTENIMIENTO**

El mantenimiento que se ha dado a este sistema ha sido básicamente correctivo y en base a requerimientos del usuario del sistema.

En muy pocas ocasiones el mantenimiento del sistema ha sido de tipo preventivo.

- **TIEMPO**

El tiempo promedio que se invierte en el mantenimiento del sistema en horas/hombre es de aproximadamente 120 horas al mes.

- **CAMBIOS POR AÑO :**

Los cambios que se realizan al sistema durante el año son aproximadamente 145, esto en cuanto a programas modificados, ya que a veces un programa puede sufrir desde la modificación o inserción de una línea de código, una variable, hasta la eliminación o inserción de rutinas enteras. Esto dependiendo de la falla o fallas, requerimientos u optimizaciones que se presenten al departamento.

- **COSTOS OPERATIVOS :**

El costo operativo aproximado del sistema actual de Responsabilidades de Cartera, tomando en cuenta los factores presentados en las siguientes tablas es de N\$6,000,000.00

<b>NOMBRE DEL RECURSO</b>	<b>UNIDADES CONSUMIDAS</b>
Papel	13 000
Microfichas	1 500
Cintas	3
Horas/hombre	120
Tiempo de proceso	36

*Tabla 6.*

*Tabla del consumo de recursos del SI36 actual*

OPERACION	ACEPTADOS	RECHAZADOS	TOTAL
Movimientos	318 596	777	319 373
Altas	3 766	807	4 573
Modificaciones	0	43	43

*Tabla 7**Tabla de Volumen de operaciones del S136 actual*

Después de conocer la estructura y funcionamiento del sistema actual llegamos a la conclusión de que es un sistema candidato para llevar a cabo el proceso de re-ingeniería de software, ya que el código fuente tiene más de 7 años de antigüedad y está escrito para un hardware obsoleto; su estructura es excesivamente compleja, utiliza demasiados parámetros y los módulos son excesivamente largos por lo que el mantenimiento se hace más complejo.

**APLICACION DE LA RE-INGENIERIA DE SOFTWARE AL SISTEMA  
RESPONSABILIDADES DE CARTERA (S136)**

**OBJETIVO :**

El nuevo sistema tiene como objetivo permitir el registro y control de la información de los riesgos de cartera, para proporcionar al Banco de México la Relación de responsabilidades de Cartera, así como la generación de información necesaria y oportuna para el Balance Oficial y la toma de decisiones.

**COBERTURA :**

Este nuevo sistema se lleva a cabo en el Corporativo de Sistemas de BANAMEX, cubriendo la información a nivel institucional de los sistemas operacionales (Direcciones Regionales, Entidades manuales, etc. estos sistemas son S400, S404, S680, S080 entre otros).

**ALCANCES :**

- Consulta y actualización en tiempo real.
- Utilización de herramientas CASE.
- Enlaces con los sistemas de cartera y contabilidad.
- Enlaces con el Centro de Información Gerencial.
- Manejo de información del SENICREB.

**PRODUCTOS :**

- Relación de responsabilidades para el Banco de México.
- Sectores para el Balance Oficial.
- Información para la Comisión Nacional Bancaria.
- Información gerencial por fondos, cuentas, rangos.
- SENICREB.

**DESCRIPCION DEL SISTEMA :**

El diseño del sistema se llevó a cabo en LDA dando como resultado los diagramas y diccionario de datos.

El sistema está organizado por procesos los cuales interactúan entre sí para obtener los productos arriba mencionados. Estos procesos son :

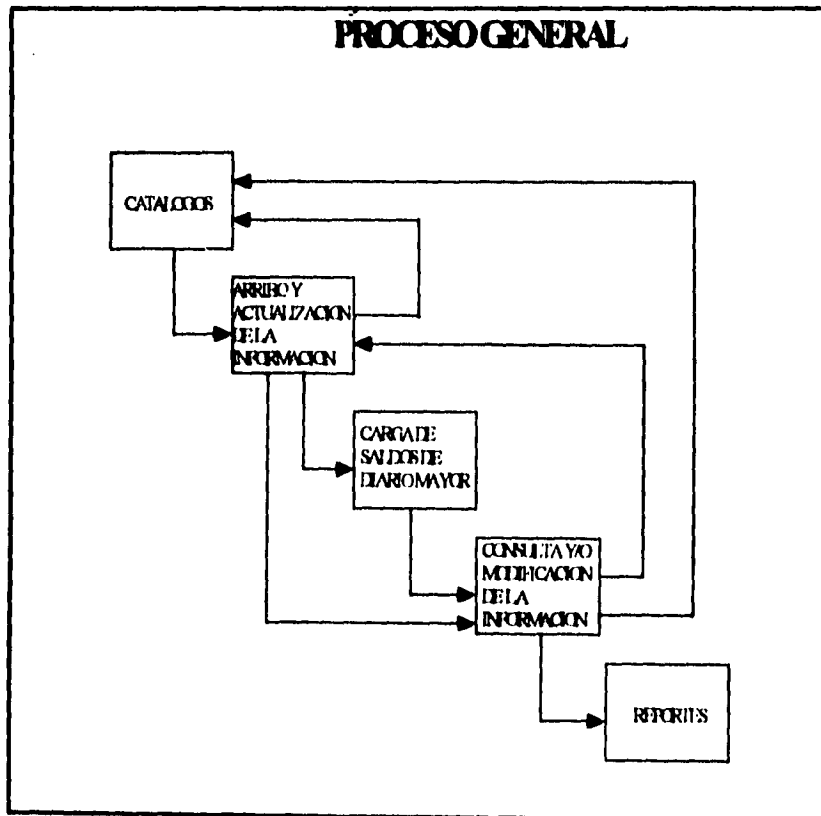
**Catálogos :** Donde se lleva a cabo una carga de información, consultas para verificar la información (vía terminal) y actualización la cual se ejecuta cuando es necesario cambiar, eliminar o integrar nueva información (vía terminal).

**Arribo y actualización de la información :** aquí se llevan a cabo los procesos de normalización y validación de datos y actualización de la información (eliminación de duplicidades).

**Carga de saldos de diario mayor :** Aquí se lleva a cabo la integración de la información de los sistemas S115 y S250, así como la carga de saldos por sucursal, por tipo de moneda y por cuenta, y se emiten los saldos por sucursal y globales así como un comparativo de diferencias contra riesgos.

**Consultas y/o modificación de la información :** Aquí el usuario puede consultar y modificar por línea y bajo cierto esquema de seguridad los catálogos de riesgos, clientes, control de arribos, totales por sucursal y totales globales; siendo los principales procesos altas, bajas, modificaciones y consultas. Así mismo, puede verificar los archivos arribados y faltantes así como, los problemas encontrados en los archivos.

**Reportes :** Aquí se lleva a cabo la emisión de listados, microfichas y archivos.



*Fig. No. 15*  
*Proceso General del Nuevo Sistema*

Para llevar a cabo el proceso de re-ingeniería al sistema S136 Responsabilidades de Cartera, se considero el Ciclo de la re-ingeniería de software presentado en el capítulo II .

**Captura :**

Se tomó el código fuente disponible en la documentación del sistema actual y se analizó la definición de los datos en general, así como su flujo y procesamiento. Como resultados de esta fase obtuvimos :

**- Lista de procesos con sus actividades:**

- A0. Sistema de Responsabilidades de Cartera**
  - A1. Proceso y envío de entidades manuales**
    - A1.1 Captura y registro de riesgos de entidades manuales
    - A1.2 Recibir y enviar diskettes
  - A2. Recepción, control y validación de información de los sistemas de cartera**
    - A2.1 Procesar información
    - A2.2 Control de arribo y validación de los importes de moneda extranjera
    - A2.3 Extraer los datos para complementar las altas del sistema
  - A3. Actualizar la información**
    - A3.1 Corrección de datos y cifras de los sistemas de cartera
      - A3.1.1 Integrar altas y saldos del sistema
      - A3.1.2 Corregir errores diagnosticados por Banco de México
      - A3.1.3 Corregir información y saldos de los sistemas de cartera
    - A3.2 Ajuste de cifras
      - A3.2.1 Integrar un movimiento de riesgos en forma global por el saldo de la cuenta 1310
        - A3.2.1.1 Verificar el saldo de cuenta 1310 para cada sucursal
        - A3.2.1.2 Crear un movimiento global para esa cuenta
      - A3.2.2 Ajustar cifras de acuerdo a la Cédula de agrupación en los totales por cuenta de diario mayor
    - A3.3 Solicitar la última corrida
      - A3.3.1 Checar cifras vs. Cédula de agrupación
      - A3.3.2 Enviar memorándum
  - A4. Generación de archivos y reportes**
    - A4.1 Generar cintas, listados y microfichas para Banco de México
    - A4.2 Generar archivo para el sistema S115
    - A4.3 Generar archivo para el Cig
    - A4.4 Generar listados y microfichas para Comisión Nacional Bancaria
    - A4.5 Generar listados para las Direcciones Regionales
    - A4.6 Generar listados para Pronósticos y Planeación Financiera
    - A4.7 Generar relación de Fondos para Información estratégica y Financiera



**A4.8 Generar listados para Normas y Técnicas de Crédito**

**A4.9 Generar listados para Cartera Metropolitana**

**A5. Entrega de información**

**A5.1 Enviar con un propio la cinta, listados y microfichas al Banco de México**

**A5.2 Enviar con un propio listado y microfichas a la Comisión Nacional Bancaria**

**A5.3 Enviar por valija listado a las Direcciones Regionales**

**A5.4 Enviar por valija listado a Pronósticos y Planeación Financiera**

**A5.5 Enviar por valija relación de Fondos a Información Estratégica y Financiera**

**A5.6 Enviar por valija listado a Normas y Técnicas de Crédito.**

**A5.7 Entregar listado a Cartera Metropolitana**

**- Las Entidades y Atributos****Actividades:**

<u>No. de actividad BANAMEX</u>	AC-NUACBA	9(05)
No. de actividad Banco de México	AC-NUACBM	9(08)
Nombre de la actividad	AC-NOMAC	X(30)
Actividad para menores	NUMAC-MEN	9(02)
No. de Sector	ST-NUSEC	9(02)
No. Sector Económico	EC-NUSECO	9(02)
Tipo de Actividad	MC-TIACT	9(02)
Clave válida	AC-CVEVALIDA	9(01)

**Localidades:**

<u>No. de localidad BANAMEX</u>	LC-NULOBA	9(05)
No. de localidad Banco de México	LC-NULOBM	9(08)
Nombre de la localidad	LC-NOMLOC	X(30)
No. Entidad Federativa	EF-NUENFE	9(02)

**Sectores:**

<u>No. de sector</u>	ST-NUSEC	9(02)
Nombre del sector	ST-NOMSEC	X(30)

**Fondos:**

<u>No. de Fondo</u>	FN-NUMFON	9(02)
Letra de Fondo	FN-NOMFON	X(30)
	FN-LETRA	X(01)

**Cuentas:**

<u>No. de cuenta BANAMEX</u>	CT-NUCTB	9(04)
No. cuenta Banxico	CT-NUCTBXICO	9(02)
Tipo de Cuenta	CT-TICTA	9(01)
Nombre de la cuenta	CT-NOMCTA	X(30)
No. de sector	ST-NUSEC	9(02)
No. de fondo	FN-NUMFON	9(02)

**Banca:**

<u>No. de Banca</u>	BC-NUBC	9(02)
Nombre de la Banca	BC-NOMBC	X(309)

**Tipo de Actividad:**

<u>Tipo de actividad</u>	MC-TIACT	9(02)
Nombre del tipo de actividad	NOM-ACT	X(30)
Clave de actividad para menores	MEN-CVEACT	9(06)

**Entidades Federativas**

<u>No. de entidad federativa</u>	E F-NUENFE	9(02)
Nombre de Entidad Federativa	EF-NOENFE	X(30)

**Riesgos:**

<u>No. sucursal</u>	SC-NUMSUC	(804)
<u>No. de cliente</u>	CL-NUMCTE	9(13)
<u>Cuenta origen</u>	RS-CTAOR	9(04)
Cuenta contable	RS-CTACONT	9(04)
Importe	RS-IMPOR	9(10)
Tipo de moneda	MN-TIMO	9(03)
Tipo de cartera	CR-TICARTE	9(02)
Sistema origen	CL-SISORIGEN	9(02)
Nivel de riesgo	RS-NIVRS	X(01)
No. de CSI	CL-NUCSI	X(02)
No. de fondo	FN-NUMFON	9(02)
RFC	CL-RFC	X(13)

**Monedas:**

<u>No. de moneda</u>	MN-NUMON	9(03)
Nombre de moneda	MN-NOMON	X(22)
Tipo de cambio	MN-TICAM	9(03)
Tipo de moneda	MN-TIMON	9(03)

**Clientes:**

<u>No. de sucursal</u>	SC-NUMSUC	9(04)
<u>No. cliente</u>	CL-NUMCTE	X(13)
Nombre de cliente	CL-NOMCTE	X(55)
RFC	CL-RFC	X(13)
No. actividad Banco de México	AC-NUACBM	9(05)
No. localidad Banco de México	LC-NULOBM	9(05)
No. de Banca	BC-NUMBC	9(03)
No. de sector	ST-NUSEC	9(02)
Tipo de vivienda	RS-TIVIV	9(02)
Clave de aval	CL-CVEAVAL	9(06)
Grupo Económico	CL-CPOECO	9(02)
Sistema origen No. CSI	CL-SISORIGEN	9(02)
Nivel de riesgo	RS-NIVRS	X(01)

**Sucursales**

<u>No. sucursal</u>	SC-NUMSUC	9(04)
Nombre de la sucursal	SC-NOMBRE	X(30)
Dirección regional	DI-NUDIREG	9(04)

**CAPÍTULO IV****APLICACION DE LA RE-INGENIERIA DE SOFTWARE****Tipo de Cartera****Tipo de cartera****CR-TICARTE 9(02)****Nombre del tipo de cartera****CR-NOMTICARTE X(30)****Sector Económico****No. de sector económico****EC-NUSECO 9(02)****Nombre de sector económico****EC-NOMSEC X(20)****Actividades para menores****Cve. act. para menores****MEN-CVACT 9(06)****Nombre de la act. para menores****MEN-NOMACT X(30)****- Relaciones con su matriz .**

**TESIS SIN PAGINACION**

**COMPLETA LA INFORMACION**

RESPONSABILIDADES DE CARTERA

ENTIDADES VS. APLICACIONES

PROYECTO-RESPONSABILIDADES DE CARTERA  
 VERSION: RND DE APLICACIONES SISTEMAS  
 FECHA: JUNIO 1993

	ALTA A CLIENTES	MODIF. A CLIENTES	CONSULTA A CLIENTES	ALTA A TIPOS	MODIF. A TIPOS	CONSULTA A TIPOS	CONSULTA A TAJAS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	CONSULTA A ESTADOS DE ESTADO	
ACTIVIDADES																							X	
LOCALIDADES																								X
SECTOR																								
FONDO																								
CUENTAS									X						X		X	X						
BANCA																								
TIPO DE ACTIVIDAD										X														
ENTIDADES FEDERATIVAS																		X						X
RIESGOS				X	X	X			X			X	X											
CLIENTES	X	X	X									X												
MONEDAS									X						X		X	X						
SUCURSALES														X			X							
TIPO DE CARTERA										X									X					
SECTOR ECONOMICO																								
RANGOS DE IMPORTE										X														
ACTIV. PARA MENOR										X														
DIR. REGIONAL																								

LEYES: X: PL. DISPONIBLES Y: PL. ESTABLES

RESPONSABILIDADES DE CARTERA

ENTIDADES VS. APLICACIONES

PROYECTO RESPONSABILIDADES DE CARTERA  
 VERSION: RED DE APLICACIONES RESPONSABLES  
 FROM: 08/02/1993

	MONTE DE LA CARTERA TOTAL	MONTE DE LA CARTERA ACTIVA	MONTE DE LA CARTERA PASIVA	MONTE DE LA CARTERA NETA	MONTE DE LA CARTERA TOTAL	MONTE DE LA CARTERA ACTIVA	MONTE DE LA CARTERA PASIVA	MONTE DE LA CARTERA NETA	MONTE DE LA CARTERA TOTAL	MONTE DE LA CARTERA ACTIVA	MONTE DE LA CARTERA PASIVA	MONTE DE LA CARTERA NETA	MONTE DE LA CARTERA TOTAL	MONTE DE LA CARTERA ACTIVA	MONTE DE LA CARTERA PASIVA	MONTE DE LA CARTERA NETA
ACTIVIDADES																
LOCALIDADES																
SECTOR																
FONDD																
CUENTAS																
BANCA																
TIPO DE ACTIVIDAD	X															
ENTIDADES FEDERATIVAS																
RIESGOS																
CLIENTES																
MONEDAS																
SUCURSALES																
TIPO DE CARRERA																
SECTOR ECONOMICO																
RANGOS DE IMPORTE																
ACTIV. PARA MENOR	X															
DIR. REGIONAL																

LEYENDA: X=ACTIVO, V=PASIVO, D=DEBITO

RESPONSABILIDADES DE CARTERA

DOCUMENTOS VS. PROCESOS

DIRECCION: RESPONSABILIDADES DE CARTERA  
VERSION:  
FECHA: SEPTIEMBRE 1998.

	ANEXO 1	ANEXO 2	ANEXO 3	ANEXO 4	ANEXO 5	ANEXO 6	ANEXO 7	ANEXO 8	ANEXO 9	ANEXO 10	ANEXO 11	ANEXO 12	ANEXO 13	ANEXO 14	ANEXO 15	ANEXO 16	ANEXO 17	ANEXO 18	ANEXO 19	ANEXO 20	ANEXO 21	ANEXO 22	ANEXO 23	ANEXO 24	ANEXO 25			
RELACION DE CUENTAS Y SUBCUENTAS																												
REPORTE DE SALDOS DE LA DIVISION LINEA BANAMEX																												
RELACION DE RESPONSABILIDADES DE CARTERA PARA BANCO DE MEXICO																												
LISTADO DE RELACION DE CLIENTES DE CARTERA																												



**Ingeniería reversiva :**

Se llevo a cabo el proceso de análisis del sistema actual para identificar los componentes del mismo y sus interrelaciones con otros sistemas de BANAMEX. obteniendo

- Diagrama de flujo de procesos desde el nivel 0 hasta el nivel 4.
- Diagrama general de fases del proceso.

**Expansión :**

Tomando los resultados de la ingeniería reversiva se obtuvo una lista de requerimientos de entrada, de acuerdo a entrevistas con el usuario.

**Ingeniería hacia adelante :**

Se retomaron las políticas, objetivos y beneficios del sistema, las cuales son :

- Se simplificará la intervención del usuario
- Se tendrá la información integra de los sistemas operativos
- Se integrará la validación de Banco de México al sistema para evitar rechazos por parte de éste
- Posibilidad de retroalimentación a los sistemas operativos con la información correcta
- Se evitará al máximo el consumo de papel
- Utilización de estructuras externas que faciliten el mantenimiento
- Se utilizará sistema de catálogos generales
- Selección flexible de los movimientos
- Se contará con la información para procesos estadísticos
- Eliminación de la impresión de información no utilizada
- Utilización de la tecnología de Bases de datos
- Utilización de herramientas modernas para el desarrollo de sistemas
- Operación amigable para el nuevo sistema
- Se disminuirá la interacción de usuario-operación
- Oportunidad y veracidad de la información.

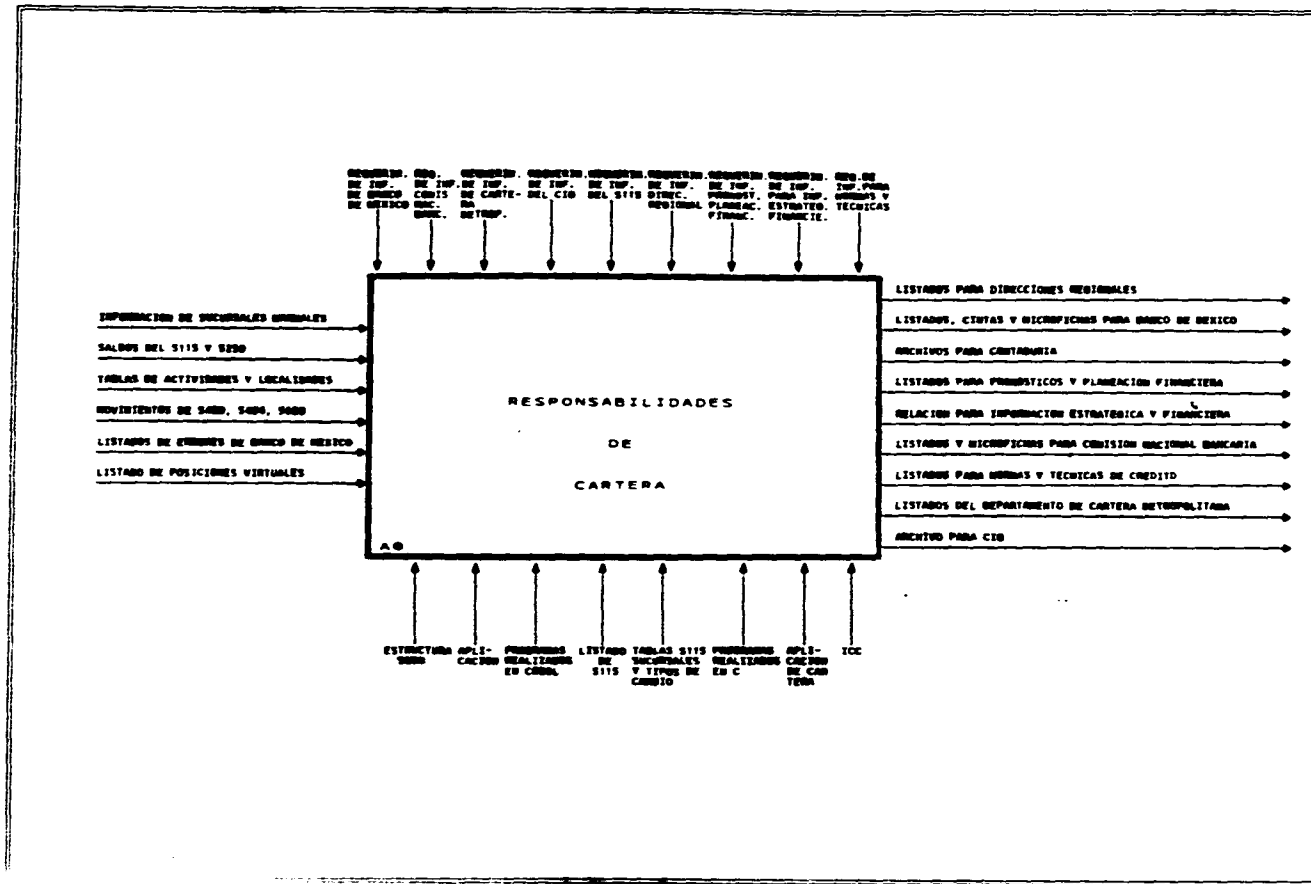
**Optimización :**

Para conocer las necesidades del usuario se realizaron entrevistas y se aplicaron cuestionarios, mediante los cuales concluimos que sus principales necesidades eran la optimización de tiempos de proceso, reducción en los volúmenes de papel y de la intervención manual del usuario.

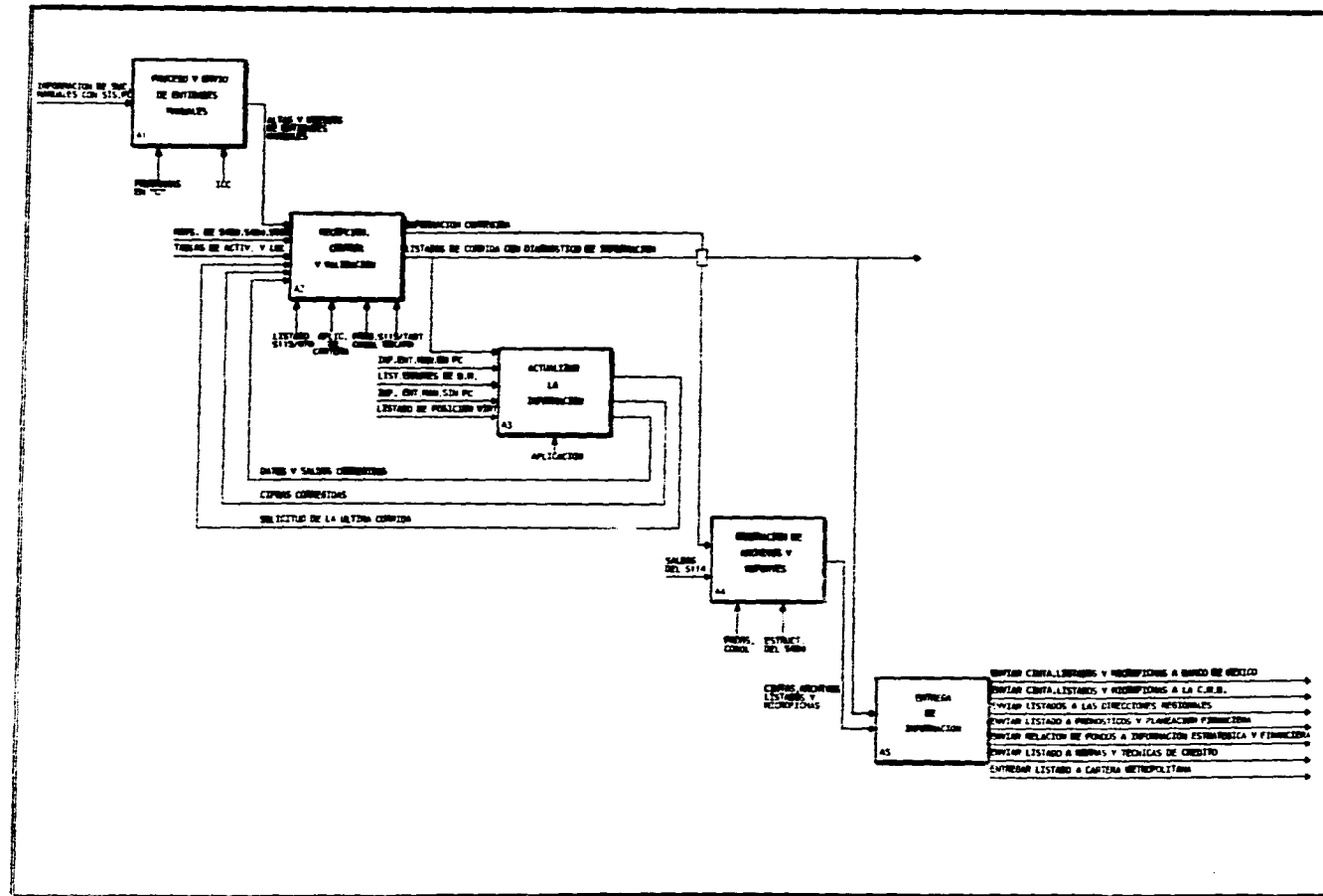
**TESIS SIN PAGINACION**

**COMPLETA LA INFORMACION**

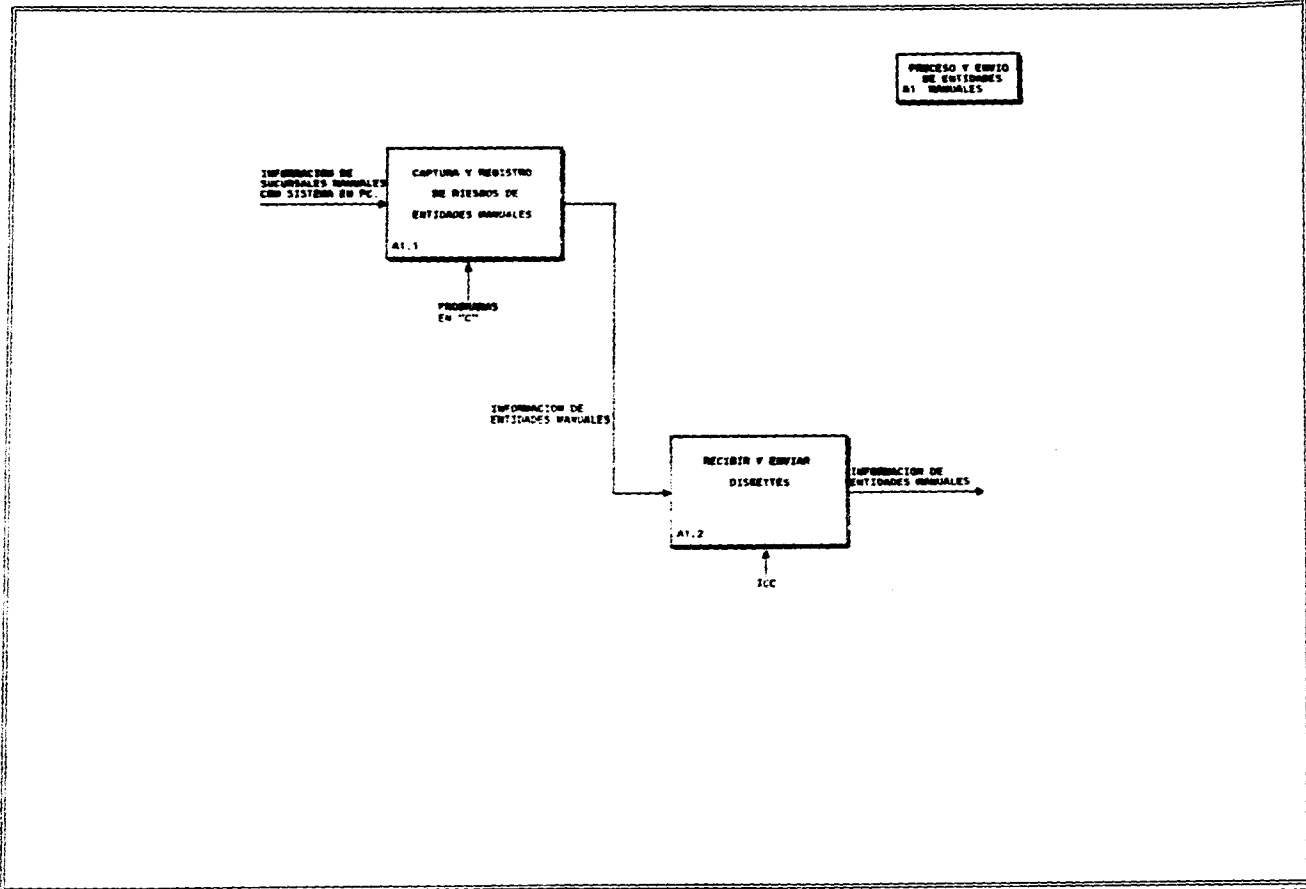
## RESPONSABILIDADES DE CARTERA



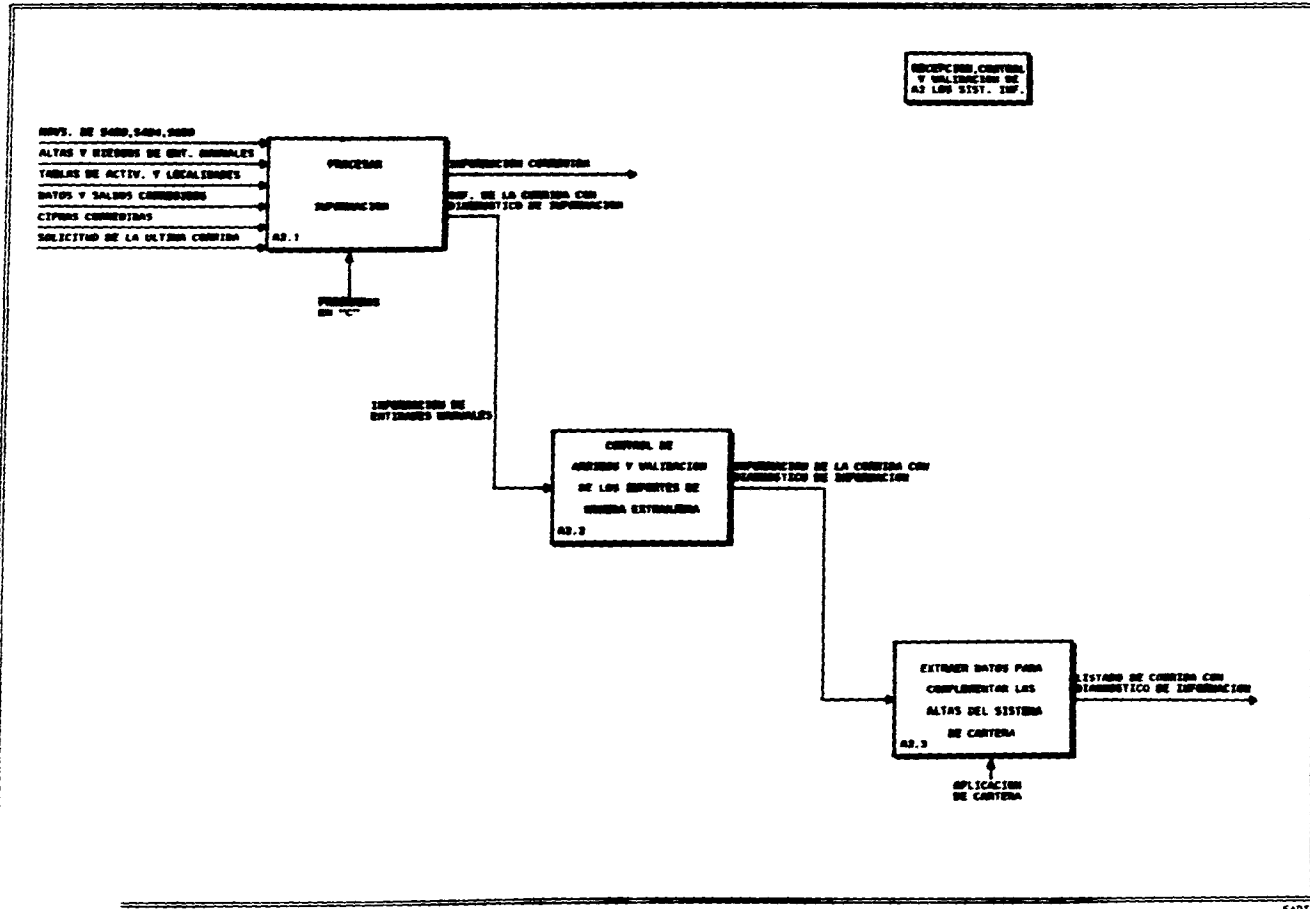
RESPONSABILIDADES DE CARTERA



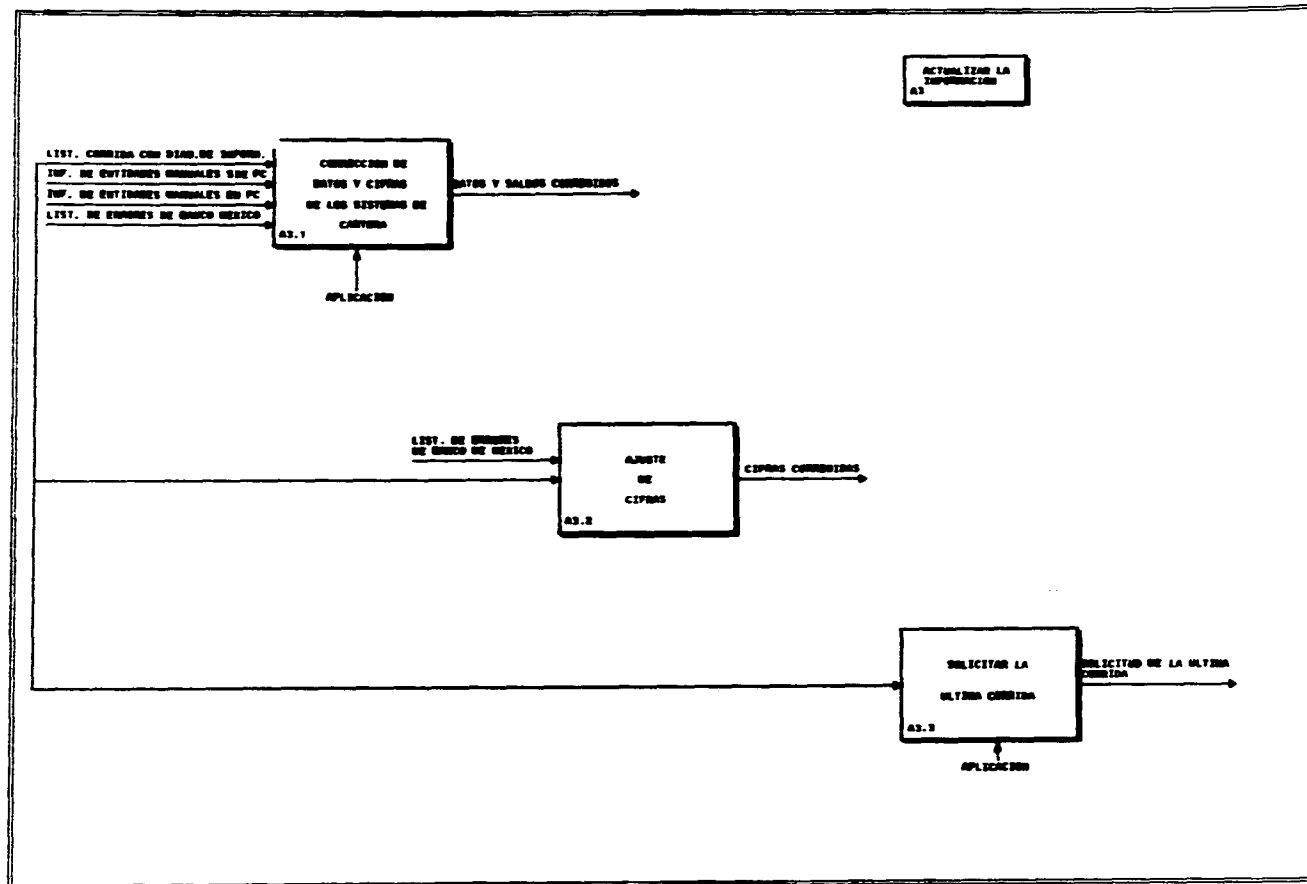
RESPONSABILIDADES DE CARTERA



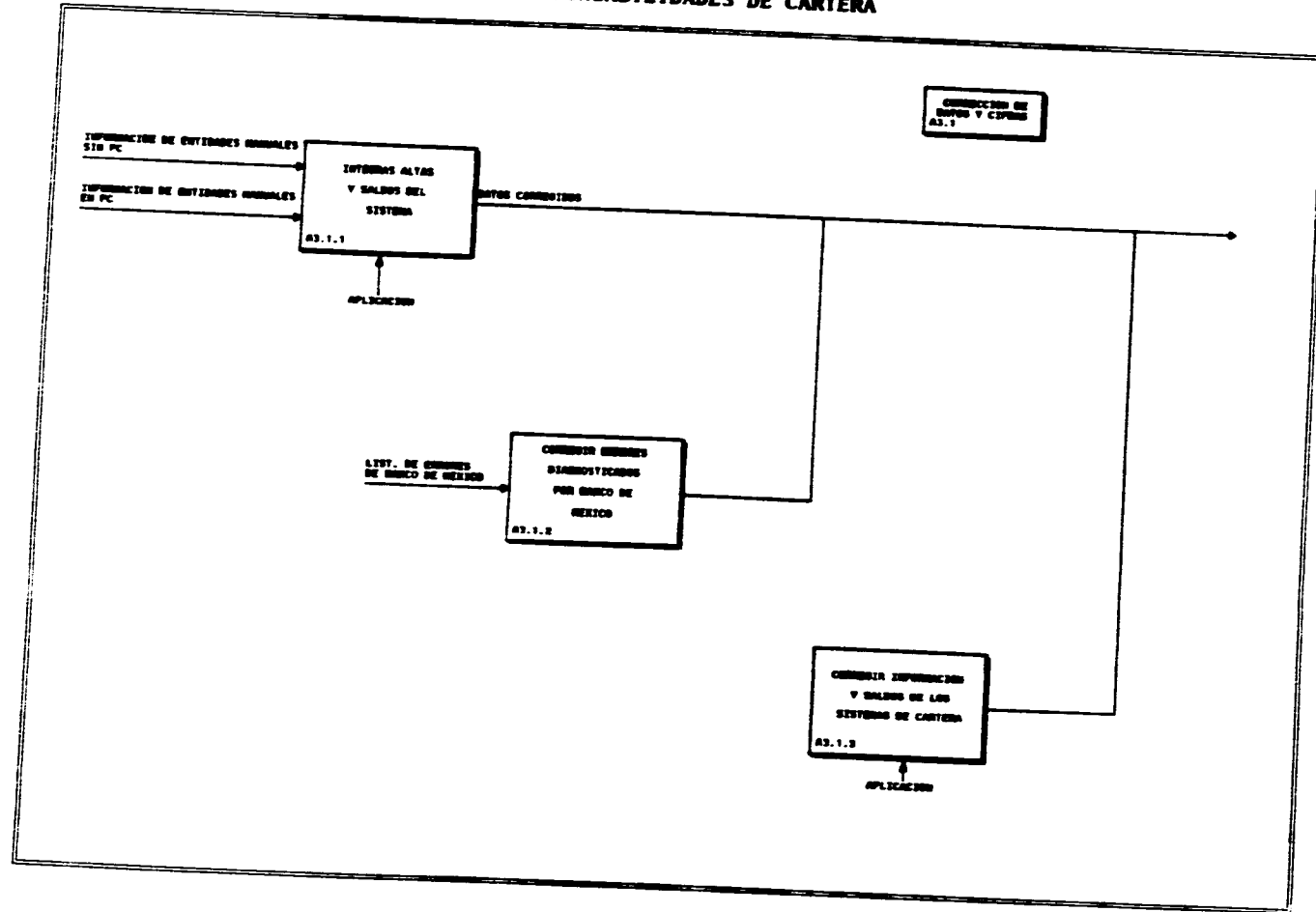
# RESPONSABILIDADES DE CARTERA



## RESPONSABILIDADES DE CARTERA

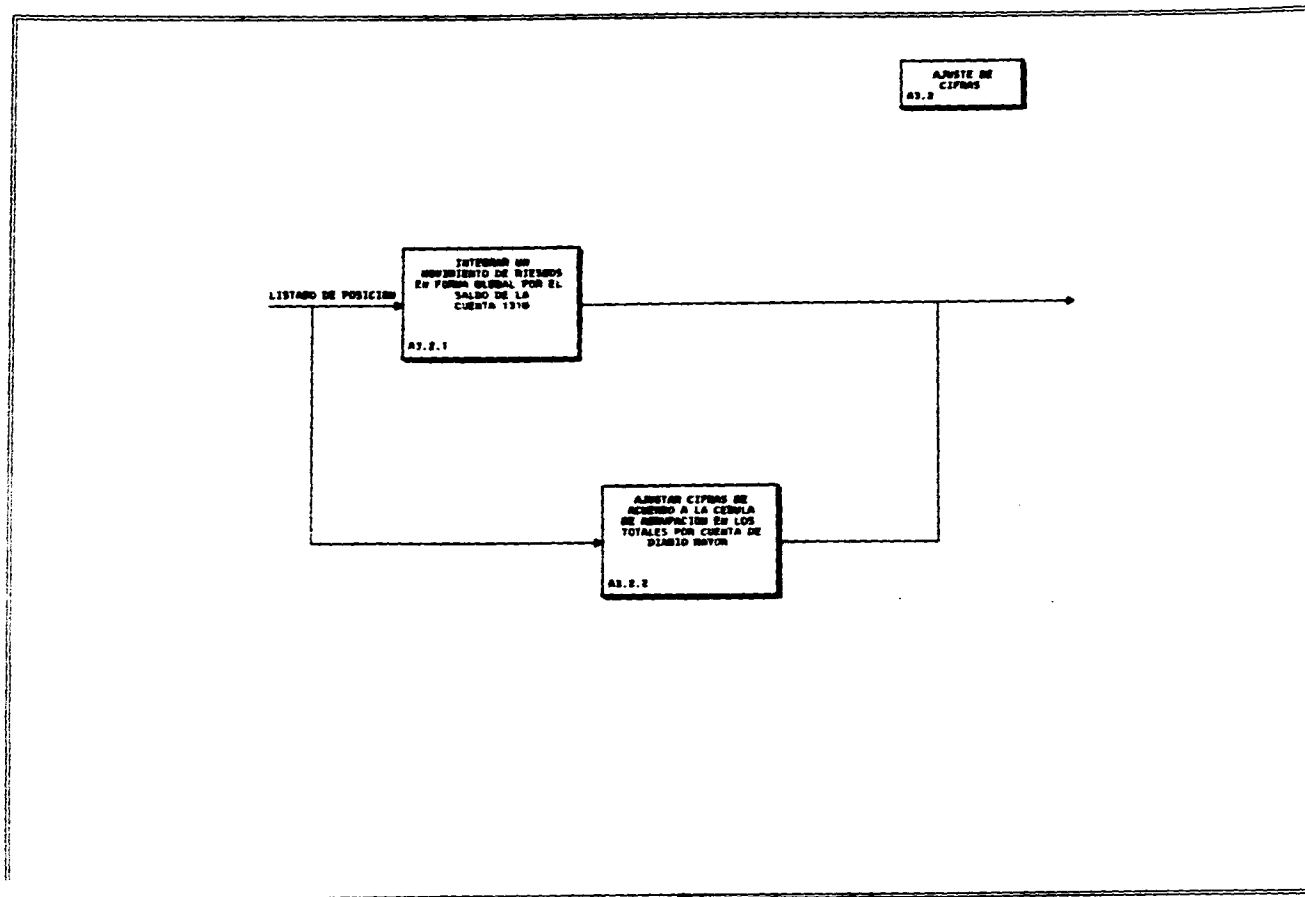


# RESPONSABILIDADES DE CARTERA

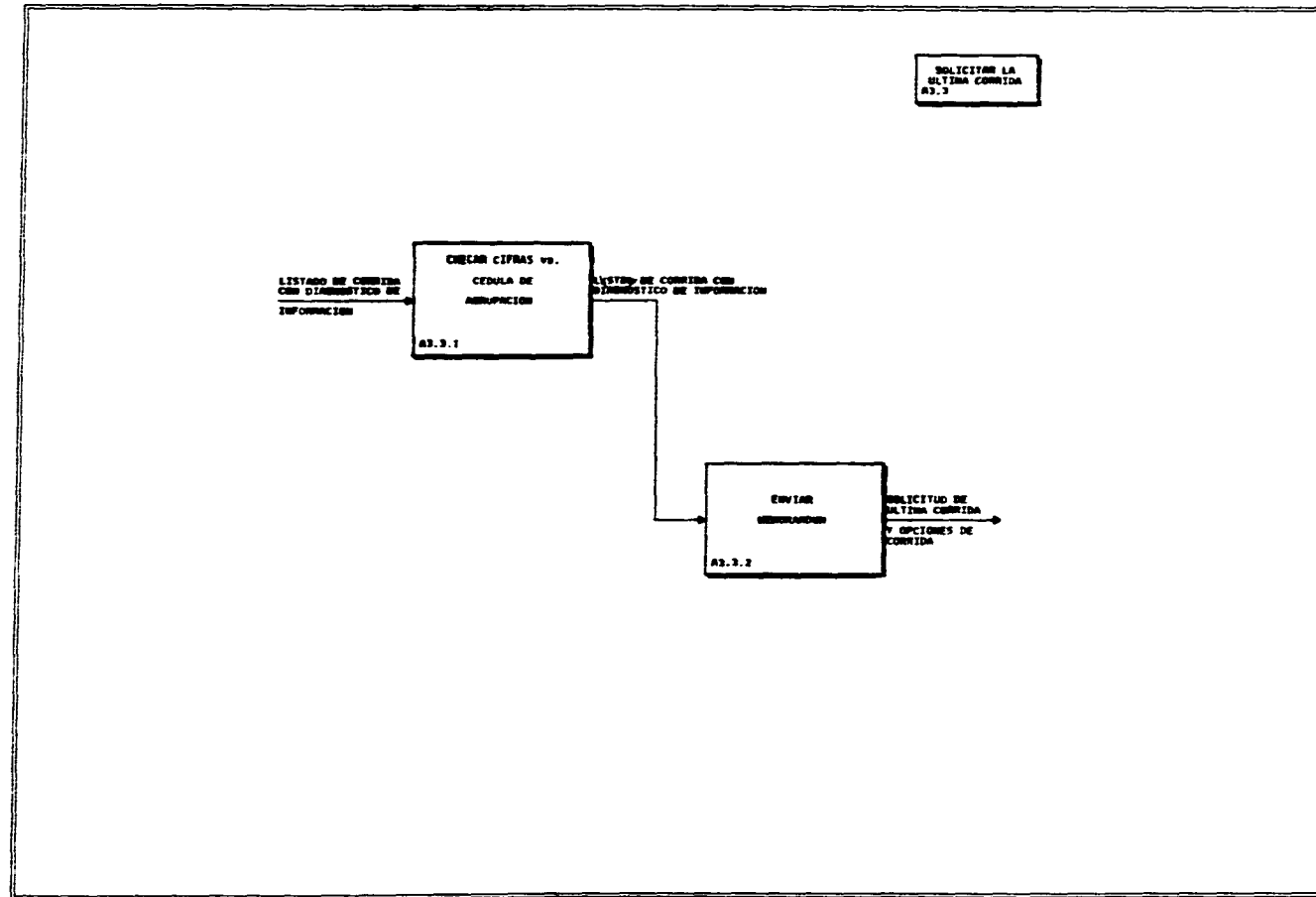




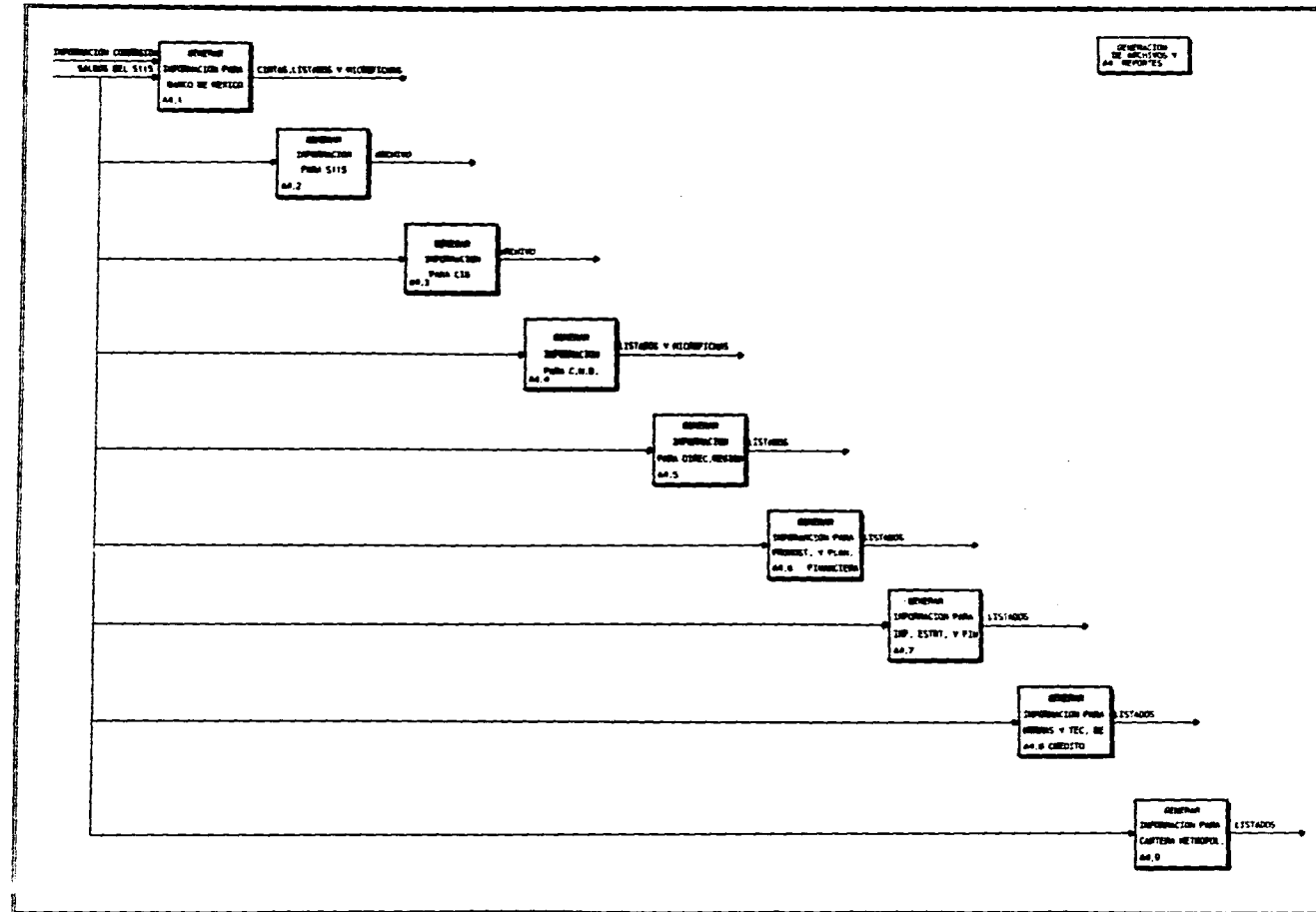
## RESPONSABILIDADES DE CARTERA



# RESPONSABILIDADES DE CARTERA

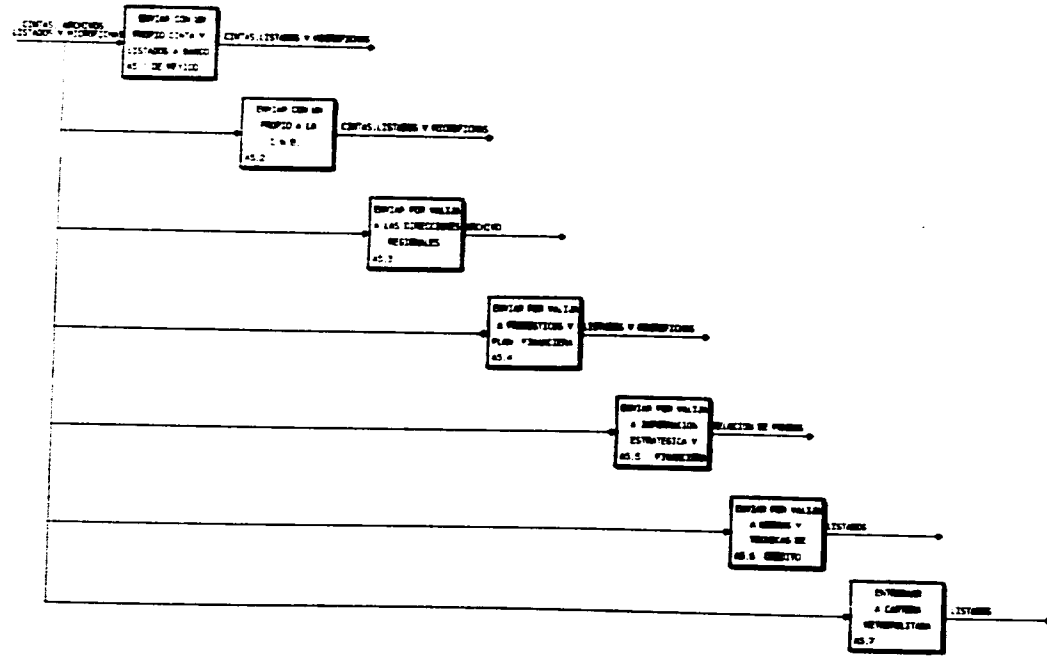


### RESPONSABILIDADES DE CARTERA



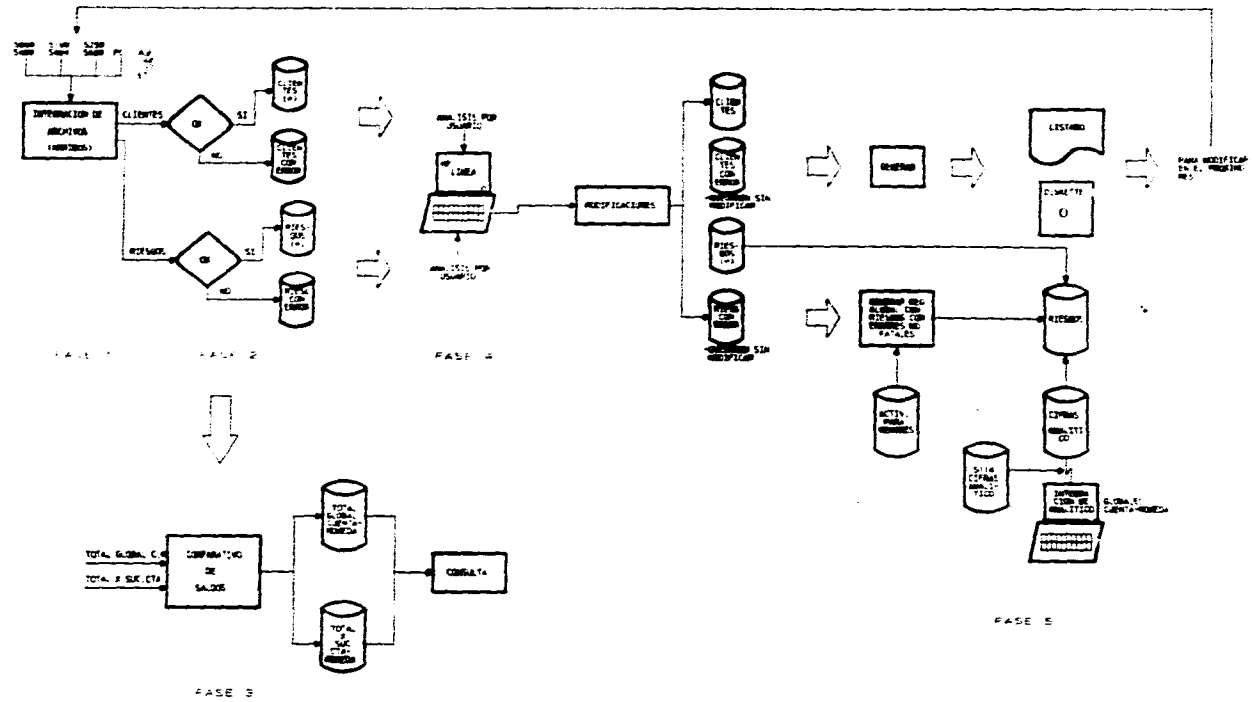
RESPONSABILIDADES DE CARTERA

SECRETARÍA DE ECONOMÍA



RELACION DE RESPONSABILIDADES DE CARTERA

INFORMACION CON TOTAL  
DE RESPONSABILIDADES DE CADA UNO  
DE LOS CLIENTES



INFORMACION CON TOTAL  
DE RESPONSABILIDADES DE CADA UNO  
DE LOS CLIENTES

RESPONSABILIDADES DE CARTERA

IDENTIFICACION DE LOS REQUERIMIENTOS DE ENTRADA

<u>FUENTES DE INFORMACION</u>	<u>FRECUENCIA DE ACTUALIZACION</u>	<u>VOLUMENES DE ENTRADA</u>	<u>MODO</u>	<u>REQUERIMIENTO DE VALIDACION</u>	<u>REQUERIMIENTO DE SEGURIDAD</u>
MOVIMIENTOS DE LOS SIS-TEMAS 5000, 5400, 5600 Y PC	MENSUAL	101,426	SISTEMA	EXISTENCIA DE DE LOS SIST. EXTERNOS	CARGA
SALDOS DEL 5114	MENSUAL		SISTEMA	EXISTENCIA DE SALDOS DEL 5114	COMPARATIVO
CONCENTRACION CONTABLE 5052 (CUENTA-SUCURSAL)	MENSUAL		SISTEMA	EXISTENCIA DE CTA/-SUCURSAL DEL 5052	COMPARATIVO
ESTRUCTURA DEL 5000	MENSUAL		SISTEMA	EXISTENCIA DE LA ESTRUCTURA	CARGA
5070 TIPOS DE CAMBIO	MENSUAL		SISTEMA	EXISTENCIA DE LOS TIPOS DE CAMBIO	CARGA
5136/FILE/TABLAS060	MENSUAL		SISTEMA	EXISTENCIA DEL ARCHIVO DE TABLAS	CARGA
CATALOGO DE TIPO DE CARTERA	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DEL TIPO DE CARTERA	CARGA Y ACTUALIZACION
CATALOGO DEL SECTOR ECONOMICO PARA SHYCP	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DEL SECTOR ECONOMICO	CARGA ACTUALIZACION
CATALOGO DE ACT. PARA MENORES	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DE ACT. PARA MENORES	CARGA ACTUALIZACION
CATALOGO DE CUENTAS	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DE CUENTAS	CARGA Y ACTUALIZACION
CATALOGO DEL TIPO DE ACT.	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DEL TIPO DE ACT.	CARGA Y ACTUALIZACION
CATALOGO DE BANCAS	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DE BANCAS	CARGA Y ACTUALIZACION
CATALOGO DE FONDOS	MENSUAL		MANUAL	EXISTENCIA DEL CATALOGO DE FONDOS	CARGA Y ACTUALIZACION

RESPONSABILIDADES DE CARTERA

IDENTIFICACION DE LOS REQUERIMIENTOS DE APLICACION

<u>APLICACIONES</u>	<u>USUARIOS</u>	<u>FRECUENCIA</u>	<u>MODO</u>	<u>REQUERIMIENTOS DE SEGURIDAD</u>
ALTA A CLIENTES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	TERMINAL	USUARIO AUTORIZADO
MODIFICACIONES DE CLIENTES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA	USUARIO AUTORIZADO
CONSULTA A CLIENTES	PERSONAL DE CONTADURIA CORPORATIVA	EVENTUAL	PANTALLA	USUARIO QUE LO REQUIERA
ALTA A RIESGOS	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	TERMINAL	USUARIO AUTORIZADO
MODIFICACIONES DE RIESGOS	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA	USUARIO AUTORIZADO
CONSULTA A RIESGOS	PERSONAL DE CONTADURIA CORPORATIVA	EVENTUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO QUE LO REQUIERA
CONSULTA A CIFRAS DE ARCHIVO POR XFER	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA DEL GLOBAL CUENTA-MONEDA	PERSONAL DE CONTADURIA CORPORATIVA	EVENTUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA A LAS CIFRAS DE CONTROL	PERSONAL DE CONTADURIA CORPORATIVA	EVENTUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA A LA DESCRIPCION DE SALDOS EN CUENTAS	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA DE ARCH-SALDOS CARTERA VENCIDA Y VIGENTE	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA DE CIFRAS DE CONTROL POR FONDOS	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO
CONSULTA DEL COMPARATIVO DE SALDOS VS. DIARIO MAYOR POR CUENTA	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA Y OPCION DE IMPRE.	USUARIO AUTORIZADO

RESPONSABILIDADES DE CARTERA

IDENTIFICACION DE LOS REQUERIMIENTOS DE APLICACION

<u>APLICACIONES</u>	<u>USUARIOS</u>	<u>FRECUENCIA</u>	<u>MODOS</u>	<u>REQUERIMIENTOS DE SEGURIDAD</u>
CONSULTA A LA ACTUALIZACION DEL ARCHIVO DE TABLAS DE ACTIVIDADES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA CON OPC. DE IMPRE.	USUARIOS AUTORIZADOS
CONSULTA A LA ACTUALIZACION DEL ARCHIVO DE TABLAS DE LOCALIDADES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PANTALLA CON OPC. DE IMPRE.	USUARIOS AUTORIZADOS
REPORTE DE LISTADO DE TOTALES POR CUENTA	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	CINTA Y PAPEL	NINGUNO
REPORTE DEL CIERRE DE RESPONSABILIDADES DE CARTERA	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE LA RELACION DE TOTALES DE CTAS MAYORES Y MENORES	BANCO DE MEXICO Y COMISION NACIONAL BANCA-RIA	MENSUAL	PAPEL	NINGUNO
REPORTE DE RECHAZOS POR CLIENTE	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE LISTADO DE MODIFICACIONES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE PRESTATARIOS MENORES	BANCO DE MEXICO Y COMISION NACIONAL BANCA-RIA	MENSUAL	PAPEL Y MICROFICHA	NINGUNO
REPORTE DE MAYORES	BANCO DE MEXICO Y COMISION NACIONAL BANCA-RIA	MENSUAL	PAPEL Y CINTA	NINGUNO
REPORTE DE SALDOS POR CLIENTE	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE LA CLAS. DE CARTERA POR ACTIVIDAD	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE LA AUDITORIA DE CINTAS BANKICO MAYORES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO



RESPONSABILIDADES DE CARTERA

IDENTIFICACION DE LOS REQUERIMIENTOS DE APLICACION

<u>APLICACIONES</u>	<u>USUARIOS</u>	<u>FRECUENCIA</u>	<u>MODOS</u>	<u>REQUERIMIENTOS DE SEGURIDAD</u>
REPORTE DE DISTRIBUCION DE LA CARTERA POR SECTOR Y ENT. FED. PARA LA CARTERA PROPIA	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DE DISTRIBUCION DE LA CARTERA POR SECTOR Y ENT. FED. PARA LA CARTERA REDESCONTADA	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DEL LISTADO DE ALTAS DE CLIENTES	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO
REPORTE DEL LISTADO ALFABETICO POR SECTOR	PERSONAL DE CONTADURIA CORPORATIVA	MENSUAL	PAPEL	NINGUNO

## RESPONSABILIDADES DE CARTERA

### CUESTIONARIO

NOMBRE DEL LISTADO: \_\_\_\_\_

GENERADO POR EL PASO: \_\_\_\_\_

CON QUE FRECUENCIA SE RECIBE: \_\_\_\_\_

DESPUES DE UN EVENTO ESPECIFICO O CUANDO SE SOLICITA: \_\_\_\_\_

QUIENES SON LOS USUARIOS: \_\_\_\_\_

UBICACION: \_\_\_\_\_ TEL: \_\_\_\_\_ EXT.: \_\_\_\_\_

TIPO DE LISTADO POR SU USO: \_\_\_\_\_  
(CONTROL, VALIDACION, CONSULTA)

SE CONSERVA?:     SI     NO

CUANTO TIEMPO: \_\_\_\_\_

CON QUE FRECUENCIA SE CONSULTA: \_\_\_\_\_

GENERA: MICROFICHA:     PAPEL:     CINTA:

SE UTILIZAN FORMAS ESPECIALES (DESCRIBIR): \_\_\_\_\_

CUANTAS COPIAS GENERA Y A QUIEN SE ENTREGAN: \_\_\_\_\_

CUAL ES EL VOLUMEN DEL LISTADO: \_\_\_\_\_

SE DEVUELVEN ALCUNAS COPIAS: \_\_\_\_\_

DESCRIPCION DEL LISTADO: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

CLASIFICADO POR: \_\_\_\_\_

CORDTES O TOTALES: \_\_\_\_\_

QUE UTILIDAD TIENE ESTE LISTADO: \_\_\_\_\_

COMENTARIOS: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

PERSONA ENTREVISTADA: \_\_\_\_\_

FECHA DE ENTREVISTA: \_\_\_\_\_

RESPONSABILIDADES DE CARRERA

CUESTIONARIO

NOMBRE DEL LISTADO: \_\_\_\_\_  
GENERADO POR EL PASO: 010  
CON QUE FRECUENCIA SE RECIBE: 1 VEE EN CADA CORRIDA  
DESPUES DE UN EVENTO ESPECIFICO O CUANDO SE SOLICITA: CADA CORRIDA  
QUIENES SON LOS USUARIOS: COMIB. Z. D. D.  
UBICACION: \_\_\_\_\_ TEL: \_\_\_\_\_ EXT.: \_\_\_\_\_  
TIPO DE LISTADO POR SU USO: CONTROL Y CONSULTA  
(CONTROL, VALIDACION, CONSULTA)  
SE CONSERVA: SI  NO   
CUANTO TIEMPO: UN AÑO  
CON QUE FRECUENCIA SE CONSULTA: CONSTANTEMENTE  
GENERA: MICROFICHA:  PAPEL:  CINTA:   
SE UTILIZAN FORMAS ESPECIALES (DESCRIBIR): NO  
CUANTAS COPIAS GENERA Y A QUIEN SE ENTREGAN: 1  
CUAL ES EL VOLUMEN DEL LISTADO: 8 HOJAS  
SE DEVUELVEN ALGUNAS COPIAS: NO  
DESCRIPCION DEL LISTADO: GENERA DIVERSOS REPORTES CON CIFRAS DE CONTROL Y CONSULTA PARA INFORMACION GERENCIAL  
CLASIFICADO POR: \_\_\_\_\_  
CORTES O TOTALES: \_\_\_\_\_  
QUE UTILIDAD TIENE ESTE LISTADO: CIFRAS DE CONTROL Y CREAR  
COMENTARIOS: \_\_\_\_\_  
\_\_\_\_\_

PERSONA ENTREVISTADA: Eloy Estrella  
FECHA DE ENTREVISTA: 10-Marzo-93

**Generación :**

Con base en el análisis anterior se comenzó el diseño del sistema utilizando la herramienta CASE LDA (LINC DESIGN ASSISTANT)

LDA es una herramienta basada en computadoras personales, la cual en conjunción con el medio de desarrollo de LINC, soporta las siguientes fases de desarrollo de sistemas:

- Análisis
- Diseño
- Programación
- Revisión

A continuación se muestra la estructura de LDA (las partes que contiene la herramienta para tener un esquema del proyecto):

- Modelo
- Area Funcional
- Actividad
- Objeto
- Dato

**LDA en el ciclo de vida del desarrollo de sistemas.****1.- Investigación del sistema.**

- Identificar metas y objetivos del negocio.
- Nombrar el modelo del negocio.  
Los requerimientos de información para una organización se definen en la forma del modelo del negocio.
- Registrar las reglas del negocio.
- Identificar actividades registradas.
- Registro de actividades.
- Registro de áreas funcionales.
- Asociación de actividades/áreas funcionales.
- Producir la descripción general del sistema.
- Permite la impresión de:
  - Objetivos del negocio
  - Gráfica del modelo total del negocio
  - Gráfica del área funcional que será seleccionada para el desarrollo
  - Reglas del negocio para el modelo completo, describiendo prioridades para la implantación de áreas funcionales
  - Documentación completa de todas las actividades.

**2.- Definición del sistema**

- Definir el modelo de objetos LINC
- Adicionar o eliminar objetos de la gráfica de una actividad
- Adicionar o eliminar lógicas globales
- Definir nombre de datos y pantallas
- Construir el diccionario de datos
- Definición de datos usando la ventana de datos
- Registro de datos usando la matriz de objetos/datos
- Definir las reglas del negocio
- Definir un componente
- Construir un prototipo
- Definir interfaces externas
- Definir diferentes vistas
- Definir relaciones entre objetos
- Definir reportes
- Definir formatos de reportes
- Revisión técnica de la funcionalidad
- Revisión de la matriz de perfiles o vistas
- Analizar eficiencia utilizando una auditoria de diseño
- Estimar el tamaño del proyecto mediante el análisis de puntos por función
- Revisar la matriz de evento
- Documentar los requerimientos de seguridad
- Documentar los requerimientos de integridad
- Revisar los límites de implantación
- Producir documentación.

**3.- Desarrollo del sistema**

- Cargar el modelo en el medio ambiente de desarrollo de LINC.
- Completar las especificaciones en el medio ambiente de desarrollo de LINC y generar el sistema de información.

**4.- implantación del sistema**

- LDA no se utiliza durante esta fase del ciclo de vida de desarrollo de sistemas.

**5.- Revisión del sistema**

• **Auditoría de diseño**

- . Racionalización del evento
- . Análisis de datos
- . Racionalización de perfiles
- . Utilización de objetos (identificación del código muerto)
- . Análisis de palabras reservadas

• **Análisis de puntos por función**

- . Documentación para formar y soportar la base del reporte de revisión del sistema
- . Pintado de pantallas y reportes basados en PC's

**Hardware para LDA:**

- Computadora personal Unisys 386 o compatible
- 5 MB de memoria RAM como mínimo
- 20 MB de espacio en disco duro como mínimo
- Monitor EGA o VGA
- Cualquier dispositivo de ratón calificado para uso bajo OS/2 o Windows.
- Impresora
- Llave de seguridad SENTINEL-C versión B.

**Software para LDA:**

- MS-DOS versión 3.0 o posterior
- Windows versión 3.0
- Fonts courier, time y Helvética instaladas bajo windows.
- Una utilería para transferencia de archivos desde y hacia el sistema central (ej. Infoview.)
- Procesador de palabras (opcional)

**Comentarios sobre LDA:**

LDA puede ser monousuario, es decir, sólo una persona puede accederlo y con un sólo modelo. A partir de la versión 3.0 puede ser multiusuario.

Para la instalación de LDA en Windows es necesario adicionar parámetros específicos, para el archivo WIN.INI

Para su instalación con OS/2 es necesario modificar el archivo config.sys.

En LDA se puede cargar una área funcional ya existente, para lo cual es necesario especificar en los elementos que ya existen, si éste será reemplazado o no.

Al cargar un modelo de LDA a LINC, las pantallas de entrada son cargadas como eventos.

Cuando se inicia un proyecto LDA obliga a estandarizar el diccionario de datos, permite la creación del prototipo del sistema, se puede medir la complejidad del sistema y programar el tiempo de duración del proyecto por medio del análisis de puntos por función (FPA), permite establecer las reglas bajo las cuales se va a regir el sistema, permite hacer el pintado de pantallas y frames de reportes.

**LDA tiene el siguiente contenido:**

- **Uso de diccionario**
- **Dibujo de pantallas**
- **Dibujo de reportes**
- **Uso de gráficas**
- **Uso de matrices**
- **Auditoría de diseño**
- **Análisis de puntos por función para evaluación de tiempo de desarrollo del sistema.**
- **Generación de código de LINC de la PC a la serie A.**
- **Ingeniería reversiva de código de LINC de la serie A a la PC.**



Ahora presentamos una breve descripción de LINC software que será utilizado para la fase de desarrollo :

**LINC (Compilador Interactivo para Línea y Red)**

Es una herramienta diseñada para utilizarse en todos los modelos de UNISYS de las series B1000, Medium System/V Series.

LINC reduce los tiempos de generación de sistemas en línea, asiste a los diseñadores en el desarrollo de sistemas en LINCII que tienen editor de sintaxis en línea, pantalla para dibujar, sistema de seguridad, capacidad de configuración para red, diccionario datos y ayuda en línea.

**Hardware para LINC**

- Terminal T-27 conectada a un equipo A-12 para desarrollo
- Equipo A-19 para producción para el procesamiento del desarrollo

**Software para LINC**

- DMS II
- COBOL
- COMS
- TPS para impresión de listados

La aplicación de la herramienta CASE LDA dio como resultado los siguientes esquemas :

- Reglas del negocio
- Directorio del modelo: es un árbol gráfico del diseño general del sistema donde presentamos todas las actividades con sus atributos.
- Diagrama de flujo general del negocio con sus relaciones.
- Diccionario de datos.
- Matriz del área funcional con sus actividades.
- Diagramas para mostrar cada una de las actividades con sus relaciones .
- Análisis de puntos por función.
- Muestra de reportes y pantallas del sistema .

**TESIS SIN PAGINACION**

**COMPLETA LA INFORMACION**

**RESPONSABILIDADES DE CARTERA**

Desarrollo de un nuevo sistema que nos permita el registro y control de la información de los Riesgos de Cartera, para proporcionar al Banco de México la información que requiera, así como para Balance Oficial y toma de decisiones.

**Pasos a seguir en el sistema:****Al inicio del sistema:**

- Generar archivos vacíos de Catálogos para que el usuario integre los datos.
- Generar archivos de catálogos de Actividades Banxico y Localidades Banxico en forma automática para ser complementados por el usuario (si es el caso).

**Cada mes:**

- Borrar información del mes pasado en archivos de saldos.
- Cargar automáticamente Sucursales y Monedas (tipo de cambio)
- Modificación de catálogos de datos por el usuario (si es el caso)
- Cargar y validar la información de los sistemas origen (S400,S404,S600,etc)
- Actualizar Riesgos y Clientes
- Integrar los saldos del Sistema Central Contable (S115) y de la Concentración de Saldos (S052)
- Consulta de Riesgos, Clientes, Totales y sus diferencias para su modificación
- Integrar los saldos de Analítico en el total de Cuentas para la adecuación de Cifras en forma automática
- Adecuación automática de cifras (Incluir los registros de Riesgos con problemas no fatales o globales y generar registros por las diferencias vs Analítico)
- Generación de Reportes

Con el sistema Responsabilidades de Cartera, se pretende obtener los siguientes beneficios:

- La disminución de la intervención manual por parte del usuario
- La eliminación al máximo de papel
- Reducción en los tiempos de entrega de la información
- Eliminación de registros que no se encuentran en el Maestro
- Eliminación de errores en la información que va a Banco de México
- Reducción en los tiempos de respuesta
- Eliminación de información no utilizada
- Enlaces con diversos sistemas
- Consulta en tiempo real
- Utilización de herramientas C.A.S.E.
- Eliminación de programas duplicados
- Eliminación de validaciones

Estrategia recomendada para la implantación S

Se iniciará con el módulo de interfases de entrada, dentro de misma fase inicial se incluirán los siguientes módulos:

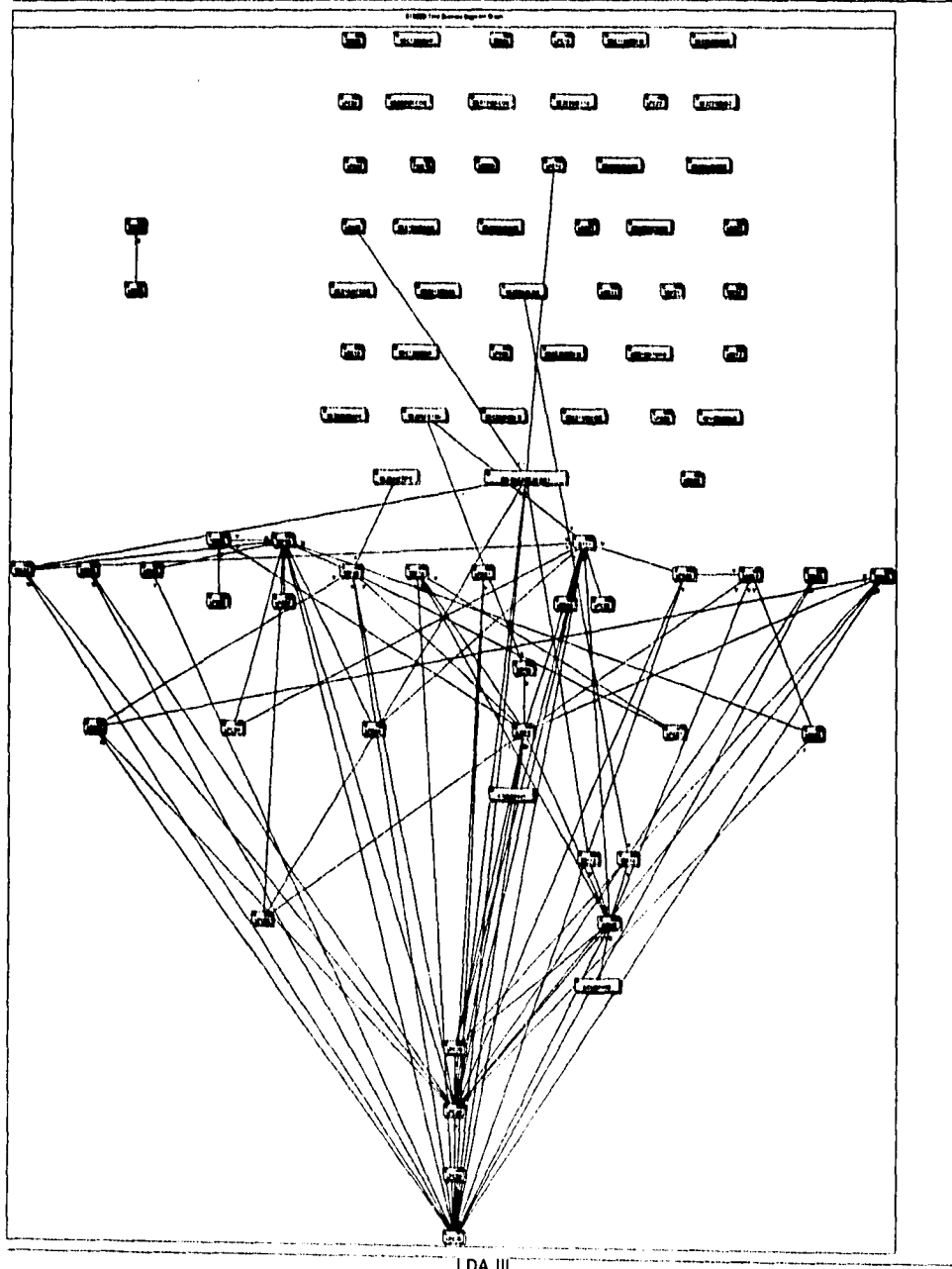
- Control de arribos
- Diagnostico de las altas
- Mantenimiento de catalogos
- Menú de cifras
- Adecuación de cifras manualmente
- Adecuación automática de saldos
- Modificaciones de la información de datos de clientes
- Información gerencial
- Información para Banco de México

- Generación de reportes y archivos

Que serán utilizados para tener un buen control de todas las transacciones que se realizan mensualmente.

Problemas críticos que se pueden dar

- Demasiada intervención manual del usuario
- Complementan la información con los sistemas operativos
- Se desconoce la validación de la información que hace banco de México
  - Gran cantidad de rechazos no propios del sistema debido a la inexistencia de la información en los sistemas operativos
- Excesivos consumos y desperdicios de papel
- Información incorrecta de los sistemas operativos (ejem: nombre)
- Utilización de estructuras propias codificadas internamente
- No utiliza catálogos generales, tienen tablas integradas a la programación
- Información no utilizada
- Discriminación impropia de movimientos de los sistemas operativos
- Inconsistencia en las cifras de la posición virtual vs. los totales por cuenta de las entidades contables
- Falta de información para procesos estadísticos
- Manejo de altos volúmenes de información impresa no utilizada
- Dificultades en el mantenimiento a los programas
- Difícil operación del sistema
  - Demasiada interacción entre el usuario y operación
- Falta de oportunidades en la información



LDA III

BANAMEX		BANCO NACIONAL DE MEXICO	
S1368D		RESPONSABILIDADES DE CARTERA	
S1368D-DEFAULT		ADECUAR SALDOS AUTOMATICAMENTE	
ADSDOATM		CDNS.DESCRIP.DE SALDOS EN CTAS	
LPC02	I-O	DEFAULT	Number 9 IMPORTE
IMPORTE		DEFAULT	Number 4 NO.CTA.BANAMEX
NUCTB		DEFAULT	Alpha 1 OPCION
OPCION		CONSULTA A TOTALES POR CUENTA	
LPC15	I-O	DEFAULT	Number 9 MONEDA EXTRANJER
ME		DEFAULT	Number 9 MONEDA NACIONAL
MN		DEFAULT	Number 4 NO.CTA.BANAMEX
NUCTB		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 8 TOTALES DE COLUM
TOTALES		MENU DE ADECUACION AUTOMATICA	
LPC17	I-O	DEFAULT	Alpha 1 OPCION
OPCIDN		CONS.DEL EDO.ANALITICO DE CTAS	
LPC31	I-O	DEFAULT	Number 12 CUENTA CONTABLE
CTACONT		DEFAULT	Number 4 NO.CTA.BANAMEX
NUCTB		DEFAULT	Number 2 NO.SECTOR
NUSEC		DEFAULT	Number 9 SALDO MON.EXT.
SDOME		DEFAULT	Number 9 SALDO MON.NAL
SDOMN		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 8 TOTALES DE COLUM
TOTALES		CONS.SECTORIZACION DE CARTERA	
LPC32	I-O	DEFAULT	Number 3 AGENCIA
AGENCIA		DEFAULT	Number 12 CUENTA CONTABLE
CTACONT		DEFAULT	Number 5 SECTOR 14
SEC14		DEFAULT	Number 5 SECTOR 22
SEC22		DEFAULT	Number 5 SECTOR 23
SEC23		DEFAULT	Number 5 SECTOR 26
SEC26		DEFAULT	Number 5 SECTOR31
SEC31		DEFAULT	Number 5 SECTOR 32
SEC32		DEFAULT	Number 5 SECTOR41
SEC41		DEFAULT	Number 5 SECTOR42
SEC42		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 8 TOTALES DE COLUM
TOTALES		CONS.RES.DE LA CLAS.DE CARTERA	
LPC33	I-O	DEFAULT	Number 6 ACTIVIDAD GENER.
ACTGEN		DEFAULT	Number 9 CARTERA REDESCON
CAREDESCON		DEFAULT	Number 9 CARTERA VENCIDA
CARVENCIDA		DEFAULT	Number 9 CARTERA VIGENTE
CARVIGENTE		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 8 TOTALES DE COLUM
TOTALES		CONSULTA A LA ACTUAL.DE ACTIV.	
LPC39	I-O	CONSULTA DIGITOS DE CONTROL	
LPC41	I-O	CONSULTA CIFRAS DE CONTROL	
LPC42	I-O	CONS.CLTES DE CART.VENCIDA	
LPC43	I-O	DEFAULT	Alpha 55 NOMBRE DE CLIEE
NOMCTE		OUTPUT	Number 4 NO.CTA.BANAMEX
NUCTB		DEFAULT	Number 4 NO.SUCURSAL
NUMSUC		DEFAULT	Number 9 SALDO ACTUAL
SDOACT		DEFAULT	Number 9 SALDO ANTERIOR
SDOANT		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 9 VARIACION/IMPORT
VARIACION		CONS.CLTES DE CART.REDESCONT	
LPC44	I-O	DEFAULT	Number 4 NO.CTA.BANAMEX
NUCTB		DEFAULT	Number 4 NOMCTE
NUMSUC		DEFAULT	Number 9 SALDO ACTUAL
SDOACT		DEFAULT	Number 9 SALDO ANTERIOR
SDOANT		DEFAULT	Number 9 TOTAL DE SALDOS
TOTAL		DEFAULT	Number 9 VARIACION/IMPORT
VARIACION		DEFAULT	Number 9

<ul style="list-style-type: none"> <li><input type="checkbox"/> TBT21 I-O</li> <li><input type="checkbox"/> TBT22 I-O             <ul style="list-style-type: none"> <li><input type="checkbox"/> ME</li> <li><input type="checkbox"/> MN</li> <li><input type="checkbox"/> NUCTB</li> <li><input type="checkbox"/> NUMSUC</li> <li><input type="checkbox"/> TOTAL</li> </ul> </li> <li><input type="checkbox"/> TBT23 I-O</li> <li><input checked="" type="checkbox"/> CATALOGOS             <ul style="list-style-type: none"> <li><input type="checkbox"/> LPC14 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> OPCION</li> </ul> </li> <li><input type="checkbox"/> TBT01 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMSUB</li> <li><input type="checkbox"/> NUMSUB</li> </ul> </li> <li><input type="checkbox"/> TBT02 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMCTA</li> <li><input type="checkbox"/> NUCTB</li> <li><input type="checkbox"/> NUMSUB</li> </ul> </li> <li><input type="checkbox"/> TBT04 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMEDO</li> <li><input type="checkbox"/> NUEDO</li> </ul> </li> <li><input type="checkbox"/> TBT05 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> LETRA</li> <li><input type="checkbox"/> NOMFONDO</li> <li><input type="checkbox"/> NUMFON</li> </ul> </li> <li><input type="checkbox"/> TBT08 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMSE</li> <li><input type="checkbox"/> NUSECO</li> </ul> </li> <li><input type="checkbox"/> TBT08 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> ACTGEN</li> <li><input type="checkbox"/> NOMACGEN</li> </ul> </li> <li><input type="checkbox"/> TBT09 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> ESTATUS</li> <li><input type="checkbox"/> NOMONEDA</li> <li><input type="checkbox"/> NUMON</li> <li><input type="checkbox"/> TICAMB</li> <li><input type="checkbox"/> TIMON</li> </ul> </li> <li><input type="checkbox"/> TBT10 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMSEC</li> <li><input type="checkbox"/> NUSEC</li> </ul> </li> <li><input type="checkbox"/> TBT11 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMSUC</li> <li><input type="checkbox"/> NUMSUC</li> </ul> </li> <li><input type="checkbox"/> TBT12 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> NOMBCA</li> <li><input type="checkbox"/> NUMBC</li> </ul> </li> <li><input type="checkbox"/> TBT14 I-O                 <ul style="list-style-type: none"> <li><input type="checkbox"/> CVENUM</li> <li><input type="checkbox"/> NOM-LOC-AC</li> <li><input type="checkbox"/> NUM-ACT-LO</li> <li><input type="checkbox"/> NUMREG</li> <li><input type="checkbox"/> VALIDA</li> </ul> </li> </ul> </li> <li><input checked="" type="checkbox"/> REZ24MTIPO             <ul style="list-style-type: none"> <li><input type="checkbox"/> REZ24MTIPO/00</li> <li><input type="checkbox"/> REZ24MTIPO/90                 <ul style="list-style-type: none"> <li><input type="checkbox"/> BLANCO</li> <li><input type="checkbox"/> FECANT</li> <li><input type="checkbox"/> FECPRO</li> <li><input type="checkbox"/> FECREAL</li> <li><input type="checkbox"/> HORA</li> <li><input type="checkbox"/> VERSION</li> </ul> </li> <li><input type="checkbox"/> REZ24MTIPO/91</li> </ul> </li> </ul>	<p>ARCHIVO DE GLOBAL/CTA-MONEDA</p> <p>ARCHIVO OE TOTAL/SUC-CTA-MONED</p> <p>DEFAULT Number 9 MONEDA EXTRANJER</p> <p>DEFAULT Number 8 MONEDA NACIONAL</p> <p>DEFAULT Number 4 NO.CTA.BANAMEX</p> <p>DEFAULT Number 4 NO.SUCURSAL</p> <p>DEFAULT Number 9 TOTAL DE SALDOS</p> <p>GLOBAL CLIENTE-MONEDA</p> <p>CARGA Y ACTUAL.DE CATALOGOS</p> <p>MENU DE CATALOGOS</p> <p>DEFAULT Alpha 1 OPCION</p> <p>CATALOGO DE SUBCUENTAS</p> <p>DEFAULT Alpha 30 NOMBRE SUBCUENTA</p> <p>DEFAULT Number 2 NUM. SUBCUENTA</p> <p>CATALOGO DE CUENTAS</p> <p>DEFAULT Alpha 30 NOMBRE DE CUENTA</p> <p>DEFAULT Number 4 NO.CTA.BANAMEX</p> <p>DEFAULT Number 2 SUBCUENTA</p> <p>CATALOGO DE LAS ENT.FEDER.</p> <p>DEFAULT Alpha 30 NOMBRE DEL EDO.</p> <p>DEFAULT Number 2 ND.ENT.FED.</p> <p>CATALOGO DE FONDOS</p> <p>DEFAULT Alpha 1 LETRA DEL FONDO</p> <p>DEFAULT Alpha 30 NOMBRE FONDO</p> <p>DEFAULT Number 2 NO.FONDO</p> <p>CATALOGO DE SECTOR ECONOMICO</p> <p>DEFAULT Alpha 30 NOMBRE SEC.ECO.</p> <p>DEFAULT Number 2 NO.SECTOR ECONOM</p> <p>CATALOGO DE ACTIVIDADES GENERI</p> <p>DEFAULT Number 6 ACTIVIDAD GENER.</p> <p>DEFAULT Alpha 30 NOM.ACT.GENERICA</p> <p>CATALOGO DE MONEDAS</p> <p>DEFAULT Alpha 1 NOMBRE MONEDA</p> <p>DEFAULT Alpha 30 NOMBRE MONEDA</p> <p>DEFAULT Number 2 NO.MONEDA</p> <p>DEFAULT Number 7 7 TIPO DE CAMBIO</p> <p>DEFAULT Number 1 TIPO DE MONEDA</p> <p>CATALOGO DE SECTORES BANKICO</p> <p>DEFAULT Alpha 30 NOMBRE DE SECTOR</p> <p>DEFAULT Number 2 NO.SECTOR</p> <p>CATALOGO DE SUCURSALES</p> <p>DEFAULT Alpha 30 NOMBRE DE SUCURS</p> <p>DEFAULT Number 4 NO.SUCURSAL</p> <p>CATALOGO DE BANCAS</p> <p>DEFAULT Alpha 13 NOM DE LA BANCA</p> <p>DEFAULT Number 2 NO. DE BANCA</p> <p>CAT.LOCALIDADES Y ACTIVIDADES</p> <p>DEFAULT Number 5 CVENUM</p> <p>DEFAULT Alpha 60 NOMBRE</p> <p>DEFAULT Number 8 NUMERO</p> <p>DEFAULT Number 2 NUM.DE REGISTRO</p> <p>DEFAULT Number 1 VALIDA</p> <p>EXTRACT DE TIPOS DE CAMBIO</p> <p>HEADER</p> <p>Blank 8</p> <p>Date 6</p> <p>Date 6</p> <p>Date 6</p> <p>Alpha 8</p> <p>Number 3</p> <p>DETALLE</p>
--	--



<input type="checkbox"/> BLANCO	Blank	6		
<input type="checkbox"/> CAMBIO	Number	7	7 SAS; TICAMB	
<input type="checkbox"/> MONEDA	Number	2	SAS; NUMON	
<input type="checkbox"/> NOMBRE	Alpha	30	SAS; NOMONEDA	
<input type="checkbox"/> NUM-CNB	Number	4		
<input checked="" type="checkbox"/> REZ25ESTRU			EXTRACT DE ESTRUCTURA S080	
<input type="checkbox"/> REZ25ESTRU/00				
<input type="checkbox"/> REZ25ESTRU/92				
<input checked="" type="checkbox"/> REZ26ACLOC			ACCESO	
<input type="checkbox"/> REZ26ACLOC/00			EXTRACT DE ACT. Y LOCALIDADES	
<input type="checkbox"/> REZ26ACLOC/93				
<input type="checkbox"/> ACTIVA	Number	1	ACCESO A LA TABLA DE LOC.Y ACT	
<input type="checkbox"/> BLANCO	Blank	8	SAS; VALIDA	
<input type="checkbox"/> CLAVE	Number	2	SAS; NUMREG	
<input type="checkbox"/> CVEBANAMEX	Number	5	SAS; T8T14.CVENUM	
<input type="checkbox"/> LOC-ACT	Number	8	SAS; NUM-ACT-LO	
<input type="checkbox"/> NOMBRE	Alpha	60	SAS; NOM-LOC-AC	
<input checked="" type="checkbox"/> CTESYRIESGOS			CLIENTES Y RIESGOS	
<input checked="" type="checkbox"/> ACLIE I-O			ARCHIVO DE CLIENTES	
<input type="checkbox"/> CVEAVAL	DEFAULT	Number	6	CLAVE DEL AVAL
<input type="checkbox"/> ESTATUS	DEFAULT	Number	1	STATUS
<input type="checkbox"/> FECHALTA	DEFAULT	Date	6	FECHA DE ALTA
<input type="checkbox"/> FECHAULT	DEFAULT	Date	6	FECHA ULT.MOV.
<input type="checkbox"/> GPDECO	DEFAULT	Number	2	GRUPO ECONOMICO
<input type="checkbox"/> NOERROR	DEFAULT	Alpha	2	NO. DE ERROR
<input type="checkbox"/> NOMCTE	DEFAULT	Alpha	55	NOM. DEL CLIENTE
<input type="checkbox"/> NUM-ACT-LO	DEFAULT	Number	8	NUM.ACT.LOC
<input type="checkbox"/> NUMBC	DEFAULT	Number	2	NO. DE BANCA
<input type="checkbox"/> NUMCSI	DEFAULT	Alpha	2	NO. DE C.S.I
<input type="checkbox"/> NUMCTE	DEFAULT	Number	8	NO.CLIENTE
<input type="checkbox"/> NUMREG	DEFAULT	Number	2	NUM.DE REG.
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> NUSEC	DEFAULT	Number	2	NO.SECTOR
<input type="checkbox"/> RFC	DEFAULT	Alpha	13	RFC
<input type="checkbox"/> SISORIGEN	DEFAULT	Number	2	SISITEMA ORIGEN
<input type="checkbox"/> TIPOVIV	DEFAULT	Number	2	TIPO DE VIVIENDA
<input checked="" type="checkbox"/> ARIES I-O			ARCHIVO DE RIESGOS	
<input type="checkbox"/> CTACONT	DEFAULT	Number	12	CUENTA CONTABLE
<input type="checkbox"/> ESTATUS	DEFAULT	Number	1	STATUS
<input type="checkbox"/> IMPORTE	DEFAULT	Number	9	IMPORTE
<input type="checkbox"/> NIVRIES	DEFAULT	Alpha	1	NIVEL DE RIESGO
<input type="checkbox"/> NOERROR	DEFAULT	Alpha	2	NO. DE ERROR
<input type="checkbox"/> NUEDO	DEFAULT	Number	2	NO. ESTADD
<input type="checkbox"/> NUMCSI	DEFAULT	Alpha	2	ND. DE C.S.I
<input type="checkbox"/> NUMCTE	DEFAULT	Number	8	NO.CLIENTE
<input type="checkbox"/> NUMFON	DEFAULT	Number	2	NO.FONDO
<input type="checkbox"/> NUMON	DEFAULT	Number	2	NO.MONEDA
<input type="checkbox"/> NUMRANGO	DEFAULT	Number	2	NO.RANGO IMPORTE
<input type="checkbox"/> NUMSUB	DEFAULT	Number	2	SUBCUENTA
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> NUSEC	DEFAULT	Number	2	NO.SECTOR
<input type="checkbox"/> RFC	DEFAULT	Alpha	13	RFC
<input type="checkbox"/> SIGNO	DEFAULT	Alpha	1	SIGNO DEL IMPORT
<input type="checkbox"/> SISORIGEN	DEFAULT	Number	2	SISITEMA ORIGEN
<input type="checkbox"/> TICARTE	DEFAULT	Number	1	TIPO DE CARTERA
<input type="checkbox"/> TIMON	DEFAULT	Number	1	TIPO DE MONEDA
<input type="checkbox"/> TIPOVIV	DEFAULT	Number	2	TIPO DE VIVIENDA
<input checked="" type="checkbox"/> LPC03 I-O			CONSULTA DE RIESGOS C/ERROR	
<input type="checkbox"/> CTACONT	DEFAULT	Number	12	CUENTA CONTABLE
<input type="checkbox"/> MAINT	DEFAULT	Alpha	3	Maint Type
<input type="checkbox"/> NUMCTE	DEFAULT	Number	8	NO.CLIENTE
<input type="checkbox"/> NUMFON	DEFAULT	Number	2	NO.FONDO

<input type="checkbox"/> NUMSUC	OEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> OPCION	OEFAULT	Alpha	1	OPCION
<input type="checkbox"/> SDOACT	OEFAULT	Number	9	SALDO ACTUAL
<input type="checkbox"/> SUBCTA	OEFAULT	Number	12	CUENTA DESGLOSE
<input type="checkbox"/> TIMON	OEFAULT	Number	1	TIPO OE MONEDA
<input checked="" type="checkbox"/> LPC05 I-O				CONSULTA DE CTES. C/ERRORES
<input type="checkbox"/> NOMCTE	OEFAULT	Alpha	55	NOMBRE OEL CTE
<input type="checkbox"/> NUACBA	OEFAULT	Number	5	NO.ACT.BANAMEX
<input type="checkbox"/> NUACBM	OEFAULT	Number	8	NO.ACT.BAN. MEX.
<input type="checkbox"/> NUCTB	OEFAULT	Number	4	NO.CTA.BANAMEX
<input type="checkbox"/> NULOBA	OEFAULT	Number	5	NO.LOC.BANAMEX
<input type="checkbox"/> NULOBM	OEFAULT	Number	8	NO.LOC.BAN.MEX.
<input type="checkbox"/> NUMCTE	OEFAULT	Number	8	NO.CLIENTE
<input type="checkbox"/> NUMFON	OEFAULT	Number	2	NO.FONOO
<input type="checkbox"/> NUMON	OEFAULT	Number	2	NO.MONEDA
<input type="checkbox"/> NUMSUC	OEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> NUSEC	OEFAULT	Number	2	NO.SECTOR
<input type="checkbox"/> OPCION	OEFAULT	Alpha	1	OPCION
<input type="checkbox"/> RFC	OEFAULT	Alpha	13	RFC
<input checked="" type="checkbox"/> TABER I-O				TABLA DE ERRORES
<input type="checkbox"/> NOERROR	OEFAULT	Alpha	2	NO. OE ERROR
<input type="checkbox"/> NOMERR	OEFAULT	Alpha	25	NOMBRE ERROR
<input checked="" type="checkbox"/> ACLIEPF03				PROFILE DE CLIENTES
<input type="checkbox"/> NUMSUC			1 Ascending	NUMERO OE SUCURSAL
<input type="checkbox"/> NUMCTE			2 Ascending	
<input checked="" type="checkbox"/> ARIESPF01				PROFILE DE RIESGOS
<input type="checkbox"/> NUMSUC			1 Ascending	NUMERO DE SUCURSAL
<input type="checkbox"/> NUMCTE			2 Ascending	
<input checked="" type="checkbox"/> CTRARRIBOS				CONTROL DE ARRIBOS
<input checked="" type="checkbox"/> LPC06 I-O				CONSULTA A LAS CIFRAS XFER
<input type="checkbox"/> FECHARRIBO	OEFAULT	Date	6	FECHA OE ARRIBO
<input type="checkbox"/> INTEGRA	OEFAULT	Date	6	FECHA DE INTEGRA
<input type="checkbox"/> NOMCSI	OEFAULT	Alpha	15	NOMBRE OE C.S.I
<input type="checkbox"/> OPCION	OEFAULT	Alpha	1	OPCION
<input type="checkbox"/> SISARRIB	OEFAULT	Number	2	SISTEMA ARRIBADO
<input type="checkbox"/> TOTALIMP	OEFAULT	Number	10	TOTAL DE IMPORTE
<input type="checkbox"/> TOTALREG	OEFAULT	Number	10	TOT REG.ARRIBADO
<input checked="" type="checkbox"/> LPC11 I-O				MENU CDNTROL DE ARRIBOS
<input type="checkbox"/> OPCION	OEFAULT	Alpha	1	OPCION
<input checked="" type="checkbox"/> REZ20MS400				EXTRACT DEL S400,S404,PC
<input type="checkbox"/> REZ20MS400/00				HEADER DEL S400,S404,PC
<input type="checkbox"/> REZ20MS400/76				
<input type="checkbox"/> B1		Blank	18	
<input type="checkbox"/> BCO		Blank	14	
<input type="checkbox"/> BLANCO		Alpha	132	
<input type="checkbox"/> CSI		Alpha	2	SAS; NUMCSI
<input type="checkbox"/> NUMCOR		Number	2	
<input type="checkbox"/> PROCESO		Date	6	
<input checked="" type="checkbox"/> REZ20MS400/77				DETALLE DEL S400,S404,PC
<input type="checkbox"/> ACTIVIDAD		Number	5	SAS; NUACBA
<input type="checkbox"/> AVAL		Number	6	SAS; CVEAVAL
<input type="checkbox"/> BANCA		Number	2	SAS; NUMBC
<input type="checkbox"/> BANXICO		Number	13	
<input type="checkbox"/> BLANCO		Alpha	21	
<input type="checkbox"/> CSI		Alpha	2	SAS; NUMCSI
<input type="checkbox"/> CVECTE		Number	2	
<input type="checkbox"/> GRUPOEC		Number	2	SAS; GPOECO
<input type="checkbox"/> LOCALIDAD		Number	5	SAS; NULOBA
<input type="checkbox"/> NIVRIES		Alpha	1	SAS; NIVRIES
<input type="checkbox"/> NOMBRE		Alpha	55	SAS; NOMCTE
<input type="checkbox"/> REGFED		Alpha	13	SAS; RFC
<input type="checkbox"/> SECTOR		Number	2	SAS; NUSEC

<input type="checkbox"/> SITOR	Number	2	SAS; SITORIGEN
<input type="checkbox"/> SUCURSAL	Number	4	SAS; NUMSUC
<input type="checkbox"/> VIVIENDA	Number	2	SAS; TIPOVIV
<input checked="" type="checkbox"/> REZ20MS400/78		DETALLE 2	
<input type="checkbox"/> BANXICO	Alpha	13	
<input type="checkbox"/> BCO	Alpha	1	
<input type="checkbox"/> BLANCO	Alpha	8	
<input type="checkbox"/> CLAVE	Number	2	
<input type="checkbox"/> CUENTA	Number	12	SAS; CTACONT
<input type="checkbox"/> FONDO	Number	2	SAS; NUMFON
<input type="checkbox"/> MON	Number	2	SAS; NUMON
<input type="checkbox"/> SDO	Number	12	SAS; SALDO
<input type="checkbox"/> SUBCUENTA	Number	2	
<input type="checkbox"/> SUCURSAL	Number	4	SAS; NUMSUC
<input checked="" type="checkbox"/> REZ20MS400/78		TRAILER DEL S400,S404,PC	
<input type="checkbox"/> BCO	Alpha	122	
<input type="checkbox"/> BLANCO	Alpha	132	
<input type="checkbox"/> TOTALSDOS	Number	18	
<input type="checkbox"/> TOTALSUC	Number	4	
<input type="checkbox"/> TOTREG	Number	10	SAS; TOTALREG
<input type="checkbox"/> TOTSDOS	Number	8	SAS; TOTALES
<input checked="" type="checkbox"/> REZ21MS600		EXTRACT DEL S600	
<input type="checkbox"/> REZ21MS600/00			
<input checked="" type="checkbox"/> REZ21MS600/80		HEADER	
<input type="checkbox"/> CSI	Alpha	2	SAS; NUMCSI
<input type="checkbox"/> FPRO	Date	6	
<input type="checkbox"/> NUMCOR	Number	2	
<input checked="" type="checkbox"/> REZ21MS600/81		DETALLE ALTAS	
<input type="checkbox"/> ACTGENER	Number	6	SAS; ACTGEN
<input type="checkbox"/> ACTIVIDAD	Number	5	SAS; NUACBA
<input type="checkbox"/> AVAL	Number	6	SAS; CVEAVAL
<input type="checkbox"/> BANCA	Number	2	SAS; NUMBC
<input type="checkbox"/> BANXICO	Alpha	13	
<input type="checkbox"/> CLAVE	Number	2	
<input type="checkbox"/> GRUPOECO	Number	2	SAS; GPOECO
<input type="checkbox"/> LOCALIDAD	Number	5	SAS; NULOBA
<input type="checkbox"/> NOMBRE	Alpha	55	SAS; NOMCTE
<input type="checkbox"/> REGFED	Alpha	13	SAS; RFC
<input type="checkbox"/> SECTOR	Number	2	SAS; NUSEC
<input type="checkbox"/> SISTEMA	Number	1	
<input type="checkbox"/> SUCURSAL	Number	4	SAS; NUMSUC
<input type="checkbox"/> TIPERSONA	Number	1	SAS; TIPOPER
<input type="checkbox"/> TIPORIESGO	Alpha	1	SAS; NIVRIES
<input type="checkbox"/> VIVIENDA	Number	2	SAS; TIPOVIV
<input checked="" type="checkbox"/> REZ21MS600/82		DETALLE2 RIESGOS	
<input type="checkbox"/> BANXICO	Alpha	13	
<input type="checkbox"/> CONVE	Number	2	
<input type="checkbox"/> CUENTAC	Number	12	SAS; CTACONT
<input type="checkbox"/> CUENTAN	Number	12	
<input type="checkbox"/> CVETRA	Number	2	
<input type="checkbox"/> FDNDO	Number	2	SAS; NUMFON
<input type="checkbox"/> MON	Number	2	SAS; NUMON
<input type="checkbox"/> SDO	Number	12	SAS; SALDO
<input type="checkbox"/> SIG	Alpha	1	SAS; SIGNO
<input type="checkbox"/> SUBCUENTA	Number	2	SAS; SUBCUENTA
<input type="checkbox"/> SUCURSAL	Number	4	SAS; NUMSUC
<input checked="" type="checkbox"/> REZ21MS600/83		TRAILER	
<input type="checkbox"/> TOTREG	Number	6	
<input type="checkbox"/> TOTSDOS	\$	14	2
<input type="checkbox"/> TOTSDOSIS	\$	18	
<input type="checkbox"/> TOTSUC	\$	4	
<input checked="" type="checkbox"/> REZ22MS052		EXTRACT DEL S052	

<input type="checkbox"/> REZ22MS052/00			
<input type="checkbox"/> REZ22MS052/84		HEADER S052	
<input type="checkbox"/> BLANC		Blank	18
<input type="checkbox"/> BLANCO		Blank	18
<input type="checkbox"/> BLANCOS		Blank	10
<input type="checkbox"/> CVE-REG		Alpha	2
<input type="checkbox"/> FECONT		Date	6
<input type="checkbox"/> FER		Date	6
<input type="checkbox"/> HOR		Alpha	8
<input type="checkbox"/> REZ22MS052/85		DETALLE S052	
<input type="checkbox"/> BLANC		Blank	9
<input type="checkbox"/> BLANCO		Blank	2
<input type="checkbox"/> BLANCOS		Blank	18
<input type="checkbox"/> CLAVE		Alpha	2
<input type="checkbox"/> CTAYSUB		Number	6
<input type="checkbox"/> MONEDAS		Number	2
<input type="checkbox"/> SDO-CTA		Number	4
<input type="checkbox"/> SDO-SDO		Number	18
<input type="checkbox"/> SDOCONTREG		Number	4
<input type="checkbox"/> SUCURSAL		Number	4
<input type="checkbox"/> REZ22MS052/86		TRAILER S052	
<input type="checkbox"/> BLANCO		Blank	18
<input type="checkbox"/> BLANCOS		Blank	18
<input type="checkbox"/> CLAVE		Alpha	2
<input type="checkbox"/> NOREG		Number	8
<input type="checkbox"/> SUMA		Number	18
<input type="checkbox"/> REZ23MS114		EXTRACT DEL S114	
<input type="checkbox"/> REZ23MS114/00			
<input type="checkbox"/> REZ23MS114/87		HEADER DEL S114	
<input type="checkbox"/> BLANCO		Alpha	11
<input type="checkbox"/> FECHA-SDOS		Date	6
<input type="checkbox"/> REZ23MS114/88		DETALLE DEL S114	
<input type="checkbox"/> BLANCO		Number	1
<input type="checkbox"/> CTA-070-6		Number	6
<input type="checkbox"/> CTA-RES		Number	4
<input type="checkbox"/> DEPTO		Number	1
<input type="checkbox"/> DIGITO		Number	4
<input type="checkbox"/> ESP		Number	6
<input type="checkbox"/> ESPAC		Alpha	86
<input type="checkbox"/> ESPACIO		Alpha	115
<input type="checkbox"/> FECULTMOV		Date	6
<input type="checkbox"/> SDO-070		Number	14
<input type="checkbox"/> SUCURSAL		Number	4
<input type="checkbox"/> REZ23MS114/89		TRAILER DEL S114	
<input type="checkbox"/> CLAVE		Number	2
<input type="checkbox"/> ENTMAN		ENTIDADES MANUALES(PC)	
<input type="checkbox"/> S400		SISTEMA 400	
<input type="checkbox"/> S404		SISTEMA 404	
<input type="checkbox"/> S600		SISTEMA 600	
<input type="checkbox"/> GENREPORTES		GENERACION DE REPORTES	
<input type="checkbox"/> LPC21 I-O		MENU DE REPORTES DE CATALOGOS	
<input type="checkbox"/> LPC22 I-O		MENU DE REPORTES DE INFORMACIO	
<input type="checkbox"/> LPC24 I-O		MENU GRAL.DE REPORTES	
<input type="checkbox"/> TBT24 I-D		CINTA PARA CIG	
<input type="checkbox"/> TBT25 I-O		CINTA DE MENDRES PARA B.M.	
<input type="checkbox"/> TBT27 I-O		CINTA DE MAYORES PARA B.M.	
<input type="checkbox"/> REC11MRBAM		REPORTE DE TOTALES POR CUENTA	
<input type="checkbox"/> REC11MRBAM/00			
<input type="checkbox"/> REC11MRBAM/11		ENCABEZADO	
<input type="checkbox"/> PAGINA		Number	2
<input type="checkbox"/> REC11MRBAM/5B		TOTALES POR CUENTA	
<input type="checkbox"/> CUENTA		Number	12
		SAS; CTACONT	

- MONAL
- MONEXT
- TOTA
- REC11MRBAM/59
- REC12MROFI
- REC12MROFI/00
- REC12MROFI/12
- PAGINA
- REC12MROFI/30
- NUACTION
- NUENTFED
- REC12MROFI/59
- TOTALGL
- TOTPROPIA
- TOTVAL
- REC13MROFI
- REC13MROFI/00
- REC13MROFI/13
- PAGINA
- REC13MROFI/31
- ENTFFED
- NUACTION
- REC13MROFI/60
- TOTALGL
- TOTCAREDES
- TOTVAL
- REC22MRCOG
- REC22MRCOG/00
- REC22MRCOG/22
- PAGINA
- REC22MRCOG/33
- AVAL
- BANCA
- CALIFICA
- ERRONEO
- MON
- NOMBRECTE
- NUMACTIV
- NUMCUENTA
- NUMLOC
- REGFED
- SECTOR
- SUBCUENTA
- SUCURSAL
- REC22MRCOG/64
- MON-EXT
- MON-NAL
- NUMCTA
- TOT
- TOT-SUMA
- REC23MCCOG
- REC23MCCOG/00
- REC23MCCOG/23
- PAGINA
- REC23MCCOG/34
- ACTIVIDAD
- AVAL
- BANCA
- CALIFICA
- ERRONEO
- FONDO
- LOCALIDAD

Number	9	SAS; MN
Number	9	SAS; ME
Number	9	SAS; TOTAL
TOTALES		
REPORTE DIST.DE CARTERA PROPIA		
ENCABEZADO		
Number	2	
DETALLE CARTERA PROPIA		
Number	8	SAS; NUACBM
Number	2	SAS; NUEDO
TOTALES DE LA CARTERA		
Number	8	SAS; TOTALES
Number	12	
Number	12	
REPORTE DIST.CART REDESCONT		
ENCABEZADO		
Number	2	
DETALLE DE CART.REDESCONTADA		
Number	2	SAS; NUEDO
Number	8	SAS; NUACBM
TOTALES DE LA CARTERA		
Number	8	SAS; TOTALES
Number	12	
Number	12	
REPORTE DE RECHAZOS POR CTE		
ENCABEZADO		
Number	2	
DETALLE DE LOS RECHAZOS		
Number	6	SAS; CVEAVAL
Number	2	SAS; NUMBC
Alpha	1	SAS; NIVRIES
Alpha	2	SAS; NOERROR
Number	1	SAS; TIMDN
Alpha	55	SAS; NOMCTE
Number	5	SAS; NUACBA
Number	4	SAS; NUCTB
Number	5	SAS; NULOBA
Alpha	13	SAS; RFC
Number	2	SAS; NUSEC
Number	2	
Number	4	SAS; NUMSUC
CARATULA DE CONTROL		
Number	9	SAS; ME
Number	9	SAS; MN
Number	4	SAS; NUCTB
Number	9	SAS; TOTAL
Number	8	SAS; TOTALES
REPORTE DE MODIFICACIONES		
ENCABEZADO		
Number	2	
DETALLE		
Number	5	SAS; NUACBA
Number	6	SAS; CVEAVAL
Number	2	SAS; NUMBC
Alpha	1	SAS; NIVRIES
Alpha	2	SAS; NOERROR
Number	2	SAS; NUMFON
Number	5	SAS; NULOBA

<input type="checkbox"/> MON	Number	1	SAS; TIMON
<input type="checkbox"/> NOMBRE-CTE	Alpha	55	SAS; NOMCTE
<input type="checkbox"/> NUMCTA	Number	4	SAS; NUCTB
<input type="checkbox"/> REG-FED	Alpha	13	SAS; RFC
<input type="checkbox"/> SDO	Number	12	SAS; SALOO
<input type="checkbox"/> SECTOR	Number	2	SAS; NUSEC
<input type="checkbox"/> SUBCUENTA	Number	2	
<input type="checkbox"/> SUCURSAL	Number	4	SAS; NUMSUC
<input type="checkbox"/> REC23MCCOG			
<input type="checkbox"/> MON-EXT	Number	9	SAS; ME
<input type="checkbox"/> MON-NAL	Number	9	SAS; MN
<input type="checkbox"/> NUMCTA	Number	4	SAS; NUCTB
<input type="checkbox"/> TOT	Number	9	SAS; TOTAL
<input type="checkbox"/> TOTA	Number	8	SAS; TOTALES
<input type="checkbox"/> REC25MCCOG			
<input type="checkbox"/> REC25MCCOG/00			
<input type="checkbox"/> REC25MCCOG/25			
<input type="checkbox"/> PAGINA	Number	2	
<input type="checkbox"/> REC25MCCOG/67			
<input type="checkbox"/> REC26MCBAM			
<input type="checkbox"/> REC26MCBAM/00			
<input type="checkbox"/> REC26MCBAM/26			
<input type="checkbox"/> REC26MCBAM/36			
<input type="checkbox"/> LOCALIDAD	Alpha	25	SAS; NOMLOC
<input type="checkbox"/> NUMONEDA	Number	1	SAS; TIMON
<input type="checkbox"/> REC26MCBAM/67			
<input type="checkbox"/> SDOCREDITO	Number	9	SAS; SDOTIPCRE
<input type="checkbox"/> TITDESC	Number	9	SAS; SDO-6101
<input type="checkbox"/> REC27MCBAM			
<input type="checkbox"/> REC27MCBAM/00			
<input type="checkbox"/> REC27MCBAM/27			
<input type="checkbox"/> PAGINA	Number	2	
<input type="checkbox"/> REC27MCBAM/37			
<input type="checkbox"/> ACTIVIDAD	Number	8	SAS; NUACBM
<input type="checkbox"/> AVALISTA	Number	6	SAS; CVEAVAL
<input type="checkbox"/> CALIF	Alpha	1	SAS; NIVRIES
<input type="checkbox"/> LOCALIDAD	Number	8	SAS; NULOBM
<input type="checkbox"/> MON	Number	1	SAS; TIMON
<input type="checkbox"/> NOMLOCAL	Alpha	25	SAS; NOMLOC
<input type="checkbox"/> REGFED	Alpha	13	SAS; RFC
<input type="checkbox"/> SECTOR	Number	2	SAS; NUSEC
<input type="checkbox"/> REC27MCBAM/68			
<input type="checkbox"/> DES-CONT	Alpha	1	SAS; DES6102
<input type="checkbox"/> DESCONT	Alpha	1	SAS; DES6101
<input type="checkbox"/> MONAL	Number	9	SAS; MN
<input type="checkbox"/> MONEXT	Number	9	SAS; ME
<input type="checkbox"/> SUMASDO	Number	9	
<input type="checkbox"/> TIPCREDITO	Number	9	SAS; SDOTIPCRE
<input type="checkbox"/> TITDESC	Number	9	SAS; SDO-6101
<input type="checkbox"/> TITDESCONT	Number	9	SAS; SDO-6102
<input type="checkbox"/> TOT	Number	9	SAS; TOTAL
<input type="checkbox"/> REC28MCCOG			
<input type="checkbox"/> REC28MCCOG/00			
<input type="checkbox"/> REC28MCCOG/2B			
<input type="checkbox"/> PAGINA	Number	2	
<input type="checkbox"/> REC28MCCOG/38			
<input type="checkbox"/> ACTIVIDAD	Number	5	
<input type="checkbox"/> BANCA	Number	2	SAS; NUMBC
<input type="checkbox"/> CALIFICA	Alpha	1	SAS; NIVRIES
<input type="checkbox"/> CUENTA	Number	4	
<input type="checkbox"/> FONDO	Number	2	
<input type="checkbox"/> LOCALIDAD	Number	5	

CONTROL  
 Number 9 SAS; ME  
 Number 9 SAS; MN  
 Number 4 SAS; NUCTB  
 Number 9 SAS; TOTAL  
 Number 8 SAS; TOTALES

COMPARATIVO POR CUENTA

ENCABEZADO  
 Number 2  
 TOTALES POR CUENTA  
 REPORTE DE PRESTATARIOS MENORE

ENCABEZADO  
 DETALLE DE MENORES  
 Alpha 25 SAS; NOMLOC  
 Number 1 SAS; TIMON  
 SALDOS  
 Number 9 SAS; SDOTIPCRE  
 Number 9 SAS; SDO-6101  
 REPORTE PRESTATARIOS MAYOR

ENCABEZADO  
 Number 2  
 DETALLE DE MAYORES  
 Number 8 SAS; NUACBM  
 Number 6 SAS; CVEAVAL  
 Alpha 1 SAS; NIVRIES  
 Number 8 SAS; NULOBM  
 Number 1 SAS; TIMON  
 Alpha 25 SAS; NOMLOC  
 Alpha 13 SAS; RFC  
 Number 2 SAS; NUSEC  
 SALDOS  
 Alpha 1 SAS; DES6102  
 Alpha 1 SAS; DES6101  
 Number 9 SAS; MN  
 Number 9 SAS; ME  
 Number 9  
 Number 9 SAS; SDOTIPCRE  
 Number 9 SAS; SDO-6101  
 Number 9 SAS; SDO-6102  
 Number 9 SAS; TOTAL  
 REPORTE DE SALDOS POR CLTE

ENCABEZADO  
 Number 2  
 DETALLE  
 Number 5  
 Number 2 SAS; NUMBC  
 Alpha 1 SAS; NIVRIES  
 Number 4  
 Number 2  
 Number 5

<input type="checkbox"/> MON	Number	1	
<input type="checkbox"/> NOMBRE	Alpha	25	
<input type="checkbox"/> REGFED	Alpha	13	
<input type="checkbox"/> SDO	Number	9	
<input type="checkbox"/> SECTOR	Number	2	SAS; NUSEC
<input type="checkbox"/> SUBCUENTA	Alpha	2	
<input type="checkbox"/> SUCURSAL	Number	4	
<input checked="" type="checkbox"/> REC29MCCOG	<b>REPORTE ALFABETICO/SECTOR</b>		
<input type="checkbox"/> REC29MCCOG/00	ENCABEZADO		
<input type="checkbox"/> REC29MCCOG/29	Number	2	
<input type="checkbox"/> PAGINA	DETALLE DEL ALFABETICO		
<input type="checkbox"/> REC29MCCOG/39	Number	8	SAS; NUACBM
<input type="checkbox"/> ACTIVIDAD	Number	6	SAS; CVEAVAL
<input type="checkbox"/> AVAL	Alpha	1	SAS; NIVRIES
<input type="checkbox"/> CALIF	Number	12	SAS; CTACONT
<input type="checkbox"/> CUENTA	Number	8	SAS; NULOBM
<input type="checkbox"/> LOCALIDAD	Alpha	55	SAS; NOMCTE
<input type="checkbox"/> NOMBRE	Number	8	SAS; NUMCTE
<input type="checkbox"/> NUMCLIENTE	Number	1	SAS; TIMON
<input type="checkbox"/> NUMONEDA	Alpha	13	SAS; RFC
<input type="checkbox"/> REGFED	Number	12	SAS; SALDO
<input type="checkbox"/> SDO	Number	2	SAS; NUSEC
<input type="checkbox"/> SECTOR	Number	4	SAS; NUMSUC
<input type="checkbox"/> SUCURSAL	<b>REPORTE DE AUOITORIA CINTAS</b>		
<input checked="" type="checkbox"/> REC30MCCOG	ENCABEZADO		
<input type="checkbox"/> REC30MCCOG/00	Number	2	
<input type="checkbox"/> REC30MCCOG/30	DETALLE DE LA AUDITORIA		
<input type="checkbox"/> PAGINA	Number	8	SAS; NUACBM
<input type="checkbox"/> REC30MCCOG/40	Number	6	SAS; CVEAVAL
<input type="checkbox"/> ACTIVIDAD	Date	6	SAS; FECHA
<input type="checkbox"/> AVAL	Number	5	
<input type="checkbox"/> FECPROC	Number	8	SAS; NULOBM
<input type="checkbox"/> INSTITU	Number	2	
<input type="checkbox"/> LOCALIDAD	Alpha	55	SAS; NOMCTE
<input type="checkbox"/> NIVELCR	Number	2	SAS; NUSEC
<input type="checkbox"/> NOMBRE	Number	1	SAS; TIPOPER
<input type="checkbox"/> SECTOR	Number	1	SAS; TIMON
<input type="checkbox"/> TIPERSONA	Number	9	SAS; TOTAL
<input type="checkbox"/> TIPOMONEDA	<b>REPORTE DE ALTAS DE CLIENTES</b>		
<input type="checkbox"/> TOTGL	ENCABEZADO		
<input checked="" type="checkbox"/> REC31MCGOG	Number	2	
<input type="checkbox"/> REC31MCGOG/00	DETALLE		
<input type="checkbox"/> REC31MCGOG/31	Number	8	SAS; NUACBM
<input type="checkbox"/> PAGINA	Number	5	SAS; NUACBA
<input type="checkbox"/> REC31MCGOG/41	Number	6	SAS; CVEAVAL
<input type="checkbox"/> ACTBM	Number	2	SAS; NUMBC
<input type="checkbox"/> ACTIVIDAD	Number	8	SAS; NUMCTE
<input type="checkbox"/> AVAL	Number	2	SAS; GPOECO
<input type="checkbox"/> BANCA	Number	5	SAS; NULOBA
<input type="checkbox"/> CLIENTE	Number	8	SAS; NULOBM
<input type="checkbox"/> GPOECONOM	Alpha	55	SAS; NOMCTE
<input type="checkbox"/> LOCALIDAD	Alpha	2	SAS; NOERROR
<input type="checkbox"/> LOCBM	Alpha	13	SAS; RFC
<input type="checkbox"/> NOMBRE	Number	2	SAS; NUSEC
<input type="checkbox"/> NUMERROR	Number	4	SAS; NUMSUC
<input type="checkbox"/> REGFED	<b>REPORTE DE LOS SEC.ECONOMIC</b>		
<input type="checkbox"/> SECTOR	ENCABEZADO		
<input type="checkbox"/> SUCURSAL	Number	2	
<input checked="" type="checkbox"/> RER10MTCOG	ENCABEZADO		
<input type="checkbox"/> RER10MTCOG/00	Number	2	
<input type="checkbox"/> RER10MTCOG/10	ENCABEZADO		
<input type="checkbox"/> PAGINA	Number	2	

<ul style="list-style-type: none"> <li><input type="checkbox"/> RER10MTCOG/27</li> <li><input checked="" type="checkbox"/> RER15MTCOG             <ul style="list-style-type: none"> <li><input type="checkbox"/> RER15MTCOG/00</li> <li><input type="checkbox"/> RER15MTCOG/15</li> <li><input type="checkbox"/> PAGINA</li> <li><input type="checkbox"/> RER15MTCOG/28</li> <li><input type="checkbox"/> RER15MTCOG/62</li> </ul> </li> <li><input checked="" type="checkbox"/> RER18MRCOG             <ul style="list-style-type: none"> <li><input type="checkbox"/> RER18MRCOG/00</li> <li><input type="checkbox"/> RER18MRCOG/18</li> <li><input type="checkbox"/> PAGINA</li> <li><input type="checkbox"/> RER18MRCOG/31</li> <li><input type="checkbox"/> RER18MRCOG/65</li> </ul> </li> <li><input checked="" type="checkbox"/> RER19MACOG             <ul style="list-style-type: none"> <li><input type="checkbox"/> RER19MACOG/00</li> <li><input type="checkbox"/> RER19MACOG/19</li> <li><input type="checkbox"/> PAGINA</li> <li><input type="checkbox"/> RER19MACOG/32</li> <li><input type="checkbox"/> CUENTA</li> <li><input type="checkbox"/> SISTEMA</li> <li><input type="checkbox"/> RER19MACOG/66</li> <li><input type="checkbox"/> MONAL</li> <li><input type="checkbox"/> MONEXT</li> </ul> </li> <li><input checked="" type="checkbox"/> REV02MGCIG             <ul style="list-style-type: none"> <li><input type="checkbox"/> REV02MGCIG/00</li> <li><input type="checkbox"/> REV02MGCIG/02</li> <li><input type="checkbox"/> PAGINA</li> <li><input type="checkbox"/> REV02MGCIG/21</li> <li><input type="checkbox"/> ACTGENER</li> <li><input checked="" type="checkbox"/> REV02MGCIG/50                 <ul style="list-style-type: none"> <li><input type="checkbox"/> CTERED</li> <li><input type="checkbox"/> CTEVE</li> <li><input type="checkbox"/> CTEVIG</li> <li><input type="checkbox"/> MONRE</li> <li><input type="checkbox"/> MONVEN</li> <li><input type="checkbox"/> MONVIG</li> <li><input type="checkbox"/> PORCTEVI</li> <li><input type="checkbox"/> PORCTVEN</li> <li><input type="checkbox"/> PORE</li> <li><input type="checkbox"/> PORED</li> <li><input type="checkbox"/> PORVEN</li> <li><input type="checkbox"/> PORVIG</li> </ul> </li> <li><input type="checkbox"/> REV02MGCIG/51</li> <li><input type="checkbox"/> CUENTA</li> <li><input type="checkbox"/> MONTO</li> <li><input type="checkbox"/> PORCENTAJE</li> <li><input type="checkbox"/> TOTCTES</li> </ul> </li> <li><input checked="" type="checkbox"/> MODIFCTES             <ul style="list-style-type: none"> <li><input type="checkbox"/> LPC16 I-O</li> <li><input type="checkbox"/> OPCION</li> </ul> </li> <li><input checked="" type="checkbox"/> PRUEBA             <ul style="list-style-type: none"> <li><input type="checkbox"/> PANT9 I-D</li> <li><input type="checkbox"/> NOMALGO</li> </ul> </li> <li><input checked="" type="checkbox"/> TOTALESR             <ul style="list-style-type: none"> <li><input type="checkbox"/> LPC25 I-O</li> <li><input type="checkbox"/> LPC26 I-O</li> <li><input type="checkbox"/> IMPORTE</li> <li><input type="checkbox"/> NOMFONDO</li> <li><input type="checkbox"/> NUMFON</li> <li><input type="checkbox"/> NUMON</li> <li><input type="checkbox"/> LPC27 I-O</li> <li><input type="checkbox"/> ME-MAY</li> </ul> </li> </ul>	<p>DETALLE DE LOS SECTORES              REPORTE DE CARAT.DE CARTERA</p> <p>ENCABEZADO              Number 2</p> <p>DETALLE DE LA CAR/CARTERA              IMPORTE DE LA CARATULA DE CART              REPORTE DE SALDOS MAYORES</p> <p>ENCABEZADO              Number 2</p> <p>DETALLE DE LAS CUENTAS              TOTALES DE LA CUENTA              REPORTE DE TOTALES POR CTA.</p> <p>ENCABEZADO              Number 2</p> <p>DETALLE              Number 12 SAS: CTACONT              Number 2 SAS: SISARRIB</p> <p>TOTALES POR CUENTA              Number 9 SAS: MN              Number 9 SAS: ME</p> <p>REPORTE DE CLAS.DE CARTERA</p> <p>ENCABEZADO              Number 2</p> <p>DETALLE              Number 6 SAS: ACTGEN</p> <p>CORTES DE LA CLAS.DE CARTERA              Number 4              Number 4              Number 4              Number 4              Number 6              Alpha 6              Number 4              Alpha 2              Number 3              Number 4              Alpha 2              Alpha 2</p> <p>TOTALES POR CUENTA              Number 12 SAS: CTACONT              Number 8 SAS: TOTALES              Number 2 2              Number 6</p> <p>MODIF.DE CLIENTES Y RIESGOS              MENU DE MODIFICACIONES              DEFAULT Alpha 1 OPCION</p> <p>pantalla de prueba              DEFAULT Alpha 30 normalgo</p> <p>TOTALES DE RIESGOS              CONS. ARCH.SDO.CART.VENC.Y VIG.              CONS.CIFRAS DE CONTROL/FONDOS              DEFAULT Number 9 IMPORTE              DEFAULT Alpha 30 NOMBRE FONOO              DEFAULT Number 2 NO.FONDO              DEFAULT Number 2 NO.MONEDA              CONS.TOT./LOC.DE MAYORES Y MEN              DEFAULT Number 6 MAYORES MON.EXT.</p>
--	--



<input type="checkbox"/> ME-MEN	DEFAULT	Number	6	MENORES MON.EXT.
<input type="checkbox"/> MN-MAY	DEFAULT	Number	6	MAYORES MON.NAL
<input type="checkbox"/> MN-MEN	DEFAULT	Number	6	MENORES MON.NAL
<input type="checkbox"/> NOMLOC	DEFAULT	Alpha	25	NOM.LOCALIDAD
<input type="checkbox"/> SDOTIPCRE	DEFAULT	Number	9	SDO.TIPO CREDITO
<input type="checkbox"/> TOT-MAY	DEFAULT	Number	6	TOTAL MAYORES
<input type="checkbox"/> TOT-MEN	DEFAULT	Number	6	TOTAL MENORES
<input checked="" type="checkbox"/> LPC28	I-O			CONSULTA CUENTAS DE DESGLOSE
<input type="checkbox"/> NUMSUB	DEFAULT	Number	2	SUBCUENTA
<input type="checkbox"/> T1314	DEFAULT	Number	4	TOTAL DE LA 1314
<input type="checkbox"/> T1315	DEFAULT	Number	4	TOTAL DE LA 1315
<input type="checkbox"/> T1319	DEFAULT	Number	4	TOTAL DE LA 1319
<input type="checkbox"/> T3102	DEFAULT	Number	4	TOTAL DE LA 3102
<input type="checkbox"/> T6102	DEFAULT	Number	4	TOTAL DE LA 6102
<input type="checkbox"/> T6103	DEFAULT	Number	4	TOTAL DE LA 6103
<input type="checkbox"/> T6105	DEFAULT	Number	4	TOTAL DE LA 6105
<input type="checkbox"/> T6301	DEFAULT	Number	4	TOTAL DE LA 6301
<input type="checkbox"/> TOTALES	DEFAULT	Number	8	TOTALES DE COLUM
<input checked="" type="checkbox"/> LPC29	I-O			CONS.CRED.REDESC./FONDOS DE FO
<input type="checkbox"/> NOMCTE	DEFAULT	Alpha	55	NOMBRE DE CLIETE
<input type="checkbox"/> NOMFONDO	DEFAULT	Alpha	30	NOMBRE FONDO
<input type="checkbox"/> NUMCTE	DEFAULT	Number	8	NO.CLIENTE
<input type="checkbox"/> NUMFON	DEFAULT	Number	2	NO.FONDO
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> SALDO	DEFAULT	Number	12	SALDO
<input type="checkbox"/> SUBCTA	DEFAULT	Number	12	CUENTA DESGLOSE
<input checked="" type="checkbox"/> LPC34	I-O			CONS.CTES DE CART.VIG.DE N#30,
<input type="checkbox"/> NOMCTE	DEFAULT	Alpha	55	NOMBRE DE CLTE
<input type="checkbox"/> NUCTB	DEFAULT	Number	4	NO.CTA.BANAMEX
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> SDOACT	DEFAULT	Number	9	SALDO ACTUAL
<input type="checkbox"/> SDOANT	DEFAULT	Number	9	SALDO ANTERIOR
<input type="checkbox"/> TOTAL	DEFAULT	Number	9	TOTAL DE SALDOS
<input type="checkbox"/> VARIACION	DEFAULT	Number	9	VARIACION/IMPORT
<input checked="" type="checkbox"/> LPC35	I-O			CONS.CLTES CON RIESGOS > N\$9,000
<input type="checkbox"/> IMPORTE	DEFAULT	Number	9	IMPORTE
<input type="checkbox"/> NOMCTE	DEFAULT	Alpha	55	NOMBRE DE CLIETE
<input type="checkbox"/> NUACBM	DEFAULT	Number	8	NO.ACT.BANCO MEX
<input type="checkbox"/> NULOBM	DEFAULT	Number	8	NO.LOC.BANXICO
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input checked="" type="checkbox"/> LPC37	I-O			TOTALES POR CUENTA
<input type="checkbox"/> DIF-VALO	DEFAULT	Number	8	DIF.VALORIZADA
<input type="checkbox"/> DIFERENCIA	DEFAULT	Number	8	DIF.DE CUENTAS
<input type="checkbox"/> NUCTB	DEFAULT	Number	4	NO.CTA.BANAMEX
<input type="checkbox"/> SDO-DM	DEFAULT	Number	8	SDO DIARIO MAYOR
<input type="checkbox"/> SDO-DM-VAL	DEFAULT	Number	8	SDO.D.M.VALORIZA
<input type="checkbox"/> SDO-RIE-VA	DEFAULT	Number	8	SDO.RIESGO VALOR
<input type="checkbox"/> SDORIESGO	DEFAULT	Number	8	SALDO DE RIESGOS
<input checked="" type="checkbox"/> LPC38	I-O			CONS.COMPARATIVO SDOS/CUENTA
<input type="checkbox"/> DIF-VALO	DEFAULT	Number	8	DIF.VALORIZADA
<input type="checkbox"/> DIFERENCIA	DEFAULT	Number	8	DIF.DE CUENTAS
<input type="checkbox"/> NOERROR	DEFAULT	Alpha	2	NO. DE ERROR
<input type="checkbox"/> NUCTB	DEFAULT	Number	4	NO.CTA.BANAMEX
<input type="checkbox"/> SDO-DM	DEFAULT	Number	8	SDO DIARIO MAYOR
<input type="checkbox"/> SDO-DM-VAL	DEFAULT	Number	8	SDD.D.M.VALORIZA
<input type="checkbox"/> SDO-RIE-VA	DEFAULT	Number	8	SDO.RIESGO VALOR
<input type="checkbox"/> SDORIESGO	DEFAULT	Number	8	SALDO OE RIESGOS
<input checked="" type="checkbox"/> LPC46	I-O			CONSULTA DEL COMP./SUCURSAL
<input type="checkbox"/> DIF-VALO	DEFAULT	Number	8	DIF.VALORIZADA
<input type="checkbox"/> DIFERENCIA	DEFAULT	Number	8	DIF.DE CUENTAS
<input type="checkbox"/> NOERROR	DEFAULT	Alpha	2	NO. DE ERROR
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL

<input type="checkbox"/> SDO-DM	DEFAULT	Number	8	SDO DIARIO MAYOR
<input type="checkbox"/> SDD-DM-VAL	DEFAULT	Number	8	SDO.D.M.VALORIZA
<input type="checkbox"/> SDO-RIE-VA	DEFAULT	Number	8	SDO.RIESGO VALOR
<input type="checkbox"/> SDORIESGD	DEFAULT	Number	8	SALDO DE RIESGOS
<input type="checkbox"/> TBT21	I-O			ARCHIVO DE GLOBAL/CTA-MONEDA
<input type="checkbox"/> TBT22	I-O			ARCHIVO DE TOTAL/SUC-CTA-MONED
<input type="checkbox"/> ME	DEFAULT	Number	9	MONEDA EXTRANJER
<input type="checkbox"/> MN	DEFAULT	Number	9	MONEDA NACIONAL
<input type="checkbox"/> NUCTB	DEFAULT	Number	4	NO.CTA.BANAMEX
<input type="checkbox"/> NUMSUC	DEFAULT	Number	4	NO.SUCURSAL
<input type="checkbox"/> TOTAL	DEFAULT	Number	9	TOTAL DE SALDOS
<input type="checkbox"/> TBT23	I-O			GLOBAL CLIENTE-MONEDA
<input type="checkbox"/> LPC20	Input			MENU DEL COMPARATIVO DE SALDO
<input type="checkbox"/> S052				SISTEMA CENTRAL CONTABLE
<input type="checkbox"/> S114				SALDOS DEL S114
<input type="checkbox"/> VALIDACIONES				VALIDACION DE CTES.Y RIESGOS
<input type="checkbox"/> LPC19	I-O			MENU DE DIAGNOSTICO
<input type="checkbox"/> OPCION	DEFAULT	Alpha	1	OPCION
<input type="checkbox"/> LPC09	Input			MENU PRINCIPAL
<input type="checkbox"/> OPCION	DEFAULT	Alpha	1	OPCION
<input type="checkbox"/> GD-ENCABEZADO1		Insertable		ENCABEZADO COMPLETO
<input type="checkbox"/> OPCION	DEFAULT	Alpha	1	OPCION

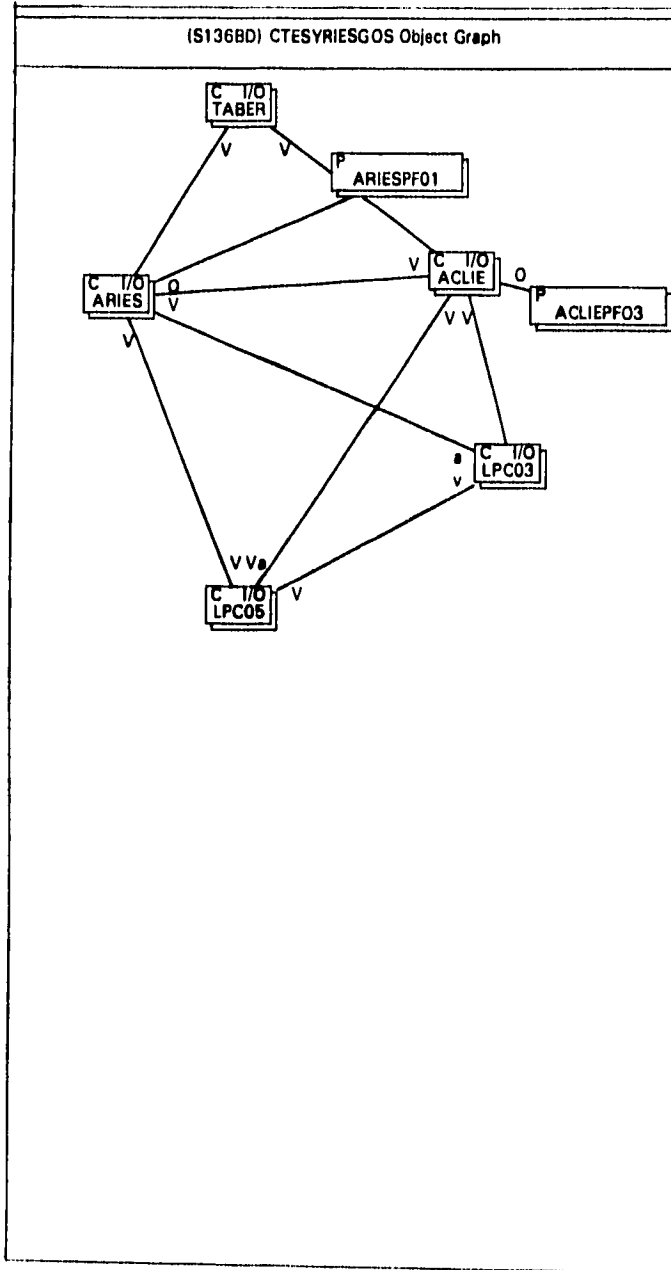
Name	Edit	Len	Dec	Description
ACTESP	Number	7		ACTIVIDAD ESPECIAL
ACTGEN	Number	6		ACTIVIDAD GENERICA
AGENCIA	Number	3		
CAREDESCON	Number	9		CARTERA REDESCONTADA
CARVENCIDA	Number	9		CARTERA VENCIDA
CARVIGENTE	Number	9		CARTERA VIGENTE
CDIG	Number	1		
CTACONT	Number	12		CUENTA CONTABLE
CVEACT	Number	1		CVE.ACTUALIZACION EN EL MES
CVEAVAL	Number	6		
DL.S6101	Alpha	1		
DL.S6102	Alpha	1		
DIF-VALO	Number	6		
DIFERENCIA	Number	8		
LSTATUS	Number	1		
FLCHA	Date	6		FECHA
FLCHALTA	Date	6		
FECHARRIBO	Date	6		
FECHAULT	Date	6		FECHA DEL ULTIMO MOVIMIENTO
FEIN	Number	4		
GPOECO	Number	2		
IMPORTE	Number	9		
INTEGRA	Date	6		FECHA DE INTEGRACION
LETRA	Alpha	1		LETRA DEL FONDO
ME	Number	9		
ME-MAY	Number	6		
ME-MEN	Number	6		
MN	Number	9		
MN-MAY	Number	6		
MN-MEN	Number	6		
MVA CEP	Number	8		MOVIMIENTOS ACEPTADOS
MVRECH	Number	8		MOVIMIENTOS RECHAZADOS
NIVRIES	Alpha	1		NIVEL DE RIESGO O CALIFICA
NOCTCARVEX	Number	9		
NOCTCRVIG	Number	9		
NOCTCVIGEN	Number	9		
NOCTESCARV	Number	9		
NOENFE	Alpha	30		
NOERROR	Alpha	2		NUMERO DE ERROR
NOM-LOC-AC	Alpha	60		NOMBRE DE LA ACT.O LOCALIDAD
NOMACGEN	Alpha	30		NOMBRE DE ACTIVIDAD GENERICA
NOMACT	Alpha	30		NOMBRE DE ACTIVIDAD ESPECIFICA
NOMACT-MEN	Alpha	30		
NOMBCA	Alpha	13		NOMBRE DE LA BANCA
NOMCSI	Alpha	15		NOMBRE DEL C.S.I.
NOMCTA	Alpha	30		
NOMCTE	Alpha	55		NOMBRE DEL CLIENTE
NOMEDO	Alpha	30		NOMBRE DEL ESTADO
NOMFONDO	Alpha	30		NOMBRE DEL FONDO
NOMLOC	Alpha	25		NOMBRE DE LA LOCALIDAD
NOMONEDA	Alpha	30		NOMBRE DE LA MONEDA
NOMSE	Alpha	30		NOMBRE DEL SECTOR ECONOMICO
NOMSEC	Alpha	30		NOMBRE DEL SECTOR BANXICO
NOMSUB	Alpha	30		NOMBRE DE SUBCUENTA
NOMSUC	Alpha	30		NOMBRE DE LA SUCURSAL
NOMTCARTE	Alpha	20		
NOREP	Number	2		
NUACBA	Number	5		
NUACBM	Number	8		
NUACTGEN	Number	2		
NUCTB	Number	4		NO. DE LA CUENTA BANAMEX
NUCTBXICO	Number	2		
NUDIREG	Number	4		
NUEDO	Number	2		NUMERO DEL ESTADO
NULOBA	Number	5		
NULOBM	Number	8		
NUM-ACT-LO	Number	6		NUMERO DE LOCALIDAD O ACTIVIDA

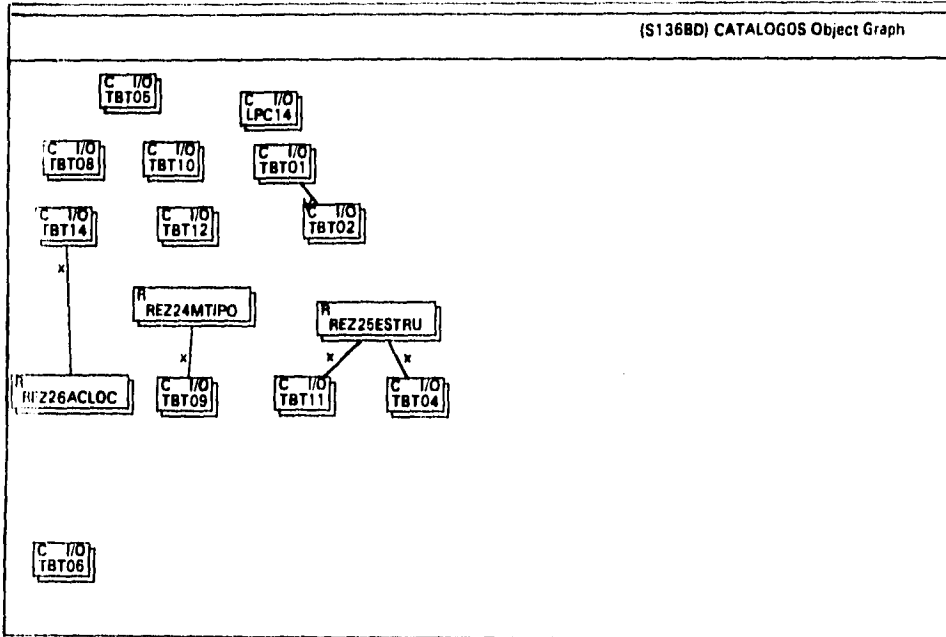
Name	Edit	Len	Dec	Description
MAC-MEN	Number	2		
MBC	Number	2		
MCON	Number	2		NUMERO DE CONVENIO
MCONT	Number	2		NUMERO DE CONTRATO
MCSI	Alpha	2		
MCTE	Number	8		
MFON	Number	2		NUMERO DE FONDO
MHO	Number	2		NUMERO DE LA HORA
MON	Number	2		NUMERO DE MONEDA
MRANGO	Number	2		
MREG	Number	2		NUMERO DE REGISTRO
MSUB	Number	2		NUMERO DE SUBCUENTA
MSUC	Number	4		NUMERO DE SUCURSAL
MUSEC	Number	2		NUMERO DE SECTOR
MUSECO	Number	2		
MVOCRE	Alpha	1		NUEVO CREDITO
OPCION	Alpha	1		OPCION DE MENUES
RANGOINFER	Number	12		
RANGOSUP	Number	12		
REGACEPT	Number	6		
REGACTDO	Number	6		
REGACTUAL	Number	6		
REGANTE	Number	6		
REGRECHAZ	Number	6		
REGSINACT	Number	6		
RFC	Alpha	13		REGISTRO FEDERAL DE CAUSANTES
RPRAC	Number	6		
RPRAN	Number	6		
RPRM	Number	6		
SALDO	Number	12		SALDD
SDO-6101	Number	9		
SDO-6102	Number	9		
SDO-OM	Number	8		
SDO-OM-VAL	Number	8		
SDO-RIE-VA	Number	8		
SDO-S114	Number	9		
SDOACT	Number	9		
SDOANT	Number	9		
SDORIESGO	Number	8		
SDOTIPCRE	Number	9		
SDOTOTAL	Number	9		
SEC14	Number	5		
SEC22	Number	5		
SEC23	Number	5		
SEC26	Number	5		
SEC31	Number	5		
SEC32	Number	5		
SEC41	Number	5		
SEC42	Number	5		
SIGNO	Alpha	1		
SISARRIB	Number	2		
SISORIGEN	Number	2		NUMERO DE SISTEMA ORIGEN
SUBCTA	Number	12		CUENTA DE OESGLOSE(SUBCTA)
T1314	Number	4		TOTAL DE LA CUENTA 1314
T1315	Number	4		
T1319	Number	4		
T3102	Number	4		
T6102	Number	4		
T6103	Number	4		
T6105	Number	4		
T6301	Number	4		
TICAMB	Number	7	7	
TICARTE	Number	1		
TIMON	Number	1		
TIPCRE	Number	2		
TIPOPER	Number	1		TIPO DE PERSONA
TIPORIE	Alpha	1		

MATRIZ Area Functional / Activity Matrix

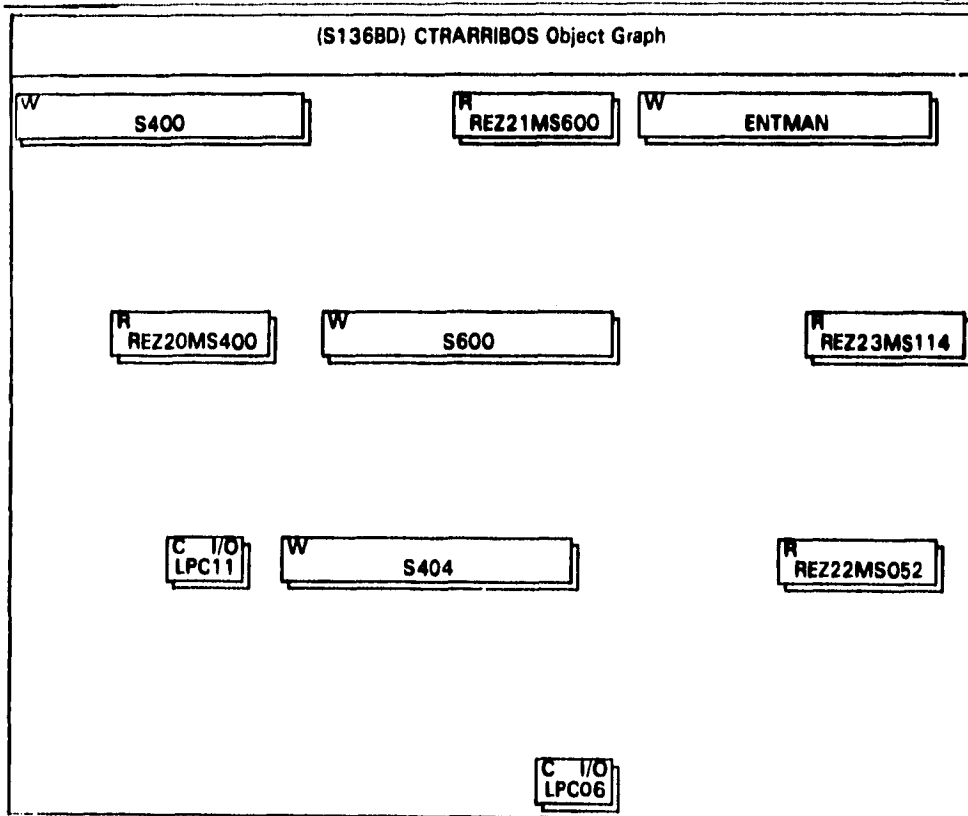
Activity	Functional Area
ADSDOATM	S136BD-DEFAULT:X
CATALOGOS	S136BD-DEFAULT:X
CTESYRIESGOS	S136BD-DEFAULT:X
CTRARRIBOS	S136BD-DEFAULT:X
GENREPORTES	S136BD-DEFAULT:X
MODIFCTES	S136BD-DEFAULT:X
PRUEBA	S136BD-DEFAULT:X
TOTALESR	S136BD-DEFAULT:X
VALIDACIONES	S136BD-DEFAULT:X

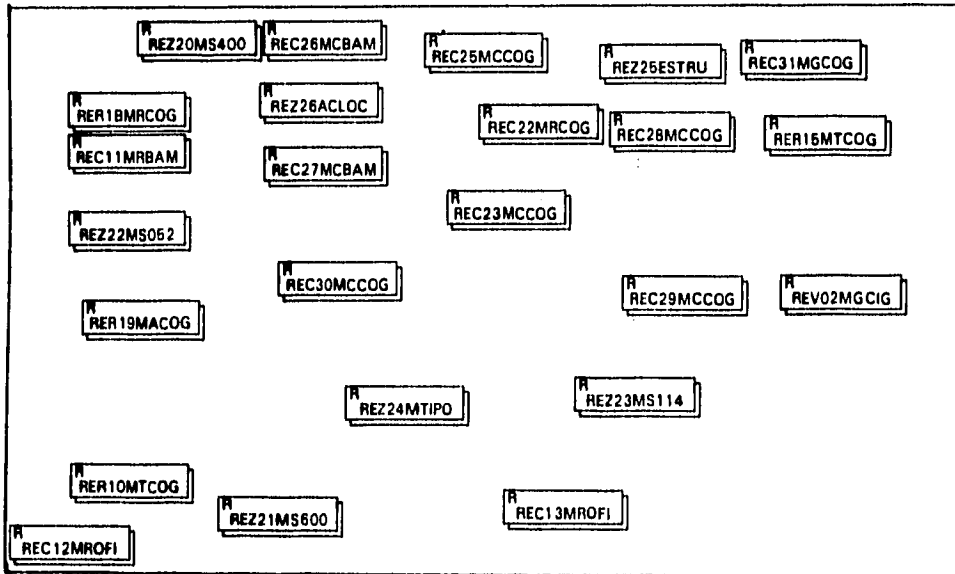
Activity	Object
ADSDOATM	TBT21:X TBT23:X TBT22:X LPC33:X LPC44:X LPC43:X LPC42:X
CATALOGOS	LPC41:X LPC39:X LPC32:X LPC31:X LPC15:X LPC17:X LPC02:X
	TBT07:X TBT14:X TBT12:X TBT11:X TBT10:X TBT09:X TBT08:X
	TBT06:X TBT05:X TBT04:X TBT02:X REZ24MTIPO:X REZ26ACLOC:X
CTESYRIESGOS	REZ25ESTRU:X LPC14:X
	TABER:X LPC05:X LPC03:X ARIESPF01:X ARIES:X ACLIEFF03:X
CTRARRIBOS	ACLIE:X
	REZ23MS114:X REZ22MS052:X REZ21MS600:X REZ20MS400:X
GENREPORTES	LPC11:X LPC08:X
	TBT27:X TBT25:X TBT24:X REV02MGCIG:X RER19MACOG:X
	RER18MRCOG:X RER15MTCOG:X RER10MTCOG:X REC31MCGOG:X
	REC30MCCOG:X REC29MCCOG:X REC28MCCOG:X REC27MCBAM:X
	REC26MCBAM:X REC25MCCOG:X REC23MCCOG:X REC22MRCOG:X
	REC13MROFI:X REC12MROFI:X REC11MRBAM:X LPC24:X LPC22:X
MODIFCTES	LPC21:X
PRUEBA	LPC16:X
TOTALESR	PANT9:X
	TBT23:X TBT22:X TBT21:X LPC46:X LPC37:X LPC38:X LPC35:X
VALIDACIONES	LPC34:X LPC29:X LPC28:X LPC27:X LPC26:X LPC25:X LPC20:X
	LPC19:X











1. Isopaca

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces		
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C
ACLIE	0	7	18	0	5	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0
ARIES	0	9	21	0	5	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0
LPC02	0	1	3	0	5	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
LPC03	0	5	9	0	5	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
LPC05	0	7	14	0	5	0	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0
LPC06	0	0	8	0	5	0	0	3	0	0	0	0	0	1	1	0	0	0	0	0	0
LPC09	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC11	0	0	1	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC14	0	0	1	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC15	0	1	5	0	5	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC16	0	0	1	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC17	0	0	1	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC19	0	0	0	0	5	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LPC20	0	0	0	0	5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
LPC21	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC22	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC24	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC25	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC26	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC26	0	2	4	0	5	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC27	0	0	9	0	5	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0
LPC28	0	0	1	10	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC29	0	0	5	7	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC31	0	0	2	7	0	5	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
LPC32	0	0	12	0	5	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC33	0	0	1	6	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC34	0	0	2	7	0	5	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
LPC35	0	0	1	5	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC37	0	0	1	7	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC38	0	0	2	8	0	5	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
LPC39	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC41	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC42	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPC43	0	0	1	1	5	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0
LPC44	0	0	0	6	0	5	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
LPC46	0	0	2	8	0	5	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
PANT9	0	0	2	0	5	0	3	0	0	0	0	0	1	0	0	1	0	0	0	0	0
TABER	0	0	3	3	0	5	0	3	0	0	0	0	1	0	0	1	0	0	0	0	0
TBT01	0	0	0	3	0	5	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0
TBT02	0	0	1	4	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT04	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT05	0	0	0	4	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT06	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT08	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT09	0	0	0	6	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT10	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT11	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT12	0	0	0	3	0	5	0	0	3	0	0	0	0	1	0	0	0	0	0	0	0
TBT14	0	0	0	8	0	5	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0
TBT21	0	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
TBT22	0	0	2	5	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
TBT23	0	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
TBT24	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TBT25	0	0	0	0	0	5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
TBT27	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

**2 Reports**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
REC11MRBAM	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC12MROFI	0	0	0	0	0	6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC13MROFI	0	0	0	0	0	6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC22MRCOG	0	0	0	0	0	19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC23MCCOG	0	0	0	0	0	21	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
REC25MCCOG	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC26MCBAM	0	0	0	0	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC27MCBAM	0	0	0	0	0	18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC28MCCOG	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC29MCCOG	0	0	0	0	0	13	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC30MCCOG	0	0	0	0	0	12	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REC31MCGOG	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RER10MTCOG	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RER15MTCOG	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RER18MRCOG	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RER19MACOG	0	0	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REVO2MGCIG	0	0	0	0	0	18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REZ20MS400	0	0	0	0	0	38	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
REZ21MS600	0	0	0	0	0	34	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
REZ22MS052	0	0	0	0	0	22	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
REZ23MS114	0	0	0	0	0	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REZ24MTIPO	0	0	0	0	0	11	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
REZ25ESTRU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
REZ26ACLOC	0	0	0	0	0	6	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

**3 Profiles**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
ACLIEPF03	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
ARIESPF01	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

**4 Wildcards**

Name	Refer		DataItems				Inputs			Outputs			Inquiries			Files			Interfaces			
	Upd	Rf	IO	O	I	Iq	S	A	C	S	A	C	S	A	C	S	A	C	S	A	C	
ENTMAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S052	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S400	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S404	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**5. Totals****5.1 External Inputs**

Simple	24	*	3	=	72
Average	45	*	4	=	180
Complex	19	*	6	=	114
<b>Total</b>	<b>88</b>				<b>366</b>

**5.2 External Outputs**

Simple	19	*	4	=	76
Average	4	*	5	=	20
Complex	0	*	7	=	0
<b>Total</b>	<b>23</b>				<b>96</b>

**5.3 External Inquiries**

Simple	6	*	3	=	18
Average	11	*	4	=	44
Complex	3	*	6	=	18
<b>Total</b>	<b>20</b>				<b>80</b>

**5.4 Logical Master Files**

Simple	42	*	7	=	294
Average	0	*	10	=	0
Complex	0	*	15	=	0
<b>Total</b>	<b>42</b>				<b>294</b>

**5.5 Interfaces to Other Systems**

Simple	0	*	5	=	0
Average	0	*	7	=	0
Complex	0	*	10	=	0
<b>Total</b>	<b>0</b>				<b>0</b>

**6 Processing Complexity**

1	Data Communications	4
2	Distributed Data or Processing	4
3	Performance Objectives	5
4	Tight Configuration	5
5	High Transaction Rate	5
6	On-line Inquiry/Data Entry	3
7	End User Efficiency	5
8	On-line Update	5
9	Complex Processing	5
10	Code Reusability	5
11	Conversion/Installation Ease	5
12	Operational Ease	5
13	Multiple Site Installation	5
14	Facilitate Change	5

**7 Totals**

Total Unadjusted Function points are 836

Total Degree of Influence is 66

Adjustment Factor is  $(0.65 + (0.01 * 66)) = 1.31$

Final Function Point Count is 1095

**ANALISIS DEL COSTO-BENEFICIO DE LA RE-INGENIERIA DE SOFTWARE**

Haciendo una evaluación de los costos operativos, beneficios y riesgos, el costo beneficio del sistema nuevo es el siguiente :

**Costo operativo del sistema actual (C.P.A.) = N\$ 6 000.00**

**Costo operativo del sistema propuesto (C.P.P.) = N\$ 3 000.00**

**3 000.00 Costo a favor**

**C.P.A / C.P.P = 6 000.00 / 3 000.00 = 2 50% de Costo a favor**

**Plazo de recuperación :**

**Costo total del nuevo sistema (C.N.S.) = N\$ 95 000.00**

**Costo a favor (C.A.F) = N\$ 3 000 00**

**C.N.S / C.A.F = 95 000.00 / 3 000.00 = 31.6 MESES.**

Para obtener los totales aproximados de cada uno de los conceptos se tomaron en cuenta insumos, recursos físicos, recursos humanos y tecnológicos y que no se desglosan por no haber obtenido autorización del banco.

Así pues, al finalizar el proceso de re-ingeniería del sistema actual de Responsabilidades de Cartera, veremos algunos de los beneficios en el siguiente cuadro comparativo :

<b>SISTEMA ACTUAL</b>	<b>SISTEMA NUEVO (Con Re-ingeniería)</b>
Utiliza archivos	Utiliza el concepto de Base de Datos
Programado en COBOL ANSI 74 (3a. generación)	Programado en LINC (4a. generación)
Hay una gran intervención del usuario (captura y corrección de errores)	Disminuye la intervención del usuario siendo todo más transparente para él.
Emite 55 reportes (13,000 páginas)	Emite sólo 16 (Reducción de un poco más del doble de páginas aproximadamente)
El usuario consulta en listados para después mediante un proceso de línea hacer algún cambio en la información.	El usuario ahora puede consultar y realizar cambios en la información mediante pantallas (23 pantallas).
El tiempo de mantenimiento es muy alto (aproximadamente 120 horas al mes)	Reducción en los reportes por falla y de requerimientos lo que permite un considerable decremento en el tiempo de mantenimiento.
Tiempo de proceso del sistema en general es de 15 a 20 días	Tiempo de proceso del sistema en general es de 3 a 5 días.
Costo N\$ 6000.00 aproximadamente	Costo de N\$ 3000.00 aproximadamente
Hay una cantidad considerable de errores en la información lo que en ocasiones se refleja en diferencias en la contabilidad.	Disminución en el número de errores por tener un mayor control de la información.
	Permite un mayor control de los registros con error para su integración al proceso.

*Tabla No. 8*

*Comparativo Sistema Actual vs. Sistema Nuevo con Re-ingeniería*



## CONCLUSIONES

Entre los temas con mayor auge dentro del mundo empresarial en México figura el de Re-ingeniería de procesos de negocios del cual encontramos una vasta bibliografía, ciclos de conferencias y gente especializada en el tema.

Al seleccionar el tema de re-ingeniería de software nos enfrentamos al problema de falta de bibliografía sobre el mismo, ya que en nuestro país el tema es relativamente nuevo aunque de alguna manera se lleva a cabo en las empresas sin saber que se está haciendo re-ingeniería de software.

Consideramos importante aplicar la re-ingeniería de software ya que en nuestro país existen sistemas en la banca mexicana por ejemplo, que datan desde los años 70's y actualmente todavía están en producción lo que implica un mantenimiento correctivo excesivo ya que éstos sistemas deben ir adecuándose al crecimiento de los volúmenes de información, a las condiciones y requerimientos de la economía conforme se van presentando, haciendo modificaciones para solucionar un determinado problema o requerimiento sin considerar las repercusiones que estos cambios pueden tener en un futuro lo que hace aún más difíciles las tareas de mantenimiento.

Al comenzar a hacer el análisis del sistema de "Responsabilidades de Cartera" nos encontramos con que no existía documentación actualizada para iniciar el proceso de re-ingeniería por lo que tuvimos que recurrir a múltiples entrevistas con las personas que conocen el sistema. Debido a esto primero se tuvo que actualizar la documentación auxiliándonos para ello de la metodología propia del Banco.

Por lo anterior, es que sugerimos que desde la creación de un sistema se vaya documentando y actualizando periódicamente; ya que las condiciones del sistema actual y su documentación son importantes para determinar si la re-ingeniería de software es práctica y de ser así como se va a realizar.

Durante el desarrollo del tema nos dimos cuenta de la importancia que reviste el llevar a cabo el ciclo de la re-ingeniería de software, pues ésta nos permite obtener un sistema con un alto grado de análisis y con un bajo nivel de mantenimiento con la utilización de las herramientas CASE. Aunque considerando el enfoque de muchas de éstas herramientas para un ambiente de cómputo en particular, debemos señalar que quizás un conjunto de las mismas no satisfaga todas las necesidades que se tienen dentro de una organización. Así mismo, existen diferentes proveedores de las herramientas CASE por lo que se debe tomar en cuenta la formas de intercambio e interrelación de datos y su modelo entre las diferentes herramientas, por lo que es recomendable que al tomar una decisión de la herramienta a utilizar prever todas las migraciones futuras, tanto en hardware como en software.

La herramienta CASE que se utilizó para el análisis y diseño fue elegida porque es la que más se amolda a los requerimientos del Banco.

## **CONCLUSIONES**

**La re-ingeniería de software para cualquier caso en particular debe estar soportada por:**

- **Enfoque y objetivos del sistema**
- **Diseño e implementación del sistema a rediseñar que debe ser reciente**
- **Que el código sea estructurado**
- **Que la documentación del sistema actual este completa**
- **Que existan herramientas que soporten la mayoría de las tareas de la re-ingeniería.**

La re-ingeniería de software puede ser considerada como la migración a un nuevo ambiente, la reducción de costos de mantenimiento, el entendimiento más fácil de la complejidad de los sistemas de aplicación, el equilibrio en los sistemas de aplicación, la reducción de los errores en el software, la recuperación mas efectiva de la información.

Por último la re-ingeniería de software implica un trabajo arduo pero que si se lleva a cabo correctamente nos permitirá ser los mejores en nuestro negocio.



**GLOSARIO**

**GLOSARIO**

**ACTIVIDAD:**

Un grupo de objetos que juntos desarrollan una función del negocio.

**AD HOC:**

Quiere decir que es la forma más adecuada para cubrir las necesidades de algo.

**ALIAS:**

Nombre dado a alguien o algo en lugar de su propio nombre. Usado en informática para dar un nombre más corto, claro o más identificable a una variable.

**AMBIENTE DISTRIBUIDO:(Distributed Computer Environment)**

Conjunto integrado de sistemas operativos y de servicios independientes de la red que soportan el desarrollo, uso y mantenimiento de sistemas.

**ARCHIVO MAESTRO:**

La colección más importante de datos que pertenecen a una aplicación específica.

**AREA FUNCIONAL:**

Un grupo de actividades. También es usada para agrupar actividades con el propósito de cargarlas en LINC.

**BATCH:**

Estructura de control lógica en la que se ejecutan una serie de instrucciones en el orden en que aparecen

**BOTTOM UP:**

Es comenzar algo de lo específico a lo general

**CARTERA VENCIDA**

Es un concepto propio del banco el cual es una relación de los clientes con problemas de pago y cuya deuda con el banco es mayor a doscientos mil nuevos pesos.

**CARTERA VIGENTE:**

Es un concepto bancario el cual es una relación de los clientes que pagan puntualmente

**CASE:**

Computer Asisted System Engineering (Ingeniería de Sistemas Asistido por Computadora)  
La utilización de Software para ayudar en la definición, elaboración, documentación y algunas otras áreas del desarrollo de programación.

**CICS :**

Customer Information Control System

**CLIENTE-SERVIDOR:**

Es una arquitectura de distribución del procesamiento de la información basada en redes. Modelo de trabajo donde el proceso se distribuye entre dos entidades: la que solicita(cliente) y la que da el servicio(servidor)

**COBOL (Common Bussines Managment System)**

Lenguaje de alto nivel desarrollado para aplicaciones de procesamiento de dtos de negocios.

**COMS (Communications Management System)**

Es un control del ambiente en línea de las series A. COMS soporta el procesamiento de multiprogramas.

## GLOSARIO

### **DB2:**

Data Base 2 Manejador de Bases de Datos de IBM para ambiente MVS(mainframes) Utiliza SQL y define en sí mismo un dialecto estándar

### **DBMS (Data Base Management System)**

Sistemas Manejador de Bases de Datos

### **DP:**

Desarrollo de Proyectos

### **ELEMENTO:**

Es una área funcional, actividad, objeto.

### **EVENTO :**

Es un campo de una de Base de Datos

### **FPA: (Function Point Analysis)**

Análisis de Puntos por Función. Técnica desarrollada para medir la complejidad de un sistema.

### **GUI (Graphic User Interface)**

Enlace de comunicación o interfaz entre un usuario y el sistema operativo de una computadora. Generalmente utiliza pantallas diseñadas con base en iconos (figuras) que representan las funciones disponibles para el usuario. Windows de Microsoft es un ejemplo de una GUI.

### **HERRAMIENTAS:**

Los elementos que se utilizan para realizar algo

### **IEE (Institute of Electrical and Electronics Engineers)**

Instituto de Ingenieros Eléctricos y Electrónicos

### **INGENIERÍA HACIA ADELANTE: (Forward Engineering)**

Es ir de una especificación a un sistema operacional de Usuario

### **INGENIERÍA REVERSIVA:**

Es la existencia de sistemas variando sus niveles de especificación

### **JOB:**

Es un proceso independiente. Un WFL es un job.

### **LDA: (Linc Design Assistant)**

LDA es una herramienta basada en computadoras personales, la cual en conjunción con el medio de desarrollo de LINC, soporta las siguientes fases de desarrollo de sistemas:

### **MODELO:**

El segmento de la organización para el cual el proyecto de desarrollo es tomado.

### **OBJETO :**

Reporte, componente, evento, pantalla de entrada.

### **PARADIGMAS:**

Es un ejemplo o modelo especialmente claro o arquetípico. Conjunto de reglas orientadas a establecer límites y a describir cómo solucionar problemas dentro de esos límites. En un sentido más amplio, puede considerarse un modelo que ayuda a comprender o que se ve y se oye.

### **PROTOTIPO:**

Simulación de las llamadas a pantallas del modelo.

### **RDBMS (Relational Data Base Management System).**

Sistema Manejador de Bases de Datos Relacionales.

## GLOSARIO

**RELACIÓN:**

Puede ser de actualización, de borrado, de alta, de lectura, de llamado, de validación automática de LINC.

**RUN TIME:**

Es el tiempo de proceso

**SENICREB:**

Servicio Nacional de Información de Crédito Bancario

**SOFTWARE:**

Término usado para describir todos los tipos de programas, incluyendo los programas del sistema operativo y los programas de aplicaciones

**TOP DOWN:**

Significa ir de lo general a lo específico

**WFL (Work Flow Language):**

Es un lenguaje de Unisys usado para construir jobs que compilan y corren programas. Los WFL's contienen variables y expresiones.

**WILDCARD:**

Cualquier entidad que no es un objeto, actividad o area funcional apropiada del modelo, por ejemplo: un archivo o un sistema no LINC.



***BIBLIOGRAFIA***

## BIBLIOGRAFIA

## BIBLIOGRAFIA

### Libros

- [1] Arnold, Robert S., Software Reengineering,  
EEUU, IEEE Computer Society, 1993
- [2] Atre, Shakuntala, Técnicas de Bases de Datos,  
México, Edit. Trillas, 1988
- [3] Ghezzi, Carlo Fundamentals of Software Engineering,  
EEUU
- [4] Holloway, Simon, Fourth Generation Systems,  
EEUU, Edit Prentice Hall
- [5] Martin, Fourth Generation Languages,  
EEUU, Volume I,II
- [6] Michael Hammer and James Champy Re-ingeniería  
Pág. 50
- [7] Pressman, Roger S., Ingeniería de Software,  
España, Edit. Mc. Graw Hill, 1993
- [8] Toniz G., Marina, Re-ingeniería de Software,  
McGraw-Hill, 1995

### Revistas

- [9] Bloor Robin, Tácticas for Downsizing, DBMS, diciembre 1992, 14-16
- [10] Bort Julie. Redes: El coco de los mainframes, Personal Computing México,  
Núm. 49, 1992, 33-38
- [11] IBM, System Journal, Vol.29, No.4, 1990.
- [12] IEEE, Software, Vol. 7, No. 1, Enero 1990, pp. 13-17
- [13] IEEE, Conference on Software Maintenance, 1989, pp. 174-179
- [14] IEEE, Transactions on Software Engineering, Agosto 1991, pp. 751-761
- [15] IEEE, Vol. 77, No. 4, Abril 1989, pp. 607-617



## BIBLIOGRAFIA

- [16]IEEE, Software Maintenance, 1987, pp.82-90
- [17]Journal of Software Maintenance, Vol 3, No. 3, Sept. 1991, pp.163-182  
Economics of Software Re-engineering", por H.M. Sneed
- [18]Journal of Software Maintenance, 1990
- [19]National Institute of Standards and Technology Special Publication, pp.500-193,  
Sept. 1991.
- [20]Software Engineering 1991
- [21]Software Magazine, Oct. 1991; Sep. 1989
- [22]Saldivar Juan, Downsizing: para una mayor eficiencia de recursos,  
Red, enero 1992, 16-20

### Conferencias

- [23]Business Reengineering in Real Life  
Nashville, Tennessee, 11 al 14 de abril de 1994



**APENDICE 1**

WDW FILE: L18/S136/WFL/LOTE/93MTP037 (05/02/95)

```
500 %          %          %          %          %          %          %          %          %          %          %
600 %          %          %          %          %          %          %          %          %          %          %
900 %          %          %          %          %          %          %          %          %          %          %
1200 %         %          %          %          %          %          %          %          %          %          %
1500 %         %          %          %          %          %          %          %          %          %          %
1800 %        %          %          %          %          %          %          %          %          %          %
2100 %        %          %          %          %          %          %          %          %          %          %
2400 %        %          %          %          %          %          %          %          %          %          %
2700 %        %          %          %          %          %          %          %          %          %          %
3000 %        %          %          %          %          %          %          %          %          %          %
3300 %        %          %          %          %          %          %          %          %          %          %
3600 %         %          %          %          %          %          %          %          %          %          %
3900 %         %          %          %          %          %          %          %          %          %          %
4200 %         %          %          %          %          %          %          %          %          %          %
4500 %         %          %          %          %          %          %          %          %          %          %
4800 %        %          %          %          %          %          %          %          %          %          %
5100 %        %          %          %          %          %          %          %          %          %          %
5400 %        %          %          %          %          %          %          %          %          %          %
5700 %        %          %          %          %          %          %          %          %          %          %
6000 %        %          %          %          %          %          %          %          %          %          %
6300 %        %          %          %          %          %          %          %          %          %          %
6600 %        %          %          %          %          %          %          %          %          %          %
6900 %        %          %          %          %          %          %          %          %          %          %
7200 %        %          %          %          %          %          %          %          %          %          %
7500 %        %          %          %          %          %          %          %          %          %          %
7800 %         %          %          %          %          %          %          %          %          %          %
8100 %  AUTOR : JOSE ANGEL SANTOS MEMDOZA  IMPDIF : EDUARDO RUBIO G
8400 %  FECHA : 6 DE AGOSTO DE 1986          FECHA : 11 ABRIL DE 1991
8700 %
9000 % BEGIN JOB S136/WFL/LOTE  (INTEGER FECHA, INTEGER CORR, INTEGER OPCION,
9300 %         INTEGER TIPO);
9600 % S INCLUDE S136/WFL/INCLUDE/LOTE;
9900 %
10200 %
10500 %  STRING BAJA, LISTADO, RESCINTO, RESCINTR, A, M, D, MA, S136BK020REL, FECH,
10800 %         S136BK040REP, S136BK040MICPEN, S136BK041MICWAY, S136BK043MICWAY,
11100 %         S136BK050REPO, S136BK050REP1, S136BK096REP, S136BK098REP,
11400 %         S136BK099REP, S136BK140FOM, S136BK150REP, S136BK190REP;
11700 %  INTEGER WFECB, WFEC3, WFEC4, WFEC1, DIA, MES, AÑO, VECES, OPTI, CORR, NUMTASH
12000 %         WFEC11, PARAM;
12300 %  TASK COPI, RESP, TASK005, TASK010, TASK020, IMP005, IMP010, IMP020, TRESP,
12600 %         TASK040, IMP040, TASK041, IMP041, TASK042, IMP042, TASK043, IMP043,
12900 %         TASK003, TASK004,
13200 %         TASK050, IMP050, TASK060, IMP060, TASK093, IMP093, TASK095, IMP095,
13500 %         TASK096, IMP096, TASK097, IMP097, TASK098, IMP098, TASK099, IMP099,
13800 %         TASK110, IMP110, TASK150, IMP150, TASK160, IMP160, TASK170, IMP170,
14100 %         TASK140, IMP140, TASK180, IMP180, TASK190, IMP190, TASK070, IMP070,
14400 %         TASK008, TASK011, TASK601;
14700 %
15000 %  %          %          %          %          %          %          %          %          %          %          %
15300 %  %          %          %          %          %          %          %          %          %          %          %
15600 %  %          %          %          %          %          %          %          %          %          %          %
15900 %  %          %          %          %          %          %          %          %          %          %          %
16200 %  %          %          %          %          %          %          %          %          %          %          %
16500 %  %          %          %          %          %          %          %          %          %          %          %
16800 %  %          %          %          %          %          %          %          %          %          %          %
17100 %  SUBROUTINE VALFECHA
17400 %  @BEGIN
17700 %  *CORR = CORR,
```

```

      END
6947C  END.
69600  %
6990C  %
70200  %
70500  %   XXXXXX   XXXXXX   XXXXXXXX
70800  %  %%   %%   %%   %%   %%
71100  %  %%   %%   %%   %%   %%
71400  %  %%   %%   %%   %%   XXXXXXXX
71700  %  %%   %%   %%   %%   %%
72000  %  %%   %%   %%   %%   %%
72300  %  %%   %%   %%   %%   %%
72600  %   XXXXXX   XXXXXX   XXXXXXXX
72900
73200  SUBROUTINE CORREPAS003;
73500  BEGIN
73800  STRING SIOND;
74100  %
74400  PAS0003;
74700  WFEC3:= WFEC3 + 100;
75000  WFEC3:= WFEC3 + CORRI;
75300  TASK003(VALUE = WFEC3);
75600  RUN S136/OBJECT/P003/95MTP029 [TASK003];
75900  OPTION = (DBBASE, FAULT, DSED, ARRAYS, CODE, FILES, DBB);
76200  %
76500  IF TASK003 ISNT COMPLETEOK OR TASK003(VALUE)=1 THEN
76800  BEGIN
77100  SIOND:= ACCEPT("FALLO S136/PAS0003 RETRY ?? SI/NO");
77400  SIOND:= HEAD(TAIL(SIOND," "),NOT " ");
77700  IF SIOND = "SI" THEN
78000  GO TO PAS0003;
78300  END;
78600  END;
78900  %
79200  %
79500  %   XXXXXX   XXXXXX   %%   %%
79800  %  %%   %%   %%   %%   %%   %%
80100  %  %%   %%   %%   %%   %%   %%
80400  %  %%   %%   %%   %%   XXXXXXXX
80700  %  %%   %%   %%   %%   %%
81000  %  %%   %%   %%   %%   %%
81300  %  %%   %%   %%   %%   %%
81600  %   XXXXXX   XXXXXX   %%
81900
82200  SUBROUTINE CORREPAS004;
82500  BEGIN
82800  STRING SIOND;
83100  %
83400  PAS0004;
83700  WFEC4:= WFEC4 + 100;
84000  WFEC4:= WFEC4 + CORRI;
84300  TASK004(VALUE = WFEC4);
84600  RUN S136/OBJECT/P004/95MTP021 [TASK004];
84900  OPTION = (DBBASE, FAULT, DSED, ARRAYS, CODE, FILES, DBB);
85200  %
85500  IF TASK004 ISNT COMPLETEDOK OR TASK004(VALUE)=1 THEN
85800  BEGIN
86100  SIOND:= ACCEPT("FALLO S136/PAS0004 RETRY ?? SI/NO");
86400  SIOND:= HEAD(TAIL(SIOND," "),NOT " ");
86700  IF SIOND = "SI" THEN
87000  GO TO PAS0004;
87300  END;
87600  END
87900  %
88200  %   XXXXXX   XXXXXX   XXXXXXXX
      %  %%   %%   %%   %%   %%

```

```

10800 % XX XX XX XX XX
10810 % XX XX XX XX XXXXXXXX
10820 % XX XX XX XX XX XX
10830 % XX XX XX XX XX XX
10840 % XX XX XX XX XX XX
10850 % XXXXXXXX XXXXXXXX XXXXXXXX
10860
10870 SUBROUTINE CORREPAS005,
10880 BEGIN
10890 STRING SIOND;
10900 %
10910 PAS0005:
10920 WFECH:= WFECH * 100;
10930 WFECH:= WFECH + CORR1;
10940 TASK005(VALUE = WFECH);
10950 RUN S136/OBJECT/P005/95MTP008 [TASK005];
10960 OPTION = (BOBASE,FAULT,DSEED,ARRAYS,CODE,FILES,DBS);
10970 FILE ABOCIFRAS (FORMMESSAGE = "A-25");
10980 BNAME = S136/LIST/P005;
10990 %
11000 IF TASK005 ISNT COMPLETEDOK OR TASK005(VALUE)=1 THEN
11010 BEGIN
11020 SIOND:= ACCEPT("FALLO S136/PAS0005 RETRY ?? SI/NO");
11030 SIOND:= HEAD(TAIL(SIOND," "),NOT " ");
11040 IF SIOND = "SI" THEN
11050 GO TO PAS0005;
11060 END;
11070 END;
11080 %
11090 IMPRIME005:
11100 RUN SYSTEM/FORZA ("S136/LIST/005")[IMPO05];
11110 IF IMPO05 ISNT COMPLETEDOK THEN
11120 GO IMPRIME005;
11130 END;
11140 %
11150 % XXXXXXXX XX XXXXXXXX
11160 % XX XX XXX XX XX
11170 % XX XX XXX XX XX
11180 % XX XX XX XX XX XX
11190 % XX XX XX XX XX XX
11200 % XX XX XX XX XX XX
11210 % XX XX XX XX XX XX
11220 % XXXXXXXX XXXXXXXX XXXXXXXX
11230 %
11240 SUBROUTINE CORREPAS0010;
11250 BEGIN
11260 STRING SIOND;
11270 %
11280 PAS0010:
11290 WFECH1:= (FECHA*100) + CORR;
11300 TASK008(VALUE=WFECH1);
11310 RUN S136/OBJECT/P008/95MTP022 [TASK008];
11320 IF TASK008 ISNT COMPLETEDOK OR TASK008(VALUE)=1 THEN
11330 BEGIN
11340 SIOND:= ACCEPT("FALLO S136/PAS0008 RETRY ?? SI/NO");
11350 SIOND:= HEAD(TAIL(SIOND," "),NOT " ");
11360 IF SIOND = "SI" THEN
11370 GO TO PAS0010;
11380 END;
11390 WFECH1:= (WFECH1*10) + TIPO;
11400 TASK010(VALUE=WFECH1);
11410 DISPLAY STRING (WFECH1,*);
11420 RUN S136/OBJECT/P010/95MTP036 [TASK010];
11430 OPTION = (BOBASE,FAULT,DSE0,ARRAYS,CODE,FILES,DBS);
11440 FILE LISTMODRFC (FORMMESSAGE = "A-25");
11450 FILE LISTMODACT (FORMMESSAGE = "A-25");

```

```

109700 FILE REPORTE-ALTA (FORMMESSAGE = "A-25"),
109800 FILE IMPRE (FORMMESSAGE = "A-25");
109900 BDNAM = S136/LIST/P010;
110000 %
110100 IF TASK010 ISNT COMPLETEDOK OR TASK010(VALUE)=1 THEN
110200 BEGIN
110300 SIONO:= ACCEPT("FALLO S136/PAS0010 RETRY ?? SI/NO");
110400 SIONO:= HEAD(TAIL(SIONO," "),NOT " ");
110500 IF SIONO = "SI" THEN
110600 GO TO PAS0010;
110700 END;
110800 %
110900 END;
111000 %
111100 PAS0011:
111200 IF CORR = 1 THEN
111300 BEGIN
111400 WFECH11:= (FECHA*100)+CORR;
111500 TASK011(VALUE=WFECH11);
111600 RUN S136/OBJECT/P011/95MTP023[TASK011];
111700 OPTION = (BDBASE,FAULT,USED,ARRAYS,CODE,FILES,BBS);
111800 FILE IMP (FORMMESSAGE = "A-25");
111900 IF TASK011 ISNT COMPLETEDOK OR TASK011(VALUE)=1 THEN
112000 BEGIN
112100 SIONO:= ACCEPT("FALLO S136/PAS0011 RETRY ?? SI/NO");
112200 SIONO:= HEAD(TAIL(SIONO," "),NOT " ");
112300 IF SIONO = "SI" THEN
112400 GO TO PAS0011;
112500 END;
112600 END;
112700 %
112800 %
112900 %
113000 %
113100 %
113200 %
113300 %
113400 %
113500 %
113600 %
113700 %
113800 %
113900 %
114000 %
114100 %
114200 %
114300 %
114400 %
114500 %
114600 %
114700 %
114800 %
114900 %
115000 %
115100 %
115200 %
115300 %
115400 %
115500 %
115600 %
115700 %
115800 %
115900 %
116000 %
116100 %
116200 %
116300 %
116400 %
116500 %
116600 %
116700 %
116800 %
116900 %
117000 %
117100 %
117200 %
117300 %
117400 %
117500 %
117600 %
117700 %
117800 %
117900 %
118000 %
118100 %
118200 %
118300 %
118400 %
118500 %
118600 %
118700 %
118800 %
118900 %
119000 %
119100 %
119200 %
119300 %
119400 %
119500 %
119600 %
119700 %
119800 %
119900 %
120000 %
120100 %
120200 %
120300 %
120400 %
120500 %
120600 %
120700 %
120800 %
120900 %
121000 %
121100 %
121200 %
121300 %
121400 %
121500 %
121600 %
121700 %
121800 %
121900 %
122000 %
122100 %
122200 %
122300 %
122400 %
122500 %
122600 %
122700 %
122800 %
122900 %
123000 %
123100 %
123200 %
123300 %
123400 %
123500 %
123600 %
123700 %
123800 %
123900 %
124000 %
124100 %
124200 %
124300 %
124400 %
124500 %
124600 %
124700 %
124800 %
124900 %
125000 %
125100 %
125200 %
125300 %
125400 %
125500 %
125600 %
125700 %
125800 %
125900 %
126000 %
126100 %
126200 %
126300 %
126400 %
126500 %
126600 %
126700 %
126800 %
126900 %
127000 %
127100 %
127200 %
127300 %
127400 %
127500 %
127600 %
127700 %
127800 %
127900 %
128000 %
128100 %
128200 %
128300 %
128400 %
128500 %
128600 %
128700 %
128800 %
128900 %
129000 %
129100 %
129200 %
129300 %
129400 %
129500 %
129600 %
129700 %
129800 %
129900 %
130000 %

```



***APENDICE 2***

TABLA DE ERRORES

A	ALTA DUPLICADO
B	CLAVE DE ACTIVIDAD ESPECIFICA ERRONEA O NO EN TABLAS
C	CLAVE DE AVAL ERRONEA
D	CLAVE DE BANCA ERRONEA
E	CLAVE DE CUENTA ERRONEA
F	CLAVE DE FONDO ERRONEA
G	CLAVE DE LOCALIDAD DEL CLIENTE ERRONEA
H	CLAVE DE MONEDA ERRONEA
I	CLAVE DE SECTOR ERRONEA
J	CLAVE DE SUBCUENTA ERRONEA
K	CLIENTE SIN NUMERO DE BANXICO O RFC ERRONEO
L	CLIENTE SIN REGISTRO EN EL ARCHIVO MAESTRO
M	CUENTA NO ES 6101 O 6102 Y LLEVA FONDO
N	NOMBRE DEL CLIENTE ERRONEO
O	NUMERO DE CLIENTE - DIGITO ERRONEO
P	SUCURSAL ERRONEA O NO EN TABLAS
Q	TIPO DE CREDITO NO CONSISTENTE CON EL SECTOR



TOTALES FINALES

BAJAS

- HASTA LA CORRIDA ANTERIOR 0
- POR ESTA CORRIDA 0
- HASTA ESTA CORRIDA 0

ALTAS

- HASTA LA CORRIDA ANTERIOR
- ACEPTADOS 0
- RECHAZADOS 0
- TOTAL 0
- POR ESTA CORRIDA 5,785
- ACEPTADOS 5,150
- RECHAZADOS 5,915
- TOTAL 5,915
- HASTA ESTA CORRIDA 5,785
- ACEPTADOS 5,150
- RECHAZADOS 5,915
- TOTAL 5,915

MODIFICACIONES

- HASTA LA CORRIDA ANTERIOR
- ACEPTADOS 0
- RECHAZADOS 0
- TOTAL 0
- POR ESTA CORRIDA 408
- ACEPTADOS 427
- RECHAZADOS 427
- TOTAL 427
- HASTA ESTA CORRIDA 408
- ACEPTADOS 427
- RECHAZADOS 427
- TOTAL 427

MOVIMIENTOS

- HASTA LA CORRIDA ANTERIOR
- ACEPTADOS 0
- RECHAZADOS 0
- TOTAL 0
- POR ESTA CORRIDA 11,887
- ACEPTADOS 10,304
- RECHAZADOS 10,304
- TOTAL 10,304
- HASTA ESTA CORRIDA 11,887
- ACEPTADOS 10,304
- RECHAZADOS 10,304
- TOTAL 10,304

REGISTROS BATCH

- TOTAL HASTA CORRIDA ANTERIOR 1,498
- CRECIDOS EN ESTA CORRIDA 4
- TOTAL SIN ACTUALIZAR 1,492
- TOTAL ACTUALIZADOS 596

ARCHIVO MAESTRO

- TOTAL DE REGISTROS MAESTRO 216,510
- TOTAL DE REGISTROS MAESTRO ECTUAL 222,261
- REGISTROS DE CLIENTES ACTUALIZADOS 66,239
- REGISTROS SIN ACTUALIZAR 151,371

8/15-000/810-012 CIFRAS DE CONTINUA DE OPERACIONES DE CAMBIO PARA BANCO DE GUATEMALA, S.A. 10:55:21 CONTINUA 01 17 4-75 0-00  
 CIFRAS DE CONTINUA AL DIA 30 DE MAY DE 1973 02/12/73

TOTAL GENERAL		MONEDA EXI.		IMPORTE
A MILLES DE PESOS DE		MONEDA EXI.		
MONEDA N.A.C.		MONEDA EXI.		
23	BANCO DE ESTADOS	7,820	16,782	22,522
25	F I R A	3,946,797	80,351	2,027,148
26	BANCO DE	196,646	1,023,015	1,819,061
27	MARIN	3,065,558	23,647	1,087,097
28	F O R A T U R	55,516	19,488	75,004
31	MARIN	67,521	21,842	84,313
32	OTROS FONDOS DE FOMENTO	26,241	163,725	219,966
35	F I R E C	168,413	7,791	168,402
36	F O P M O B A	48,352	259,645	307,797
	TOTAL GENERAL FONDOS	5,418,786	2,196,876	5,012,730

**NUMMEX, S.A.**  
**SUBDIVISION DE CONTABILIDAD CORPORATIVA**  
**ADMINISTRACION DE BASE DE DATOS CONTABLES**

**HOJA DE CUENTA DE RESPONSABILIDADES DE CARTERA PARA BANCO DE MEXICO.**

**PROCESO DE MAYO DE 1993.**  
**CORRIJA NUM. 3**

CUENTA	MONEDA NACIONAL			MONEDA EXTRANJERA		
	BALDOS 8/136	BALDOS 8/114	DIFERENCIA	BALDOS 8/136	BALDOS 8/114	DIFERENCIA
1301	53,715	53,715	0	401,102	401,102	0
1302	0,190,724	0,190,724	0	14,753,321	14,753,321	0
1303	553,313	553,313	0	0	0	0
1304	96,632	96,632	0	0	0	0
1305	7,549,029	7,549,029	0	032,611	032,611	0
1306	91,637	91,635	2	25	24	1
1307	1,367,763	1,367,763	0	47,211	47,211	0
1308	2,300,024	2,300,024	0	647,096	647,096	0
1309	225,041	225,041	0	0	0	0
1310	10,165,251	10,165,251	0	0	0	0
1311	3,322,792	3,322,792	0	0	0	0
1312	10,570,047	10,570,047	0	0	0	0
1314	1,505,004	1,505,004	0	146,159	146,159	0
1315	2,265,092	2,265,092	0	125,347	125,347	0
1317	100,026	100,026	0	1,249	1,249	0
1319	645,353	645,353	0	204,074	204,074	0
1320						
1402	5	5	0	0	0	0
6101	1,391,730		1,391,730	2,310,453		2,310,453
6102	2,266,326	3,650,056	(1,391,730)	533,575	2,052,020	(2,310,453)
6103				166,670	166,670	0
6370	19,023		19,023			
	<b>60,050,007</b>	<b>60,030,942</b>	<b>19,025</b>	<b>20,336,973</b>	<b>17,310,274</b>	<b>2,310,454</b>

**CARATULA DE CONTROL**

**DATOS DEL ARCHIVO INGESTOR**  
 CREADO POR EL PROGRAMA : (S128)SISE/OBJECT/OIG/RINTP01 ON SOBORTABASE.  
 FECHA REAL : 20/04/91  
 HORA : 11:21:47  
 VERSION DE ARCHIVO: 05

**DATOS DEL PROCESO**  
 PROGRAMA : (S128)SISE/OBJECT/180/RINTP02 ON SOBORTABASE.  
 FECHA REAL : 18/10/91  
 HORA : 21:08:04  
 VERSION PROGRAMA : 17/04/91



C/128-000/180-002

RELACION DE RESPONSABILIDADES PARA BANCO DE MEXICO,S.A.

18/10/91 21:08:04 PAG: 13

MILLONES DE PESOS

**DESCRIPCION DE SALDOS EN CTAS,**

1001	840001
1002	1972289
1003	802487
1004	11138
1005	8127438
1006	84558
1007	128200
1008	1182297
1009	71773
1010	971468
1011	1023512
1012	9424824
1013	
1014	718153
1015	718878
1016	
1017	30427
1018	
1019	72823
1020	1031957
1021	1803710

<b>ARCHI-SALDOS CARTERA VENC.VIG.</b>	
SALDO CARTERA VENCIDO INT.	1331483
CLIENTES CARTERA VENCIDO INT.	12084
SALDO CARTERA VENCIDO EXT.	105014
CLIENTES CARTERA VENCIDO EXT.	123
SALDO CARTERA VICENTE INT.	28221428
CLIENTES CARTERA VICENTE INT.	13182370
SALDO CARTERA VICENTE EXT.	1818
CLIENTES CARTERA VICENTE EXT.	30034010

<b>TOTALES GENERALES ARCHI-SALDOS</b>	
SALDO TOTAL ARCHI-SALDOS	
REGISTROS TOTALES ARCHI-SALDOS	33017

<b>ARCHI-CLIENTES CARTERA VENCION</b>	
SALDO TOTAL EN CARTERA VENCION	532280



R/135-000/010

CORRIDA

BANAMEX  
SUBDIVISION CORPORATIVA DE CONTABILIDAD

RELACION DE RESPONSABILIDADES DE CARTERA PARA BANCO DE MEXICO  
LISTADO DE RECHAZOS POR CLIENTE AL 30 DE SEPTIEMBRE DE 1993

4/10/93

21:35:36

MILES DE N. PESOS M0

MON	SUCURSAL	No CLIENTE	NOMBRE DEL CLIENTE	CUENTA	SUBCUENTA	FONDO	SALDO	N.F.C.	LOCALIDAD	ACTIVIDAD
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	2101			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1304		40	125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	999999	0211013
3	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1315	14		125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	999999999	201105	0211013
2	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1300		25	125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	999999999	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	999	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013
1	2	2100700041	MALDONADO, RADILLO/GABRIEL	1302			125	MARAC51201	201105	0211013

## CARATULA DE CONTROL

CUENTA	N. NACIONAL	MON. EXT.	TOTAL
1301			
1302	1,500	125	1,625
1303			
1304			
1305			
1306	250		250
1314			
1315	125		125
6103			
TOTAL	1,875	125	2,000

CUENTA	MON. NDL	MON. EST	TOTAL. GRAL.
0100	10.000	10.000	20.000
0101	70.000	7.000	77.000
0102	10.000	770	10.770
0103	10.000		10.000
0104	000		000
0105	000	110	0110
0106	000	20.000	20.000
0107	000	7.100	7.100
0108	000.000	77.000	777.000

REC 0108 LEICOS MONEXES (AUMENTO) 001



RELACION DE OPERACIONES DE CUENTAS PARA BANCO DE MEXICO S.A. 10/03/99  
 CONSIDERACIONES POR CLIENTE DE LA UNIDAD BANCA EL 10/03/99 DE 1993  
 HILEN DE N. PERCE N.3

REC 0108 LEICOS MONEXES : 0.000  
 IMPORTE TOTAL MONEXES : 11.700.000  
 IMPORTE TOTAL MONEXES Y MONEXES : 11.701.701



**APENDICE 3**

4 January 1994

Report Painter REC20MRCOG

R/S000-000/000-000

PAGINA : \_\_\_\_\_

BANCO NACIONAL DE MEXICO S.A.  
RESPONSABILIDADES DE CARTERA

FECHA DE PROCESO : \_\_\_\_\_

FECHA REAL \_\_\_\_\_

HORA REAL \_\_\_\_\_

CIERRE DE RESPONSABILIDADES DE CARTERA

MONEDA NACIONAL			MONEDA EXTRANJERA			
CTA	SALDOS S/136	SALDOS S/114	DIFERENCIA	SALDOS S/136	SALDOS S/114	DIFERENCIA

TOTAL \_\_\_\_\_



19 October 1993

Screen Painter TBT22

BANCO NACIONAL DE MEXICO S.A. RESPONSABILIDADES DE CARTERA GLOBAL CUENTA-MONEDA (RIESGOS CON ERROR Y OK)				FECHA-----
CUENTA	MONEDA NACIONAL	MONEDA EXTRANJERA	TOTAL	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	
--	-----	-----	-----	

OPCION-  
N=SIG.HOJA S=SALIR M=MENU PRINCIPAL

6 January 1994

Screen Painter TBT22

-----			FECHA----
SUCURSAL ----		BANCO NACIONAL DE MEXICO S.A. RESPONSABILIDADES DE CARTERA TOTAL POR SUCURSAL CUENTA-MONEDA	
NO. CUENTA	MONEDA NACIONAL	MONEDA EXTRANJERA	TOTAL
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----
----	-----	-----	-----

OPCION---  
N=SIG.HOJA S=SALIR M=MENU PRINCIPAL



6 January 1994

Report Printer REC11MRBAM

R/S000 000/000 000

BANCO NACIONAL DE MEXICO S.A.  
RESPONSABILIDADES DE CARTERA

PAGINA : \_\_\_\_\_  
FECHA DE PROCESO : \_\_\_\_\_

FECHA REAL \_\_\_\_\_  
HORA REAL \_\_\_\_\_

CARATULA DE TOTALES POR CUENTA DE LA CINTA BANXICO

CUENTA	MAYORES			MENORES			TOTAL		
	MON.NAL	MON.EXT	TOTAL	MON.NAL	MON.EXT	TOTAL	MON.NAL.	MON.EXT.	TOTAL

REGISTROS LEIDOS MENORES: \_\_\_\_\_ REGISTROS LEIDOS MAYORES: \_\_\_\_\_  
IMPORTE TOTAL MENORES: \_\_\_\_\_ IMPORTE TOTAL MAYORES: \_\_\_\_\_  
TOTAL MAYORES Y MENORES: \_\_\_\_\_

4 January 1994	Screen Painter LPC02	
-----	----	FECHA----
	BANCO NACIONAL DE MEXICO S.A.	
	RESPONSABILIDADES DE CARTERA	
	DESCRIPCION DE SALDOS EN CUENTAS	
CUENTA	SALDO	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
----	-----	
		OPCION---
		N=SIG.HOJA S=SALIR M=MENU PRINCIPAL

4 January 1994

Screen Painter LPC25

-----  
BANCO NACIONAL DE MEXICO S.A.  
RESPONSABILIDADES DE CARTERA  
ARCHI-SALDOS CARTERA VENCIDA Y VIGENTE

FECHA----

SALDO CARTERA VENCIDA NAL.	-----
CLIENTES CARTERA VENCIDA NAL.	-----
SALDO CARTERA VENCIDA EXT.	-----
CLIENTES CARTERA VENCIDA EXT.	-----
SALDO CARTERA VIGENTE NAL.	-----
CLIENTES CARTERA VIGENTE NAL.	-----
SALDO CARTERA VIGENTE EXT.	-----
CLIENTES CARTERA VIGENTE EXT.	-----

TOTALES GENERALES ARCHI-SALDOS	
SALDO TOTAL ARCHI-SALDOS	
REGISTROS TOTALES ARCHI-SALDOS	-----

ARCHI-CLIENTES CARTERA VENCIDA	
SALDO TOTAL EN CARTERA VENCIDA	-----

N=SIG.HOJA S=SALIR M=MENU PRINCIPAL

4 January 1994		Screen Painter LPC26		
		BANCO NACIONAL DE MEXICO S.A.		FECHA----
		RESPONSABILIDADES DE CARTERA		
		CIFRAS DE CONTROL POR FONDOS		
TOTAL GENERAL				
NO.				
FONDO	NCMBRE DE FONDO	M.N.	M.E.	IMPORTE
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
---	-----	-----	-----	-----
OPCION --- N=SIG.HOJA S=SALIR M=MENU PRINCIPAL				