UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES "ACATLAN"

COMPARACION EN DESARROLLO DE SISTEMAS, LENGUAJES DE TERCERA Y CUARTA GENERACION

T E S I S

QUE PARA OBTENER EL TITULO DE:
LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACION
P R E S E N T A :
MARCO AURELIO REYNA PADILLA

ASESOR: ING. RUBEN ROMERO RUIZ

UNAM
CAMPUS ACATLÁN

SANTA CRUZ ACATLAN, NAUCALPAN, EDO. DE MEX. NOVIEMBRE 1999

COMPARACION EN DESARROLLO DE SISTEMAS, LENGUAJES DE TERCERA Y CUARTA GENERACION

AGRADECIMIENTOS

A Dios y a la U.N.A.M

Proveedores invaluables de medios para la realización de este trabajo.

Al Ing. Rubén Romero y a mis sinodales

Por la atención a este trabajo y por todas las muestras de apoyo que he recibido durante todos mis años como estudiante.

A mis Padres y Hermanos

Mi ejemplo más cercano de superación y primera fuente de amor.

A Carmen Aguilar

Por el tiempo que hemos vivido juntos y lo que la vida me ha dado a tu lado.

A Jazmín Xirayo

Desde tu llegada la ingenuidad y la espontaneidad son dos amigos que constantemente aparecen en mi vida.

CONTENIDO

INTRODUCCION

La evolución de los lenguajes de programación está fuertemente influenciada por la necesidad de desarrollar sistemas en forma más rápida, la complejidad de requerimientos y el cambio continuo del medio ambiente de los sistemas demandan tecnologías de desarrollo más poderosas.

La transición entre las diversas generaciones de los lenguajes de programación ha tendido, por un lado a eliminar la necesidad de conocer la forma en la que operan las computadoras, y por otro, a generar lenguajes cuyo código sea más poderoso; permitiendo que una instrucción de un lenguaje sea equivalente a varias instrucciones de los lenguajes precedentes. Esto acelera el proceso de desarrollo de sistemas.

Los lenguajes de primera y segunda generación son los lenguajes en código máquina y ensambladores respectivamente, su aparición data de las decadas de los 50´s y 60´s. En estas generaciones la arquitectura de las computadoras proporcionaba capacidades de procesamiento y de almacenamiento muy pobres, requiriendo espacios físicos muy grandes y muy acondicionados para operar correctamente.

El código máquina es un lenguaje de interpretación directa para la computadora mediante el encendido y apagado de circuitos. Algunas de sus características son:

- un mínimo requerimiento de espacio de almacenamiento al no requerir compiladores para realizar la traducción de código (el código es alimentado e inmediatamente procesado)
- Da rapidez durante la ejecución pues al residir en la memoria evita demoras por búsquedas de código en otros

dispositivos de almacenamiento. Sin embargo, la dificultad en la interpretación de este código hace complejo el desarrollo de sistemas, aún de mediana extensión, así como su mantenimiento.

Los lenguajes ensambladores están constituidos por un conjunto de instrucciones en forma de código mnemónico. Si bien estos lenguajes son más comprensibles que los lenguajes máquina (haciendo más fácil para el programador el desarrollo y mantenimiento de programas), aún no poseen características que favorezcan el desarrollo de sistemas (definición de tipos de datos, estructuras de datos, estructuras de control, etc.)

A mitad de la década de los 60´s el ingreso de una nueva arquitectura de computadoras basada en circuitos integrados abre nuevas perspectivas a la ciencía de la computación. Con una mayor capacidad de procesamiento y de almacenamiento de información se desarrollan los lenguajes de tercera generación; los cuales están orientados a favorecer el diseño, desarrollo y mantenimiento de sistemas de más largo alcance. Las características de estos lenguajes son las siguientes:

Son modulares, por lo que contienen una metodología que permite manejar un orden en los programas, con estos lenguajes se puede hacer un estudio del sistema en su totalidad y encontrar un conjunto de componentes que lo definan, mediante un procedimiento iterativo. Estos componentes serán traducidos en los procedimientos que constituyen el sistema informático.

Los lenguajes de tercera generación hacen uso de las estructuras de datos, establecen una tipificación de datos y operaciones

válidas sobre éstos; automatizando así operaciones que anteriormente eran responsabilidad del programador.

Los lenguajes de tercera generación motivan al programador a eliminar el uso de la instrucción (GOTO) mediante la definición de estructuras de control (IF - THEN - ELSE, CASE - OF, REPEAT - UNTIL, DO - WHILE). Estas estructuras hacen más claro el desarrollo de sistemas y facilitan su mantenimiento, pues permiten estructurar mejor las operaciones a realizar de acuerdo a la situación que se presente durante el procesamiento (ejecución o no de un bloque de instrucciones de acuerdo a una decisión. Ejecución de un bloque de instrucciones como resultado de una decisión que se realizo antes o después de ejecutar al bloque de instrucciones).

La recursividad es otra estructura de control que aparece con estos lenguajes y permite a un programa llamarse a sí mismo iterativamente mediante la evaluación de una condición.

En la década de los 80´s la aparición de arquitecturas de computadoras basadas en circuitos densamente integrados proporciona nuevas perspectivas en el campo de la computación, las redes de computo y los sistemas de administración de base de datos son aplicaciones que se hacen comunes en esta época. Los Sistemas de Administración de bases de datos dan un nuevo valor a la información; ésta ya no es más un conjunto de archivos diseñados para satisfacer las necesidades de alguna(s) aplicación(es) y para su uso exclusivo. Ahora la información es diseñada y almacenada para ser compartida por todas las aplicaciones que están relacionadas con ella y existe en forma

independiente a las últimas. Esto proporciona enormes beneficios como son minimización de redundancia y maximización del uso de la información. Nuevos aspectos de administración de la información, surgen, tales como: establecimiento de seguridad de la información, esquemas de procesamiento distribuido y control de acceso concurrente. Los lenguajes de cuarta generación son un producto que aparece en esta época.

El diccionario de datos es una herramienta del sistema de administración de base de datos que permite registrar características de la información que está almacenada en la base de datos. Este es manejado por un sistema de Diccionario de Datos, cuya función es hacer accesible la información contenida en la base de datos. Las herramientas del lenguaje de cuarta generación utilizan el sistema de diccionario de datos para generar productos para el programador que, con lenguajes precedentes eran realizadas por los programadores consumiéndoles gran parte del tiempo de desarrollo.

El generador de pantallas obtiene información de los atributos que constituyen las tablas y las relaciones existentes entre éstas del sistema de diccionario de datos. Con esta información permite generar pantallas que serán utilizadas para mantener actualizada la información registrada en la base de datos; mediante el uso del diccionario, el desarrollador ahorra gran parte del código que tenía que elaborar con lenguajes de generaciones anteriores, para realizar las operaciones de ingreso y consulta de información.

El lenguaje de consulta obtiene información del diccionario de datos para procesar la información de la base de datos. Permite realizar consultas y actualizaciones sobre conjuntos de registros de alguna(s) tabla(s), estas operaciones son especificadas mediante condiciones relacionales.

Para ejecutar los comandos del lenguaje no es necesario considerar la forma en la que se encuentran organizadas las tablas, ni la forma en la que serán procesados los registros, la aplicación de estos comandos durante la programación reduce substancialmente el tiempo y esfuerzo dedicado para la generación de programas.

El generador de reportes obtiene información de las tablas mediante el sistema de diccionario de datos, el diseño de un reporte está sujeto a la estructura de las tablas que lo constituyen. Una vez que son designadas las tablas, el trabajo del usuario se reduce a situar campos, etiquetas, imágenes y líneas en un esquema. El generador presentará la información de las tablas implicadas de acuerdo al esquema proporcionado. El uso del generador tiene un amplio impacto en la presentación de información, sobre todo en el ambiente administrativo de las organizaciones, donde permite de una forma muy eficiente la presentación de información. La velocidad con que se generan los reportes, da posibilidad a un amplio análisis de la información mediante la presentación de ésta en formas muy variadas y muy complejas.

Algunas de las funciones que realiza el generador de reportes son: Salto automático de hoja, establecimiento de secciones

(niveles de ordenación) de la información que se presenta en el reporte, impresión automática de encabezados y pies de sección, definición de campos calculados basados en expresiones; así como establecimiento de funciones sumarias (suma, promedio, conteo, etc.) dentro de las secciones que constituyen el reporte y presentación ordenada de la información cuando se relacionan dos o más tablas en un reporte; presentando los registros de la primera en forma ordenada y los de la segunda tabla en correspondencia a la relación que tenga con la primera.

El generador de programas es una herramienta que permite la integración de los productos generados con las herramientas anteriormente mencionadas para constituir un sistema funcional. Algunas funciones que realiza el generador de programas son: desarrollo de menús y asociación de un producto (forma, reporte o programa) a una opción del menú, mediante la integración de código al sistema.

El objetivo de este trabajo es explicitar los fundamentos de un lenguaje de cuarta generación y mostrar los avances que este proporciona con respecto a los lenguajes de tercera generación. Para realizar esto, se lleva acabo el diseño y desarrollo de un sistema con cada uno de estos lenguajes, esperando que en este proceso se despejen dudas del usuario con poca experiencia en el uso de los lenguajes de cuarta generación.

El desarrollo de sistemas mediante la tecnología orientada a objetos constituye una nueva tendencia cuyo estudio supera los objetivos de este trabajo. En esta tecnología la atención principal se les da a los objetos (módulos autocontenidos que

contienen los datos y los procedimientos que actuán sobre ellos). Los objetos en este caso son entes activos -a diferencia de los datos de lenguajes de generaciones anteriores- que interactuan entre ellos mediante el envío de mensajes.

De acuerdo a su utilidad los objetos pueden ser agrupados en clases y pueden crearse nuevas clases, las cuales hereden los procedimientos y datos de las clases existentes para satisfacer nuevos y cambiantes requerimientos. Esta tecnología está diseñada para auxiliar el desarrollo de sistemas de gran extensión, así como para maximizar la utilización de los programas que genera.

I. HERRAMIENTAS DE UN LENGUAJE DE CUARTA GENERACION

Los lenguajes de cuarta generación son denominados "no procedurales" en el sentido de que poseen herramientas para dar respuesta directa a los requerimientos del usuario. Estas generan productos que con lenguajes de generaciones anteriores eran elaborados mediante la producción de programas. En forma general estas herramientas se pueden catalogar en las siguientes :

I.1 SISTEMA DE ADMINISTRACION DE BASE DE DATOS

Los lenguajes de cuarta generación (4 GL) incluyen un conjunto de programas para manejar la información del sistema. Este conjunto de programas es conocido como Sistema de Administración de Base de Datos (SABD), o como sistema de administración de archivos.
El SABD otorga más prestaciones pues siempre mantiene una visión global de la información; esté considera las relaciones existentes entre las tablas, para evitar que se pierda la consistencia de la información durante las actualizaciones que sobre la base de datos se hagan " Para protegerse contra corrupciones accidentales o maliciosas de los datos, se pueden definir un conjunto de restricciones de integridad. Estas definen los estados y transiciones válidas de la base de datos, permitiendo así al sistema rechazar actualizaciones inválidas de los datos " (S. Misbah Deen,1990, Pág. 30).
Los lenguajes de cuarta generación consideran a la información contenida en la base de datos como el corazón de todo sistema de

procesamiento de datos, por lo que proporcionan herramientas para realizar los procesos que se realizan sobre ella. para el registro e ingreso de información utilizan al Sistema de administración de base de datos y al generador de pantallas, para el procesamiento de información y generación de resultados al lenguaje de consulta Query y al generador de reportes "Algunas veces el SABD proporcionará utilerías para la generación de programas y sistemas. Estas pueden ser usadas para construir sistemas más complejos, los cuales contienen significativas herramientas de procesamiento. Estas características están normalmente asociadas con los sistemas de cuarta generación" (Avison, 1985, Pág. 125).

En el caso de los sistemas de administración de base de datos, la base de datos es creada mediante la compilación de la especificación dada con el diccionario de datos. La compilación crea las tablas, asigna espacio a éstas, genera estructuras de datos y proporciona subrútinas para acceso y recuperación de información. El sistema de administración de base de datos tiene adicionalmente utilerías para optimizar la operación de la base de datos, como reconstrucción de índices y recuperación de espacio ocupado por registros suprimidos.

Una de las principales características de un SABD es que permite acceso compartido a los datos por múltiples transacciones -permitiendo así el acceso concurrente a la información- . El SABD controla la ejecución de las transacciones para garantizar que múltiples usuarios accesen valores consistentes de la base de dato "Al fin de una transacción la base de datos debe quedar en

un estado consistente, pero durante su ejecución una transacción puede provocar que la base de datos sea inconsistente. Por lo que es requerido que el sistema ejecute totalmente la transacción o no la ejecute en su totalidad". (S. Misbah Deen,1990, Pág 31).

I.2 DICCIONARIO DE DATOS

El diccionario de datos contiene la especificación de la base de datos, esta especificación es almacenada en una base de datos, por lo que se considera que el diccionario de datos es una base de datos de la base de datos. Existen diferentes tipos de diccionarios de datos los cuales son catalogados de diversas formas. Por ejemplo, de acuerdo a su funcionalidad o a la relación que lleven con el Sistema de administración de Base de Datos.

Según la relación que tengan con el Administrador de la base de datos un diccionario de datos puede considerarse independiente si tiene capacidad de trabajar en forma separada del SABD, de otra forma se dice que el diccionario es integrado o dependiente; en esta situación el diccionario de datos opera conjuntamente con el SABD.

Otra forma de categorizarlos es en activos o pasivos, un diccionario pasivo es solo un medio de documentación del sistema, la información puede alimentarse y dársele mantenimiento en el diccionario, la información después es reportada en diferentes

formas que son útiles para los usuarios del SABD. Sin embargo, no existe interacción entre el diccionario y los demás componentes. Un diccionario activo además de ser una herramienta de documentación, interactua con los demás componentes, sirviendo entre otras cosas como un verificador de las funciones que realizan los demás componentes, desempeñando así una función más productiva.

I.2.1 FUNCIONES DEL DICCIONARIO DE DATOS

El diccionario de datos puede almacenar información relacionada con la base de datos a nivel dato, proceso o medioambiente. A nivel dato además de registrar las características de estos (Longitud, tipo, representación, valores default, etc.), puede registrar sinónimos o alias con los que es conocido el dato por diferentes usuarios, así como diferentes nombres por los que el atributo puede ser conocido por diferentes lenguajes de programación, resolviendo de esta forma multiplicidades que se pudieran presentar con el uso de la información.

Basado en los archivos, datos, procesos y las operaciones que los archivos y procesos realizan sobre los datos, el diccionario de datos construye un modelo, con este modelo automatiza funciones tales como la documentación de los sistemas. De esta manera esta función es más precisa que como se realizaba con lenguajes de generaciónes anteriores; en los cuales la documentación dependía

totalmente del factor humano, era difícil de actualizar y de corregir; razón por la cual era frecuentemente abandonada.

En el desarrollo y mantenimiento de sistemas, el diccionario de datos nos permite conocer las implicaciones que puede tener un cambio en la base de datos o en las operaciones realizadas por los procesos que constituyen al sistema. Esto incrementa la productividad de estas funciones comparadas a la forma en que se realizaban con los lenguajes de generaciones previas, en los últimos no se contaba con estas herramientas y esto frecuentemente implicaba que un cambio que se realizaba, producía efectos inesperados en diversos módulos del sistema.

Para satisfacer las necesidades de los usuarios del medioambiente de la base de datos -y éstos incluyen a los equipos de administración de la base de datos, programadores, analistas, diseñadores y otros del departamento de sistemas- el sistema de diccionario de datos debe ser capaz de producir una gran variedad de reportes, los cuales incluyen reportes estandar, así como reportes generados por los usuarios. Entre los reportes estandar se encuentran :

- Listas de todos los campos de la base de datos, sinónimos y nombre del responsable del mantenimiento de éstos.
- Reportes de las relaciones entre las tablas y los campos contenidos dentro de cada relación
- Reportes detallando todas las características de los campos, registros, vistas de usuario, programas o cualquier otro elemento almacenado en el diccionario de datos.

- Reportes de referencias cruzadas mostrando, por ejemplo, todas las relaciones, pantallas, reportes, programas, sistemas y usuarios relacionados a un campo. Estos reportes son muy valiosos para realizar el mantenimiento de un sistema.

I.2.2 INTERACCION CON LAS HERRAMIENTAS DEL LENGUAJE DE CUARTA GENERACION

El sistema de diccionario de datos proporciona información para que sea utilizada por los demás elementos del medioambiente. Las formas en las que el sistema de diccionario de datos puede generar información para los demás elementos son las siguientes :

- El sistema de diccionario de datos genera información a partir del diccionario de datos, esta información sirve de entrada para las operaciones de los demás componentes del lenguaje. Este tipo de esquema es llamado "utilería puente" o "interfase estática", ya que la interacción ocurre antes de la ejecución del SABD, el sistema de diccionario de datos genera archivos que deben ser integrados con las herramientas del 4GL; en este caso las modificaciones que se realizan durante la ejecución de algún componente pueden no reflejarse inmediatamente en los archivos generados por el sistema de diccionario de datos, surgiendo así la posibilidad de definiciones inexactas.

- El sistema de diccionario de datos genera información en el sentido inverso, en este caso el sistema de diccionario de datos extrae definiciones de los programas y de la base de datos y basado en estas definiciones alimenta al diccionario de datos.
- El sistema de diccionario de datos esta totalmente integrado con el SABD. Todas las definiciones de los datos y documentación del sistema son hechas en el diccionario de datos. El SABD obtiene sus definiciones directamente del diccionario de datos. Debido a que la interacción entre el diccionario de datos y el SABD ocurre durante la ejecución del SABD, cualquier cambio realizado al diccionario de datos es reflejado automáticamente en el SABD. Esto asegura la consistencia de la definición de datos en todo el medioambiente.

I.3 GENERADOR DE PANTALLAS

Los generadores de pantallas permiten elaborar formas para ingreso y actualización de información en la base de datos. La elaboración es realizada normalmente mediante el diseño de un esquema que constituirá la pantalla , la labor del usuario del generador consiste en situar texto, indicar los campos y tablas que serán actualizados mediante la forma, así como realizar algunos detalles para mejorar la apariencia de la pantalla

(situar líneas, colores, imágenes, etc.). El usuario indica la información que se actualiza mediante la forma y el generador produce la pantalla, encargándose de la generación del código necesario para hacerla funcional, según las especificaciones dadas por el usuario.

El generador no da respuesta a todas las necesidades de procesamiento que surgen durante el ingreso de información a la base de datos, para satisfacer las necesidades particulares de procesamiento que surgen durante este proceso (actualización de contadores, actualización de archivos maestros apartir de archivos detalle, cálculos de funciones, así como algunas otras operaciones propias del proceso que se está realizando), el lenguaje de cuarta generación incluye instrucciones para manejar a la forma, permitiendo que mediante el uso de su lenguaje de programación se refine la operación de ésta.

Con un lenguaje de tercera generación en el desarrollo de una pantalla se controla el ingreso de datos a los campos, conforme se ingresa información a los campos de la pantalla, así como cuando se termina de registrar la información, se realizan algunos otros procesos requeridos mediante instrucciones del lenguaje. De esta forma con un lenguaje de tercera generación el ingreso de información se encuentra mezclado con el código requerido para realizar las funciones propias de la forma.

El 4GL genera el código para controlar el ingreso de información, y este código queda oculto para el usuario. Para permitir realizar operaciones propias del sistema, el lenguaje de cuarta generación maneja el concepto de evento; el cual está asociado a

lo que sucede en la pantalla durante la ejecución de una forma, de acuerdo a estos sucesos permite al 4Gl ejecutar operaciones definidas por el usuario.

Los eventos que normalmente maneja el 4GL suceden como resultado de teclas presionadas por el usuario o de acciones realizadas mediante el mouse, algunos de estos eventos son:

A nivel campo:

BEFORE FIELD nombre_del_campo .- Permite codificar operaciones a realizarse antes de que el cursor alcance el campo nombre_del_campo. Ejemplos de requerimientos de este tipo son asignación de un número de folio a alguna entidad como una factura, una póliza, etc.

AFTER FIELD nombre_del_campo .- Permite codificar operaciones a realizarse cuando el cursor sale del campo nombre_del_campo. Ejemplos de requerimientos de este tipo son verificación y recuperación de la información de alguna entidad en un archivo como una factura, póliza, etc.

ON FIELD INPUT.- Permite codificar operaciones a realizarse mientras se está ingresando información en un campo, un ejemplo de requerimiento de este tipo sería detectar cuando el usuario presiona una tecla en particular.

A nivel registro :

BEFORE ADD .- Permite realizar operaciones antes de agregar un registro a la base de datos

AFTER ADD .- Permite realizar operaciones después de agregar un registro a la base de datos.

Algunos otros eventos son :

BEFORE FIND .- Algunos 4Gl automatizan las operaciones de búsqueda de registros en la base de datos, llevándolas a cabo de acuerdo a relaciones establecidas por el usuario en la forma; estas relaciones se indican durante la ejecución del sistema, cuando sucede esto el 4GL incorpora este evento. Este permite realizar operaciones antes de que se empiecen a recuperar registros de la base de datos. Un ejemplo sería inicializar un acumulador a cero para sumar los valores sobre un campo, considerando para la suma los registros que satisfagan la relación establecida.

ON FIND .- Permite realizar operaciones para cada registro que se recupere de la base de datos de acuerdo a la petición realizada por el usuario de la forma. Un ejemplo sería acumular el valor de un campo de los registros recuperados a una variable de acumulación.

AFTER FIND .- Permite realizar operaciones después de que se recuperaron un conjunto de registros de la base de datos.

Actualmente existen algunos lenguajes de cuarta generación desarrollados bajo la tecnología orientada a objetos. Tal es el caso de los lenguajes de desarrollo bajo el medioambiente Windows, como el VisualBasic. Estos lenguajes poseen las herramientas que estamos mencionando; pero al tener una alta orientación gráfica, brindan mayores prestaciones para el desarrollo de pantallas que las que hemos mencionado. Algunas de las prestaciones que ofrecen son :

- Un amplio número de objetos gráficos para una mejor operación de la pantalla mediante el uso del mouse. Tal es

el caso de los controles tridimensionales que permiten realizar funciones como ejecutar una operación o salir de una pantalla, estas funciones se realizan ubicando el cursor sobre el control y presionando un botón del mouse; algunos otros controles son las barras de desplazamiento, controles gráficos para administración de dispositivos de almacenamiento, etc.

- Una variedad de objetos para captura de información como cajas de texto, cajas de texto con máscaras, Listas de texto (combo boxes), etc.

  Estos objetos poseen sus propios eventos y pueden ser programados para adecuarlos a las necesidades del desarrollador.

I.3.1 INTERACCION CON EL DICCIONARIO DE DATOS

El generador de pantallas utiliza la información contenida en el diccionario de datos para automatizar la generación de las formas y realizar algunas funciones durante la ejecución de éstas. Algunos de los puntos en los que el generador interactua con el diccionario de datos son :

- Obtiene información del diccionario de datos para determinar las relaciones existentes entre las tablas, de esta forma evita el ingreso de información que pudiera propiciar una pérdida de integridad de la información en la

base de datos, ejemplo de esto sería evitar que se suprimiera el registro de un cliente del cual aún existen facturas.

- Obtiene información de las características de los campos existentes en las formas, para realizar las siguientes operaciones con la información que el usuario está ingresando :
  - validación del tipo de información durante su ingreso
  - verificación de que el valor tecleado no exceda máximos y mínimos permitidos para el campo
  - máscaras para formateo de información
- Establecimiento de la cardinalidad entre un archivo maestro y sus archivos detalle cuando han sido relacionados en una forma (uno a uno, uno a muchos, muchos a uno o muchos a muchos). La determinación de estas relaciones permite a su vez definir la siguiente función.
- Creación y manejo de regiones multiregistro. Cuando una tabla maestra y una detalle aparecen en una forma, el generador permite visualizar múltiples registros de un archivo de detalle y automatiza funciones como el recorrido de registros, permitiendo que el desarrollador se despreocupe por operaciones como presentación parcial de registros, esta función es necesaria cuando la tabla de detalle posee más registros de los que pueden ser visualizados en la forma en un momento dado.

I.4 LENGUAJES DE CONSULTA

Con los lenguajes de tercera generación para realizar el proceso de datos normalmente se ordenan los archivos de entrada -archivos sobre los que se operará para obtener nueva información-, se procesan estos archivos en forma ordenada, conforme se va realizando el proceso de los archivos se realizan los cálculos necesarios para generar nueva información y se registran los resultados. Este ciclo es realizado de esta forma debido a la pobre organización de la información que proporcionan los lenguajes de tercera generación, lo que obliga al programador a realizar el proceso de las tablas para transformar la información.

Los lenguajes de consulta permiten la recuperación y transformación de la información contenida en la base de datos mediante comandos compactos, los cuales establecen condiciones relacionales sobre campos de las tablas, extrayendo información de aquellos registros cuyos valores satisfagan la relación especificada.

Para realizarlo los lenguajes explotan la organización de la información que proporciona el SABD, del diccionario de datos extraen la estructura de las tablas y validan la sintaxis del comando de consulta, mediante el SABD extraen la información solicitada en forma de registros, de acuerdo a la relación indicada por el usuario.

Los comandos del lenguaje de consulta extraen conjuntos de registros de la base de datos mediante operaciones definidas en base al cálculo relacional o al álgebra relacional, de estos conjuntos de registros extraidos se pueden realizar transformaciones para generar nueva información en la base de datos. Estas características permiten al programador reducir su esfuerzo para generar nueva información. Sin embargo, el programador debe acoplar sus necesidades de programación a las operaciones definidas por los comandos del lenguaje de consulta. Para resolver las necesidades de programación que excedan a estos operadores el programador puede usar el lenguaje de programación del 4GL . "Un sistema relacional organiza los datos en una base de datos, de acuerdo al modelo de datos relacional. En adición, proporciona un lenguaje de datos relacional para accesar una base de datos. El lenguaje de datos proporciona utilerías capaces de emular los operadores relacionales" (Tsichritzis,1977, Pág. 186)
Dado que el lenguaje de consulta permite operar sobre un conjunto de tablas mediante un simple comando, constituye una herramienta no-procedural, ya que elimina la necesidad de que el usuario instruya a la computadora sobre la forma como deben ejecutarse las operaciones sobre las tablas, limitándose a indicarle a la computadora qué operaciones deben realizarse.
Las operaciones que pueden realizarse mediante el lenguaje de consulta son las siguientes:
Para recuperación de información :
select lista_de_campos from tabla(s) where condición

donde :

lista_de_campos.- indica aquellos campos que se muestran como resultado de la consulta

tabla(s).- indica las tablas de las cuales se extrae la información

condición .- establece la condicion relacional bajo la cual se extrae información de la base de datos

Ejemplo.-

select nombre,puesto,edad from empleados where depto="ventas"

Este query extrae el nombre,puesto y edad de aquellos empleados que pertenezcan al departamento de ventas.

El lenguaje de consulta permite realizar algunos cálculos sumarios sobre la información contenida en la base de datos tales como acumulados, promedios, conteos, máximo y mínimo.

ejemplo .-

select sum(importe) from factura where cliente="0001"

Este comando acumula el monto de las facturas que pertenecen al cliente 0001

Para ingreso de información :

insert into nombre_de tabla lista_de_valores

donde :

nombre_de_tabla .- Es la tabla a la que se le anexarán registros

lista_de valores .- valores que se le asignarán a los atributos del nuevo registro que se anexó.

Adicionalmente el lenguaje de consulta permite realizar cálculos sumarios sobre la información contenida en las tablas de la base

de datos y con estos cálculos generar nueva información, por ejemplo:

consideremos las siguientes tablas :

Facturas	Det-factura
No_factura	No_factura
Cliente	Cliente
Importe	producto
	importe

una forma de actualizar el maestro de facturas de acuerdo al archivo de detalle es:

insert into facturas select No_factura, Cliente, sum(importe) from Det-factura Group By No_factura

Este comando acumula el importe de los productos que constituyen una factura y genera un registro por factura en el archivo maestro. En este ejemplo se realiza un cálculo sumario sobre el importe de las facturas, para realizar este cálculo se agrupan los registros mediante la cláusula Group By para aquellos registros que tengan el mismo valor en el campo No_factura.

Para actualización de la información :

Update nombre_de_tabla set nombre_de_campo=expresión where condición

Donde :

nombre_de_tabla.- Nombre de la tabla que es actualizada

nombre_de_campo.- determina que campo es actualizado

expresión .- expresión que determina que valor se le asigna al campo

condición .- condición relacional que indica cuales registros son actualizados.

Estas características dan posibilidades muy valiosas al lenguaje de consulta al permitir simplificar las operaciones de ingreso y actualización de información. Como se ha visto con los comandos de consulta se realizan operaciones sobre las tablas, esto evita realizar el recorrido de registros, revisión de la condición sobre el registro para determinar si se procesa el registro, elaboración de cálculos intermedios y registro de la nueva información.

Adicionalmente el lenguaje de consulta permite establecer condiciones sobre cualquier campo de la tabla, así como sobre cualquier combinación de campos de ésta, esto establece una diferencia fundamental con los lenguajes de tercera generación, en los cuales las tablas sólo pueden ser procesadas de acuerdo al orden que establece el índice.

Query by example. Es un lenguaje de consulta que no requiere del uso de comandos para realizar operaciones, el usuario emplea un esquema de las tablas que va a consultar y con el uso de algunos operadores puede obtener la información que requiere.

I.5 GENERADOR DE REPORTES

El generador de reportes permite elaborar reportes basados en la información almacenada en la base de datos. El usuario diseña el

reporte mediante la selección de las tablas y campos que aparecen en el reporte, una vez seleccionadas, el usuario define un esquema que es utilizado por el generador para producir el reporte.

El generador ejecuta de forma automática funciones tales como consulta a la base de datos, control de salto de página, definición de márgenes, impresión de encabezados y pies de página, manejo de drivers para impresoras (independizando el desarrollo de un reporte de una impresora en particular) e impresión ordenada de una tabla principal y una o varias tablas de detalle de acuerdo a las relaciones existentes entre éstas.

El generador de reportes obtiene información en el diccionario de datos de los campos y tablas que constituyen el reporte. Del tipo de datos, valida las operaciones que el usuario desee realizar sobre ellos (tales como establecimiento de cálculos sumarios) y proporciona funciones tales como formateo de los datos. Cuando se desarrolla un reporte que incluye más de una tabla, utiliza al diccionario de datos para determinar las relaciones de dependencia que existen entre las tablas y la cardinalidad de las relaciones, basado en esto define la forma en que se debe reportar la información.

Normalmente los generadores de reportes dan al usuario la posibilidad de seleccionar la información que será reportada mediante el uso del lenguaje de consulta. Adicionalmente proporcionan algunas otras funciones que están asociadas con esos lenguajes como son: Ordenamiento de la información y establecimiento de cálculos sumarios.

Los registros que son impresos pueden ser agrupados de acuerdo a algún orden de sorteo, dentro de cada grupo pueden definirse encabezado y pie de grupo, el desarrollador indica el criterio de agrupación, el generador se encarga de agrupar los registros al momento de ejecutarse el reporte e imprimir los encabezados y pies de grupo correspondientes.

Algunas de las formas de agrupar registros son :

Por valor de un campo : Agrupa los registros con el mismo valor en un campo, los grupos son sorteados en orden ascendente o descendente.

Por rangos de valores : Agrupa los registros de acuerdo al valor dentro de un campo según los rangos de valores.

Por número de registros : Cada grupo consistirá de un número de registros constante.

En el reporte se pueden situar campos calculados basados en expresiones formadas con los campos de las tablas, así como algunos cálculos sumarios basados en expresiones que consideren los campos de las tablas. Estos cálculos sumarios son : SUM, AVERAGE, COUNT, MINIMUM, MAXIMUM

I.6 LENGUAJE DE MANIPULACIÓN DE DATOS

Entre los servicios del administrador de base de datos puede encontrarse un lenguaje de manipulación de datos (LMD) que posee una alta capacidad de cálculo, control de flujo, operaciones de

entrada - salida y de realizar operaciones sobre la base de datos (Actualización, recuperación e intercambio dinámico de información entre el programa y la base de datos). Los LMD´s pueden ser del tipo :

Stand - alone LMD. En este caso el SABD proporciona un compilador o intérprete para el LMD. Sin embargo, estos lenguajes son débiles para realizar algunas operaciones como cálculo numérico u otras.

System call interface.- El usuario escribe un programa en un lenguaje usual (normalmente de tercera generación), los accesos a la base de datos se realizan mediante llamados a subrutinas de la base de datos. En un llamado el usuario indica el requerimiento y especifica los parámetros y variables para conocer el resultado del acceso a la base de datos.

I.7 GENERADOR DE PROGRAMAS

Un generador de programas es un lenguaje que genera código de segunda o tercera generación, de acuerdo a requerimientos proporcionados por el usuario. Ejemplos de estos son los precompiladores, los cuales combinan sus propias características con las del lenguaje al que producen programas.

El generador de programas puede ser utilizado como un elemento integrador de los reportes, formas y querys produciendo el código requerido para generar un sistema o prototipo. El generador

permite Crear un Menú y asociar acciones a ejecutar para cada opción del menú, estas acciones son:

- Menú.- permite definir un submenú para una aplicación.
- Forma.- Permite asociar una forma para poder consultar y actualizar algunas tablas de la base de datos.
- Reporte.- Permite asociar un reporte a alguna opción de un menú.
- Programa.- Permite ejecutar un programa desde alguna opción del menú.

Dependiendo de la acción que se seleccione puede ser necesario dar información adicional al generador de aplicaciones como nombre de las tablas que se requieren para realizar la acción, nombre de las formas o reporte que será asociado o nombre del programa que se va a ejecutar.

Dependiendo del lenguaje de cuarta generación, los programas que éste genere pueden ser modificados para que cumplan más precisamente con los requerimientos del usuario. Esto puede realizarse cuando el lenguaje incluye un lenguaje de tercera generación.

II. DEFINICION DE REQUERIMIENTOS DEL SISTEMA

En el presente capítulo se detallan los requerimientos de un sistema contable para materializar el desarrollo de un sistema de información. De este proceso se establecen las diferencias en el desarrollo que proporcionan las herramientas del lenguaje de cuarta generación.

El propósito del sistema contable es recibir documentos que acrediten operaciones contables de las diversas áreas que constituyen una empresa, este proceso se realiza para llevar un registro contable; así como para conocer la situación financiera de la empresa. Para realizar ésto definimos las siguientes entidades y requerimientos:

II.1 CUENTAS CONTABLES

El sistema consiste de un catálogo de cuentas, en base a éste se acumulan los saldos de las operaciones realizadas por la empresa durante un ejercicio. El ejercico consiste de un año dividido en periodos mensuales. Las cuentas contables constituyen un medio de identificación de los tipos de las operaciones que se realizan, así como un medio de agrupación de los saldos de las operaciones. La agrupación de los saldos es realizada con la finalidad de presentar estados financieros.

Las cuentas contables pueden ser cuentas de afectación las cuales pueden registrar un saldo de un movimiento o bien cuentas de acumulación, que totalizan los saldos de las cuentas de

afectación que les corresponden. Las cuentas están formadas por una cadena de caracteres de longitud fija, cuyos valores las hacen distinguibles. Esta cadena a la vez consiste de un número constante de niveles. Cada uno de los niveles a su vez consiste de un número fijo de caracteres, a partir de los cuales se determina si una cuenta acumula a otra cuenta. Esto es decidido mediante los criterios de acumulación que se definen en seguida.
ejemplo :
cuenta 100-00-00 esta cuenta consta de tres niveles contables consistiendo el primer nivel de tres caracteres y los dos restantes de dos caracteres cada uno.
Las Cuentas contables pueden ser de naturaleza acreedora o deudora. Los saldos de una cuenta de naturaleza acreedora son negativos, es decir disminuyen al capital, mientras que los saldos de una cuenta de naturaleza deudora son positivos lo cual implica que aumentan al capital.

II.2 POLIZAS CONTABLES

El registro contable se realiza mediante la elaboración de pólizas contables, estas pólizas transforman un documento correspondiente a una operación en un documento contable. Este documento identifica las cuentas afectadas por la operación y los importes por las que las afecta (ver página siguiente).

POLIZA INGRESOS

CUENTA	SUB. CUENTA	NOMBRE	PARCIAL	DEBE	HABER
110	0101	Cta. de cheques serfin		1,500.00	
210	0201	Acreedores diversos			
		Marco Aurelio Reyna Padilla			1,500.00
			SUMAS IGUALES	$ 1,500.00	

CONCEPTO

Prestamo del señor Marco Aurelio Reyna Padilla

CONCEPTO	REVISADO	AUTORIZADO	AUXILIARES	DIARIO	FECHA	POLIZA NO.

POLIZA DE DIARIO

CUENTA	SUB. CUENTA	NOMBRE	PARCIAL	DEBE	HABER
110	0101	Cuenta de cheques serfin			57 20
110	0400	I.V.A. acreditable		5 20	
460	0100	Comisiones bancarias		52 00	
		SUMAS IGUALES		57 20	57 20

CONCEPTO

comisiones bancarias del mes de Noviembre

HECHO POR	REVISADO	AUTORIZADO	AUXILIARES	DIARIO	FECHA	POLIZA NO.
					03/Nov/94	I

Seguridad 1002

CHEQUE POLIZA

COPIA DEL CHEQUE

Io. de Abril de 96

Telefonos de México S.A. 226.89

Doscientos veintiseis pesos 89/100 M.N. ##

pago telefono mes de Abril

CONCEPTO DEL PAGO

FIRMA CHEQUE RECIBIDO

DISTRIBUCION CHEQUE - EMBARCADERO COPIA COLOR ARCHIVO CON COMPROBANTES COPIA BLANCA ARCHIVO ALFABETICO CONTABILIDAD CONCILIACIONES BANCARIAS

CUENTA	SUB-CUENTA	NOMBRE	PARCIAL	DEBE	HABER
110	0101	Cuenta de cheques serfin			226.89
110	0400	I.V.A. Acreditable		20.62	
450	0900	Gastos generales Telefono		206.27	
			SUMAS IGUALES	226.89	

1011

HECHO POR: | REVISADO | AUTORIZADO | AUXILIARES | DIARIO | POLIZA No. 122

Los tipos válidos de pólizas son : Ingresos, Egresos, Diario y Cheques. Las pólizas deben ser distinguibles mediante la asignación de un consecutivo para cada una, esto es realizado para llevar control de las pólizas que constituyen un período. Las pólizas de tipo Ingreso, Egreso y Diario requieren que su numeración se reinicie en cada período, mientras que las pólizas Cheque requieren que la numeración al inicio de un período, continue a partir del último número asignado en el período anterior. Cada póliza tiene que estar cuadrada, es decir, la suma de cargos debe ser igual a la suma de abonos. No se puede registrar una póliza que tenga un movimiento que no incluya un cargo o abono.

II.3 MOVIMIENTOS CONTABLES

Los movimientos contables identifican los conceptos que constituyen una póliza y los importes que les corresponden. Asocian las cuentas correspondientes para registrar cada uno de los conceptos, identificando así como se generan operaciones dentro de la empresa.

Ejemplo : En el caso de la elaboración de una póliza cheque normalmente se tienen los siguientes movimientos :

- un movimiento de abono para reflejar la salida de dinero de la cuenta de cheques

- un movimiento (o varios) de cargo para reflejar el concepto por el que se realizó el gasto; como pudieran ser gastos de papelería, pago de renta, teléfono, luz, etc., y
- algún otro movimiento especial, como podría ser el I.V.A.

II.4 ASENTAMIENTO DE POLIZAS

Las pólizas capturadas deben poder ser modificadas y revisadas hasta que se decida que éstas afecten los saldos de las cuentas. El sistema debe generar un reporte de pólizas capturadas, para verificar que las pólizas que se van a asentar sean correctas.

II.5 SALDOS DE LAS CUENTAS

Cuando se reflejan los movimientos de las pólizas en los saldos de las cuentas se deben hacer las siguientes consideraciones :
Los saldos de las cuentas deben ser actualizados mediante un proceso que acumule los saldos de los movimientos de las pólizas registradas, dichos saldos son acumulados en las cuentas contables. Los saldos de las cuentas deben ser acumulativos, lo cual quiere decir que el saldo de una cuenta en un período dado debe ser igual a :
saldo de un período = saldo del período anterior + cargos del período - abonos del período.

Para el caso de cuentas deudoras cuyo saldo es positivo, para las cuentas acreedoras se tiene :

-saldo de un período = -saldo del período anterior + cargos del período - abonos del período, o bien :

saldo de un período = saldo del período anterior - cargos del período + abonos del período.

II.6 REPORTES FINANCIEROS

Para la presentación de estados financieros se obedecen las siguientes reglas :

Criterio para definición de cuentas de acumulación

Los valores de los niveles de la cuenta de acumulación diferentes de cero a la izquierda, deben ser iguales a los valores de los mismos niveles de la cuenta de afectación. Pero en aquellos niveles a la derecha de las cuentas de acumulación cuyo valor sea nulo o cero la cuenta de afectación debe tener al menos alguno de estos niveles con un valor diferente de cero.

Cuenta

100-00-00 es cuenta de acumulación de la cuenta :

100-10-00 y de la cuenta :

100-10-01

Una cuenta de afectación puede estar subordinada a más de una cuenta de acumulación, como puede observarse en el ejemplo anterior.

Se permite que el primer nivel de una cuenta de acumulación sea diferente del mismo nivel de una cuenta de afectación, siempre y cuando la cuenta de acumulación sea la cuenta de mayor correspondiente a la cuenta de afectación . Es decir, que el único caracter diferente de cero o nulo de la cuenta de acumulación es el primer caracter del primer nivel, los demás caracteres de la cuenta serán igual a cero o nulo. Esta cuenta será de acumulación para cualquier otra cuenta cuyo primer caracter del primer nivel sea igual.

Ejemplo :

Cuenta 100-00-00 es cuenta de mayor de la cuenta :

110-00-00 y de la cuenta :

121-01-01

En el caso de que una cuenta de afectación acumule a una cuenta de naturaleza distinta, el saldo de la cuenta de acumulación se ve disminuído por el saldo de la cuenta de afectación. En el caso del saldo de los movimientos, éstos siempre se acumulan a sus cuentas sin importar la naturaleza de éstas.

Ejemplo

cuenta de mayor 100-00-00 acumula a :

110-01-00 que a la vez acumula a:

110-01-01 y a :

110-01-02

Bajo la siguiente situación :

Cuenta

110-01-00 con naturaleza deudora y

Cuenta 110-01-01 con naturaleza deudora y saldo $ 1,500.00 y
Cuenta 110-01-02 con naturaleza acreedora y saldo $ 500.00
Tenemos después de la acumulación la siguiente situación :
Cuenta 110-01-00 con saldo $ 1,000.00
Si tuviéramos la cuenta 110-01-00 con naturaleza acreedora el saldo final de la cuenta sería - $ 1,000.00
Los reportes básicos que el sistema genera son la balanza que reporta el saldo de un rango de cuentas al final de un período, considerando las naturalezas de las cuentas para las cuales se está elaborando el reporte y los niveles de acumulación de las mismas, y el reporte auxiliar de movimientos que contiene la misma información que la Balanza de comprobación, pero adicionalmente detalla qué movimientos afectaron dentro del período solicitado el saldo de la cuenta, reportando cómo fue variando el saldo de la cuenta durante el período solicitado (A continuación se presentan algunos reportes).

La Nacional
Balance General, al 31 de Diciembre de 1995

Activo

Circulante			
Caja	$10.000,00		
Bancos	$5.000,00		
Mercancias	$15.000,00		
Clientes	$5.000,00		
Documentos por	$7.000,00		
Deudores diversos	$3.000,00	$45.000,00	
Fijo			
Edificios	$20.000,00		
Mobiliario	$12.000,00		
Equipo de reparto	$8.000,00	$40.000,00	
Cargos diferidos			
Gastos de instalación		$2.000,00	$87.000,00

Pasivo

Flotante			
Proveedores	$10.000,00		
Documentos por	$8.000,00		
Acreedores diversos	$2.000,00	$20.000,00	
Consolidado			
Hipotecas por pagar		$10.000,00	
Créditos diferidos			
Rentas cobradas por		$1.000,00	$31.000,00
Capital			**$56.000,00**

---------- ---------- ----------
Tenedor **Contador** **Propietario**

La Nacional

Balance General, al 31 de Diciembre de 1995

Activo			Pasivo			
Circulante			**Flotante**			
Caja	$10,000.00		Proveedores	$10,000.00		
Bancos	$5,000.00		Documentos por pagar	$8,000.00		
Mercancías	$15,000.00		Acreedores diversos	$2,000.00	$20,000.00	
Clientes	$5,000.00					
Documentos por cobrar	$7,000.00		**Consolidado**			
Deudores diversos	$3,000.00	$45,000.00	Hipotecas por pagar		$10,000.00	
			Créditos diferidos			
Fijo			Rentas cobradas por anticipado		$1,000.00	$31,000.00
Edificios	$20,000.00					
Mobiliario	$12,000.00		Capital			$56,000.00
Equipo de reparto	$8,000.00	$40,000.00				
Cargos diferidos						
Gastos de instalación		$2,000.00				
TOTAL ACTIVO		**$87,000.00**	**TOTAL PASIVO MAS CAPITAL**			**$87,000.00**

Tenedor

Contador

Propietario

La Nacional

Estado de resultados del 1o. de Enero al 31 de Diciembre de 1995

Ventas Totales			$195,000.00	
Menos : devoluciones sobre ventas		$3,000.00		
descuentos sobre ventas		$2,000.00	$5,000.00	
Ventas netas				$190,000.00
Inventario Inicial			$125,000.00	
Compras	$80,000.00			
Más : Gastos de compras	$2,000.00			
Compras totales		$82,000.00		
Menos : Devoluciones sobre compras	$6,000.00			
Rebajas sobre compras	$1,000.00	$7,000.00		
Compras netas			$75,000.00	
Suma			$200,000.00	
Menos : Inventario final			$60,000.00	
Costo de lo vendido				$140,000.00
Utilidad bruta				$50,000.00
Gastos de operación				
Gastos de venta				
Renta del almacen	$1,700.00			
Propaganda	$900.00			
sueldos agentes y dependientes	$3,200.00			
impuesto sobre ingresos mercantiles	$1,600.00			
consumo de luz	$100.00	$7,500.00		
Gastos de Administración				
Renta de oficinas	$1,200.00			
Sueldos de	$4,300.00			
Papelería y útiles	$300.00			
Consumo de luz	$200.00	$6,000.00	$13,500.00	
Productos				
Intereses a nuestro	$700.00			
Descuentos sobre compras	$500.00	$1,200.00		
Gastos financieros				
Intereses a nuestro	$500.00			
Descuentos sobre	$450.00			
Gastos de	$50.00	$1,000.00	$200.00	$13,300.00
Utilidad de				$36,700.00
Otros gastos				
Pérdidas en venta de mobiliario		$2,000.00		
Pérdida en venta de acciones		$600.00	$2,600.00	
Otros productos				
Comisiones		$200.00		
Dividendos cobrados		$400.00	$600.00	$2,000.00
Utilidad del ejercicio				$34,700.00

Tesorero	Contador	Propietario

III. ANALISIS DEL SISTEMA

En este capítulo se presenta el resultado del análisis del sistema, para realizarlo se ocupó el análisis estructurado. Esta técnica parte de un estudio del sistema en su totalidad, especificando la información que debe recibir y la que debe generar para cumplir con su objetivo. Posteriormente en un proceso iterativo se detectan las operaciones que se deben realizar dentro del sistema para generar esta información, sobre estas operaciones nuevamente se realiza el mismo estudio; partiendo de la información que recibe y regresa, hasta alcanzar el nivel en el que las operaciónes no pueden reducirse.

El resultado del Análisis son diagramas que detallan estas operaciones y la información que fluye entre ellas. El proceso iterativo es reflejado en la numeración que se hace de las operaciones. Esta numeración es sucesiva, comenzando con el número 1 para el sistema en su totalidad, para la siguiente etapa se asignan números sucesivos a las operaciones que se encontraron (1.1, 1.2, 1.3, ...,etc.). El último dígito de la numeración identifica a la operación y los dígitos anteriores identifican al proceso al que corresponde la operación. Por ejemplo el proceso 1.1.2 identifica a la operación 2 del proceso 1.1. Esto es remarcado mediante una nota en la esquina inferior izquierda del diagrama.

En este capítulo se desarrolla el análisis considerando un lenguaje de cuarta generación, se realiza también el análisis de un módulo considerando un lenguaje de tercera generación para establecer las diferencias entre ambos.

III.1 NOTACION DIAGRAMA DE FLUJO DE DATOS

Flujo de datos .- Representa el orden en el que se está realizando el proceso y la información que fluye en cada etapa

Proceso : opera sobre la información que esta fluyendo de alguna de estas formas :

1.- Puede transformar la estructura de los datos

2.- Puede transformar la información que recibe

Almacenamiento de datos.- depósito de datos sobre el que se pueden realizar cuatro operaciones

Leer archivo

Actualiza archivo

Agrega un nuevo registro

Borra un registro

Fuente y destino de la información

Control de flujo .- Activa o desactiva al proceso apuntado

III.2 MODELO ENTIDAD - RELACION

Con este diagrama se detallan las entidades que pertenecen al sistema (Elementos que constituyen al sistema, con operaciones y atributos claramente especificados), y las relaciones que existen entre ellas

Notación :

Entidad.- Elemento del sistema

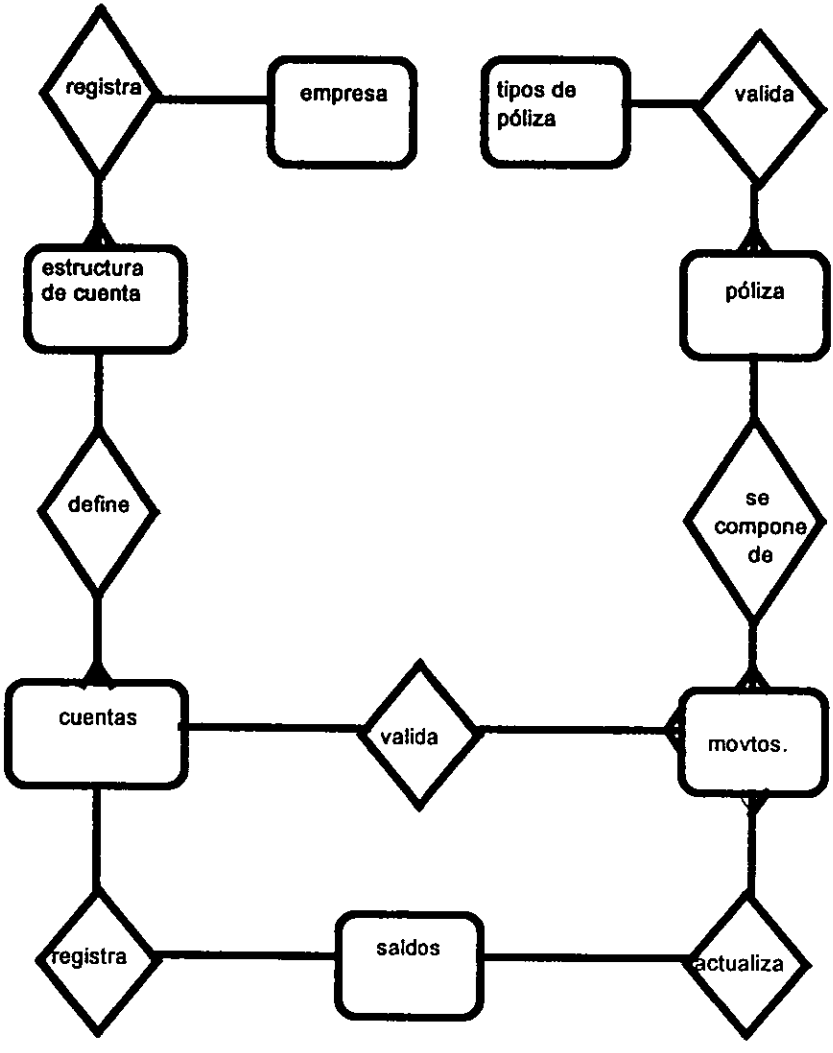
Relación : Determina la razón por la cual se establece una relación entre dos o más entidades

Cardinalidad de la relación : No. de elementos de una entidad que participan dentro de una relación. Ejemplo: un cliente posee varias facturas, un grupo posee muchos estudiantes y un estudiante puede tener varios grupos

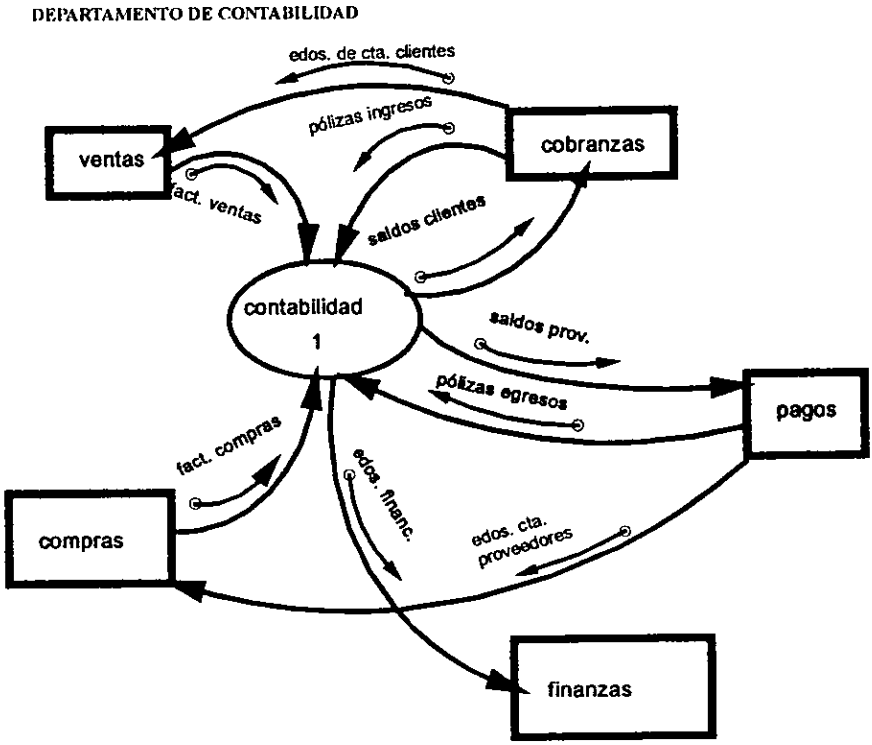
indica que muchos elementos pertenecen a la relacion

indica que sólo un elemento pertenece a la relación

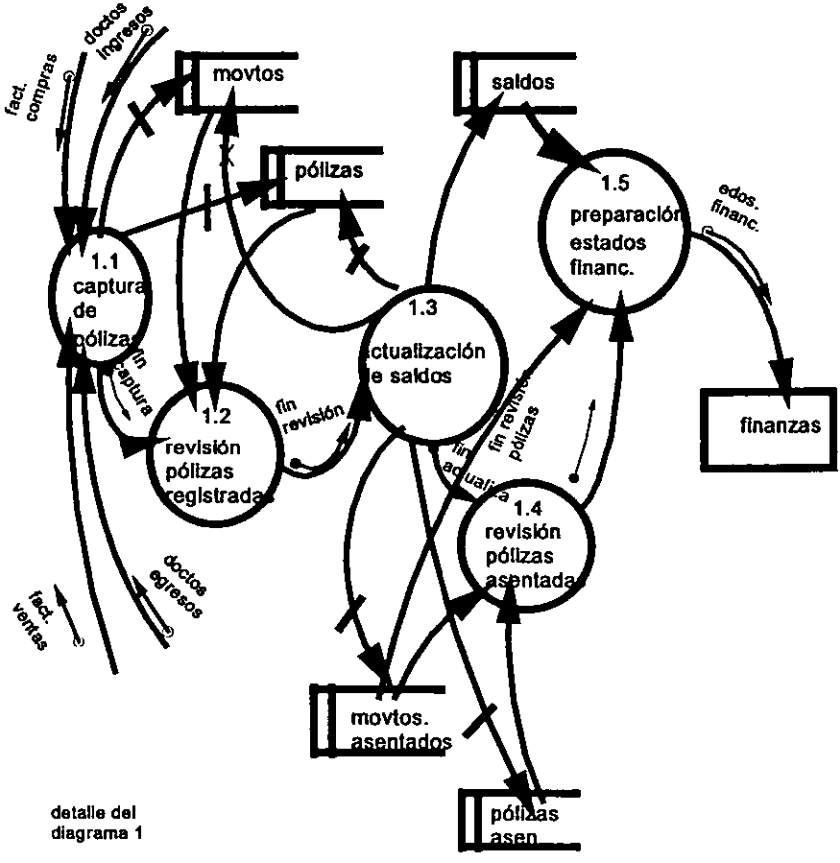
MODELO ENTIDAD RELACION DEL SISTEMA CONTABLE



III.3 ANALISIS PARA UN LENGUAJE DE CUARTA GENERACION

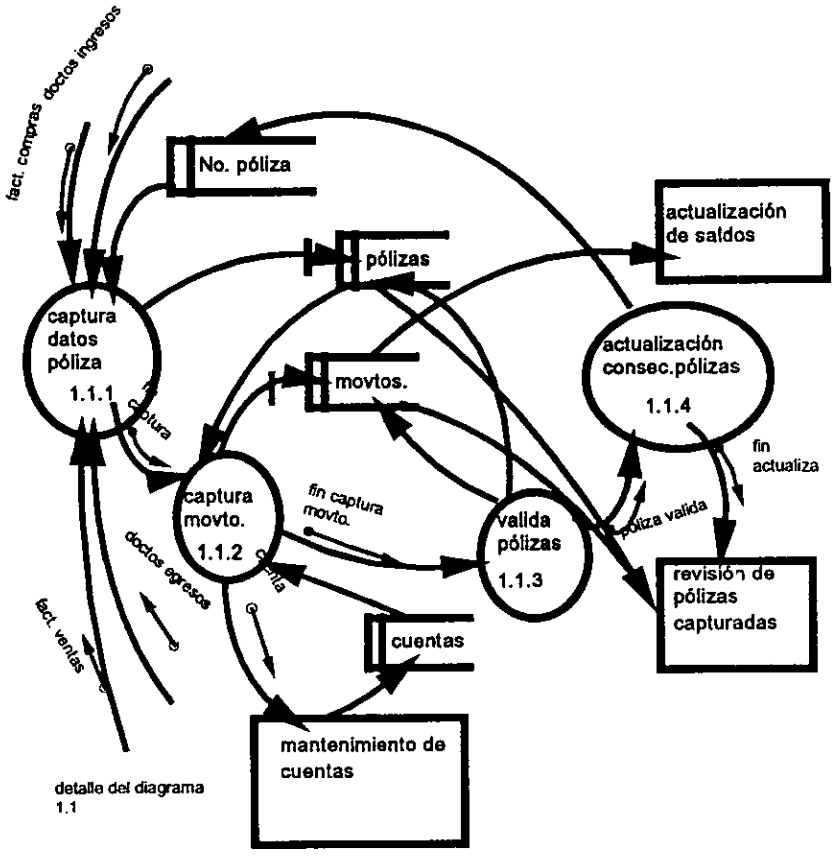


SISTEMA CONTABLE

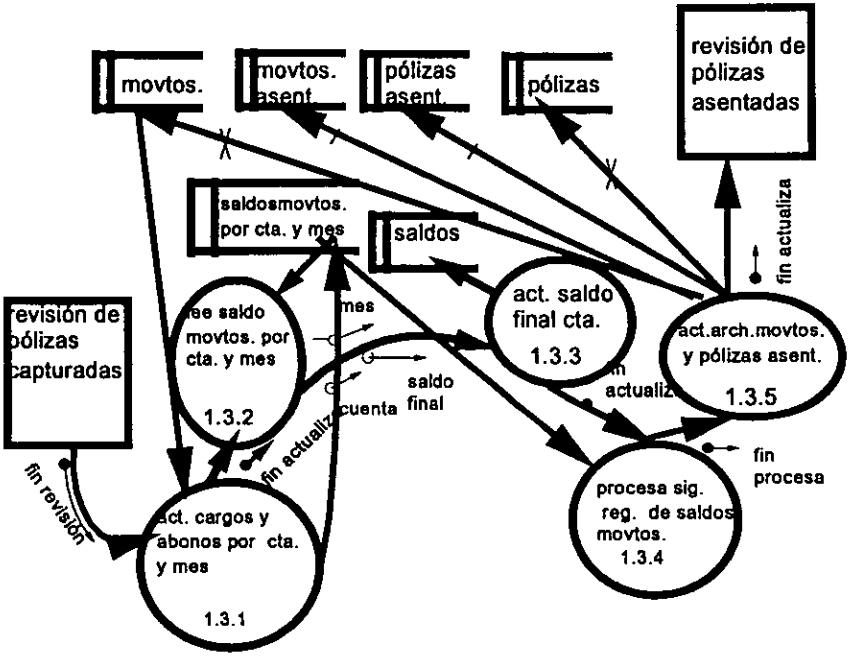


detalle del diagrama 1

CAPTURA DE MOVIMIENTOS

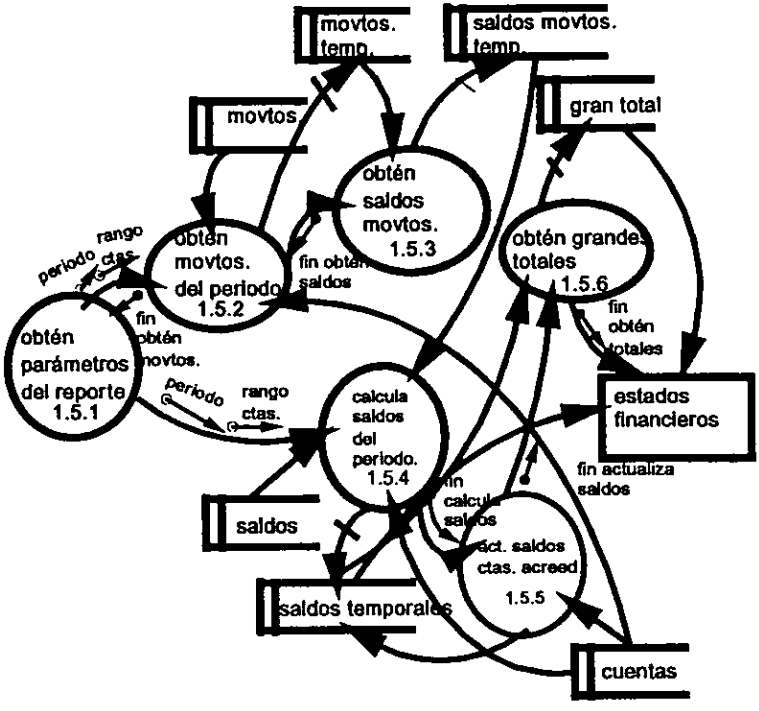


ACTUALIZACION DE MOVIMIENTOS



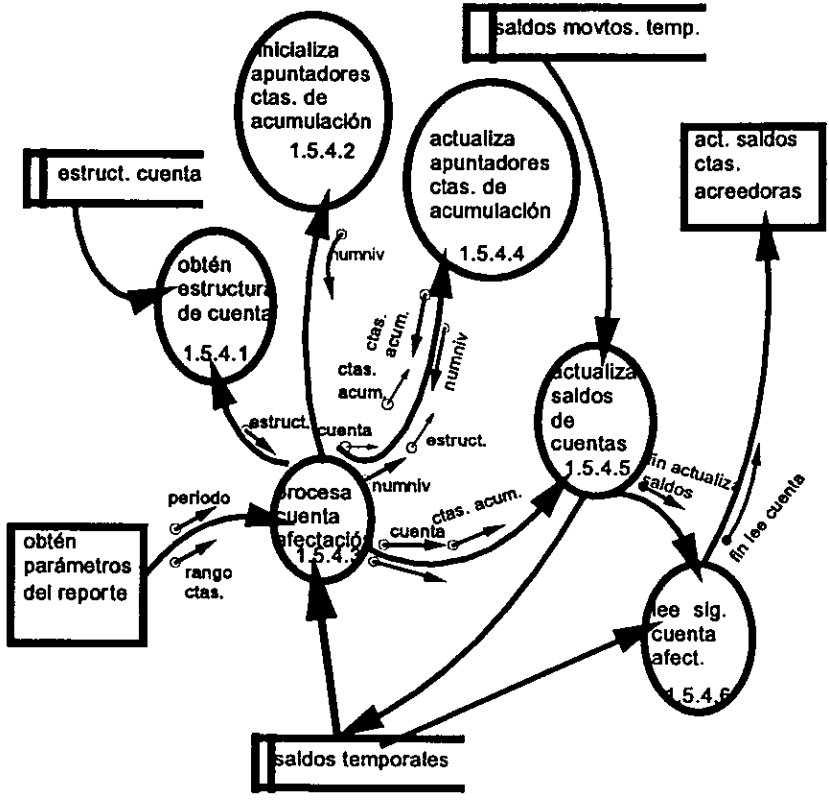
detalle del
diagrama 1.3

PREPARACION DE ESTADOS FINANCIEROS



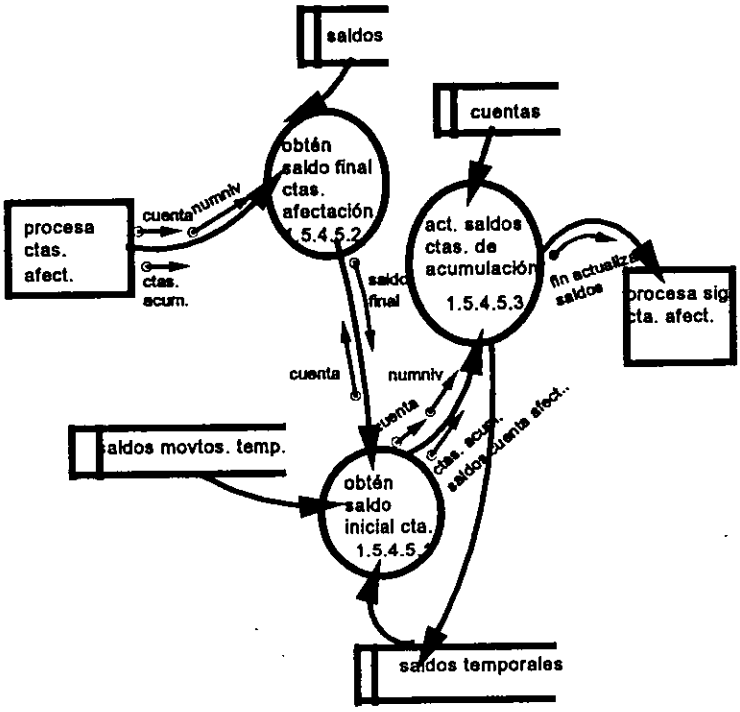
detalle del
diagrama 1.5

CALCULA SALDOS DEL PERIODO



detalle del diagrama
1.5.4

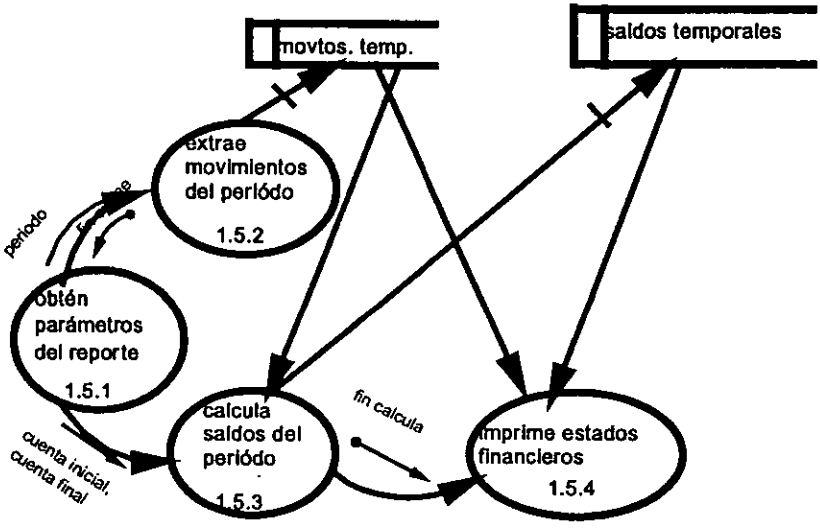
ACTUALIZA SALDOS DE CUENTAS



detalle del diagrama
1.5.4.5

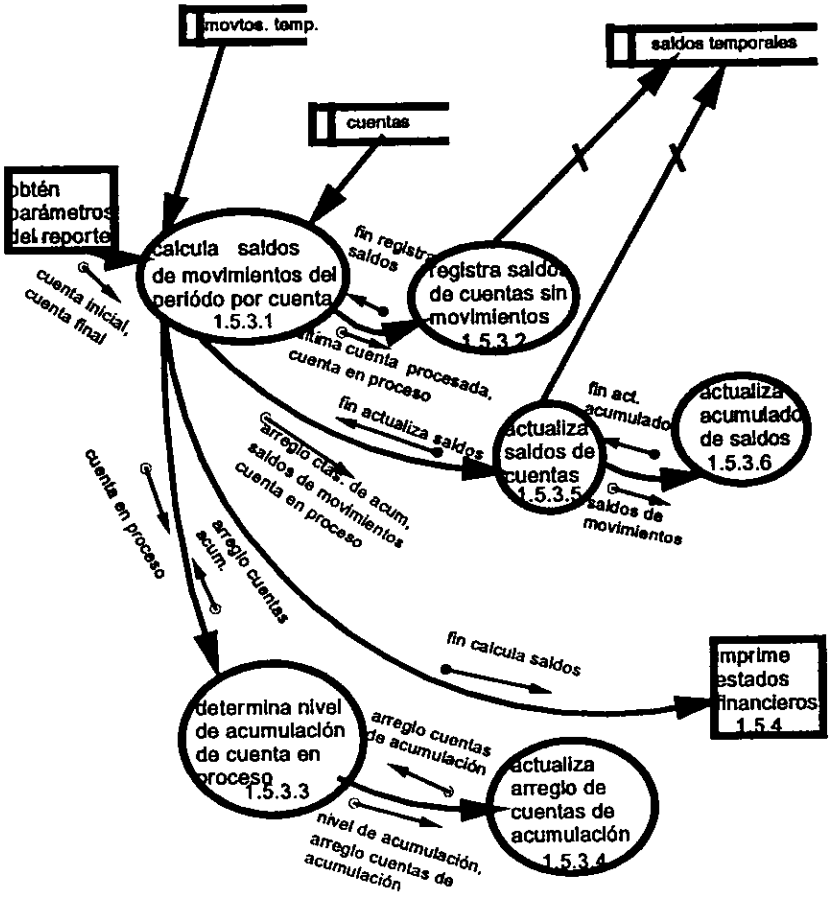
III.4 ANALISIS PARA UN LENGUAJE DE TERCERA GENERACION

PREPARACION DE ESTADOS FINANCIEROS



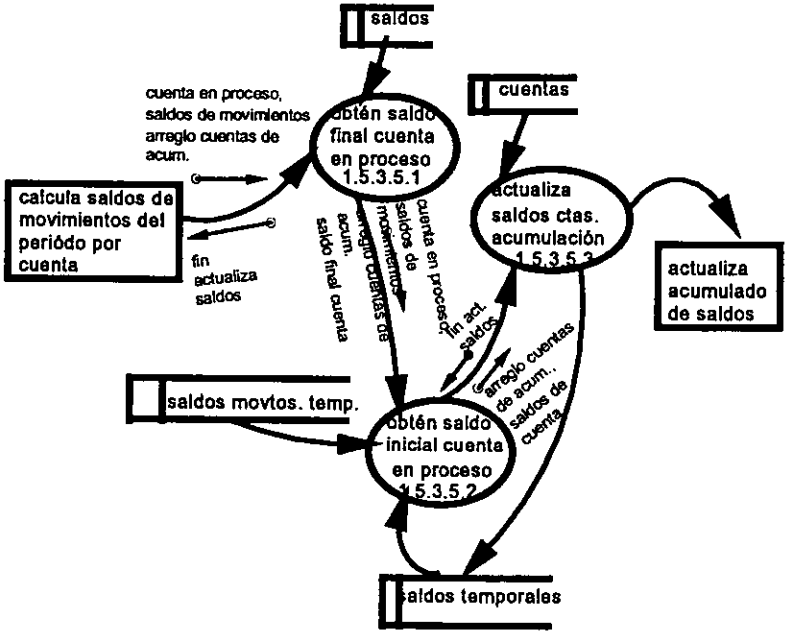
detalle del diagrama 1.5

CALCULA SALDOS DEL PERIODO



detalle del diagrama 1.5.3

ACTUALIZA SALDOS DE CUENTAS



detalle del diagrama 1.5.3.5

III.5 CONCLUSIONES :

De las secciones III.3 y III.4 concluímos que la técnica de análisis estructurado es igualmente válida para desarrollar un sistema, tanto con un lenguaje de tercera como con uno de cuarta generación. Sin embargo debido a las diferencías existentes entre estos lenguajes los resultados del análisis no son idénticos.

En el análisis para un lenguaje de tercera generación se puede observar en el módulo desarrollado, que en un solo proceso (1.5.3) se concentran gran parte de las funciones ejecutadas, como resultado de esto se observa una disminución en el número de procesos que constituyen el primer nivel del diagrama. En el análisis para un lenguaje de cuarta generación se observa una mayor concentración de procesos en el primer nivel con respecto a los demás niveles.

Esta diferencia es debida al lenguaje de consulta (query) que permite realizar cálculos sobre archivos completos mediante un solo comando, esto da independencia a las operaciones que constituyen un proceso, pues pueden realizar sus cálculos sin importar la forma en la que hayan realizado sus funciones las operaciones precedentes, o los archivos que hayan utilizado para realizarlas. En el caso del lenguaje de tercera generación el análisis considera las operaciones que se realizarán, así como los archivos que se procesarán. El cálculo de acumulados y totales es desarrollado por el programador, por lo que trata de eficientizar el procesamiento de archivos que se realiza dentro

de las operaciones, evitando al máximo duplicidades en procesamiento para no degradar el desempeño del sistema cuando éste se encuentre en operación; incrementandose las dependencias entre los procesos.

En el diagrama 1.5.3 para el lenguaje de tercera generación observamos que la operación 1.5.3.1 calcula los saldos de los movimientos para un periodo, conforme realiza sus funciones ejecuta al proceso 1.5.3.5 (Actualiza saldos de cuentas) y le transmite variables (saldos de movimientos, cuenta en proceso) para que ejecute sus funciones. Esta característica del análisis para un lenguaje de tercera generación incrementa el acoplamiento que se da entre las operaciones, pues al mismo tiempo que se procesa un archivo se realizan cálculos para otras operaciones que dependan del mismo archivo, elevándose por lo tanto el número de datos que se transmiten entre las operaciones que constituyen un diagrama.

IV. DISEÑO DEL SISTEMA

En este capítulo se presenta el diseño del sistema, para realizarlo se utilizó la técnica estructurada. El resultado del diseño son cartas de estructura que definen un esquema modular del sistema.

Se presenta el diseño para un lenguaje de cuarta generación y también el diseño de un módulo para un lenguaje de tercera generación. De estos resultados se establecen algunas diferencias.

NOTACION PARA EL DISEÑO DEL SISTEMA

Proceso : Realiza alguna operación definida para el sistema

Flujo de datos .- Representa la información que fluye en la operación

Flujo de control .- Representa información que determina una decisión para un proceso

Conector .- Indica la continuación de un diagrama

IV.1 DISEÑO DE LA BASE DE DATOS

La estructura de la base de datos después de realizar su normalización, es presentada en el siguiente esquema

ENTIDAD	ID	ATRIBUTO	DOMINIO	DESCRIPCIÓN	RELACIÓN
EMPRESA	cve_cia	clave de compañía	alfabético	clave de identificación de la compañía	struc_cta
	nom_cia	nombre de compañía	alfabético	nombre de la compañía	
	dir_dat	directorio de archivos de datos	alfabético	directorio de almacenamiento de los archivos de registro de información de la compañía	
	regfedc	registro federal de causantes	alfabético	registro federal de causantes de la compañía	
	direcc	dirección de la compañía	alfabético	dirección de localización de la compañía	
	colonia	colonia de la compañía	alfabético	colonia de localización de la compañía	
	codpos	código postal	alfabético	código postal de la compañía	
	ciudad	ciudad	alfabético	ciudad donde reside la compañía	
	país	país	alfabético	país de residencia de la compañía	
	niveles	número de niveles de la cuenta contable	entero	número de niveles del que consiste la cuenta contable de la compañía	

ENTIDAD	ID	ATRIBUTO	DOMINIO	DESCRIPCIÓN	RELACIÓN
CUENTAS	Num-cta	número de cuenta	alfabético	identificador de la cuenta contable	movtos saldos
	tipo-cta	tipo de la cuenta	alfabético	A.- cuenta de activo P.- cuenta de pasivo C.- cuenta de capital	
	Acum	indicador de acumulación	alfabético	S.- cuenta de acumulación N.- cuenta de afectación	
	Nat-cta	naturaleza de la cuenta	alfabético	D.- cuenta deudora A.- cuenta acreedora	
	desc-cta	descripción de la cuenta	alfabético	descripción de la cuenta contable	
	stat-cta	estatus de la cuenta	alfabético	A.- cuenta activa B.- cuenta dada de baja	

TIP_POL	tipo-pol	tipo de póliza	alfabético	D.- póliza de diario I .- póliza de ingreso E.- póliza de egresos C.- póliza de cheque	pólizas
	desc_pol	descripción del tipo de póliza	alfabético	descripción de tipos de póliza	
	no-pol	número de póliza	entero	consecutivo para asignación de número de póliza durante la captura	

POLIZAS	tipo-pol	tipo de póliza	alfabético	tipo de la póliza	tip_pol movtos
	mes	mes de la póliza	entero	mes de la póliza	movtos
	no-pol	número de póliza	entero	número de póliza	movtos
	referencia	referencia de la póliza	alfabético	referencia para identificación de la póliza	
	fech-pol	fecha de la póliza	date	fecha en la que se generó la póliza	
	importe	importe de la póliza	real	importe de la póliza	
	descrip	descripción de la póliza	alfabético	descripción de la póliza	
	status	estatus de la póliza	alfabético	P.- póliza no asentada A.- póliza asentada	

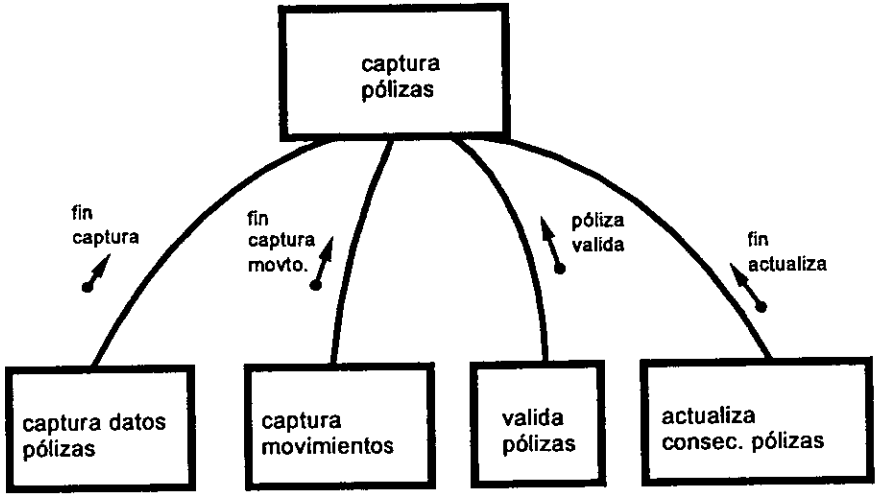
MOVTOS	tipo-pol	tipo de póliza	alfabético	tipo de póliza a la que corresponde el movimiento	pólizas
	mes	mes de póliza	entero	mes de la póliza a la que corresponde el movimiento	pólizas
	no-pol	número de póliza	entero	número de póliza a la que corresponde el movimiento	pólizas
	num_mov	número de movimiento	entero	número de movimiento de la póliza	
	num-cta	cuenta del movimiento	alfabético	cuenta a la que afecta el movimiento	cuentas
	desc-cta	descripción de la cuenta del movto	alfabético	descripción de la cuenta	
	referencia	referencia del movimiento	alfabético	identificador del movimiento	
	cargo	importe del cargo del movimiento	real	importe del cargo del movimiento	
	abono	importe del abono del movimiento	real	importe del abono del movimiento	
	stat-mov	estatus del movimiento	alfabético	P.- movimiento no asentado A.- movimiento asentado	

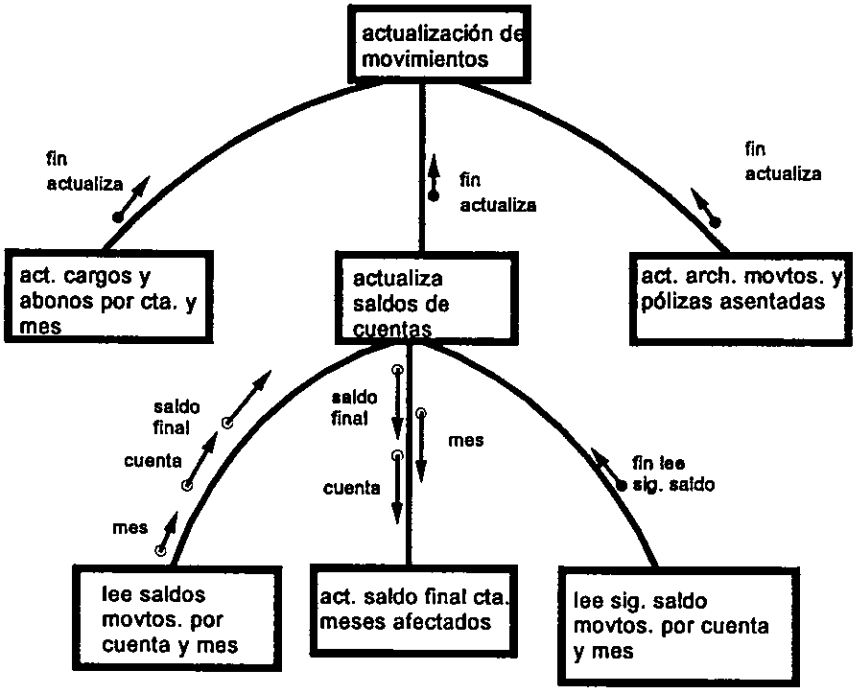
struc_cta	cve_cia	clave de la compañía	alfabético	clave de la compañía de la que se define estructura de la cuenta contable	empresa
	niv_cta	número de nivel de la cuenta	entero	número de nivel de la cuenta contable que se define	
	lon-niv	longitud del nivel	entero	longitud de caracteres del nivel contable que se está definiendo	

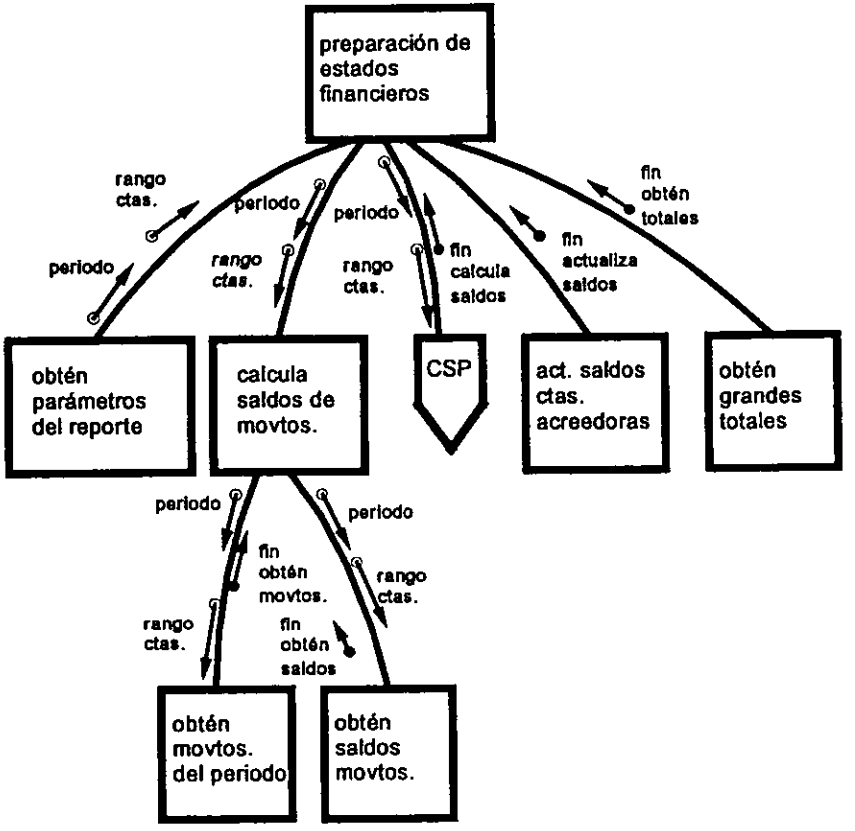
SALDOS	num-cta	número de cuenta	alfabético	número de cuenta al que corresponde el saldo	cuentas
	saldo1	saldo período 1	real	saldo del período 1	
	saldo2	saldo período 2	real	saldo del período 2	
	saldo3	saldo período 3	real	saldo del período 3	
	saldo4	saldo período 4	real	saldo del período 4	
	saldo5	saldo período 5	real	saldo del período 5	
	saldo6	saldo período 6	real	saldo del período 6	
	saldo7	saldo período 7	real	saldo del período 7	
	saldo8	saldo período 8	real	saldo del período 8	
	saldo9	saldo período 9	real	saldo del período 9	
	saldo10	saldo período 10	real	saldo del período 10	
	saldo11	saldo período 11	real	saldo del período 11	
	saldo12	saldo período 12	real	saldo del período 12	
	saldo13	saldo período 13	real	saldo del período 13	

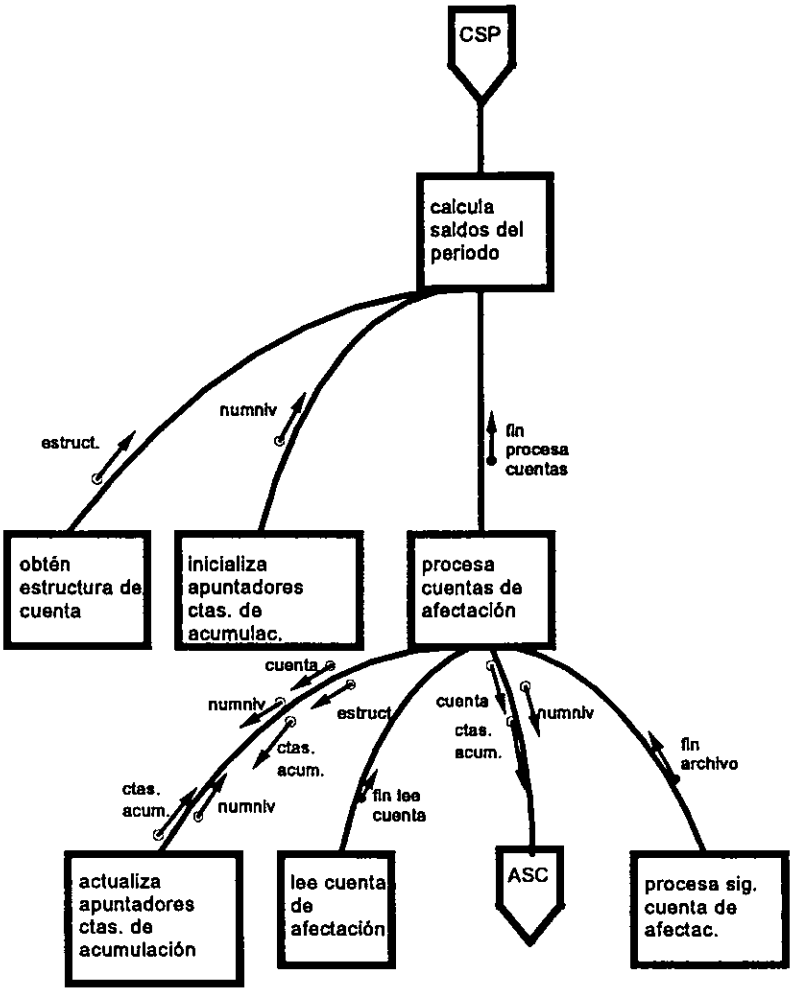
IV.3 DISEÑO PARA UN LENGUAJE DE CUARTA GENERACION

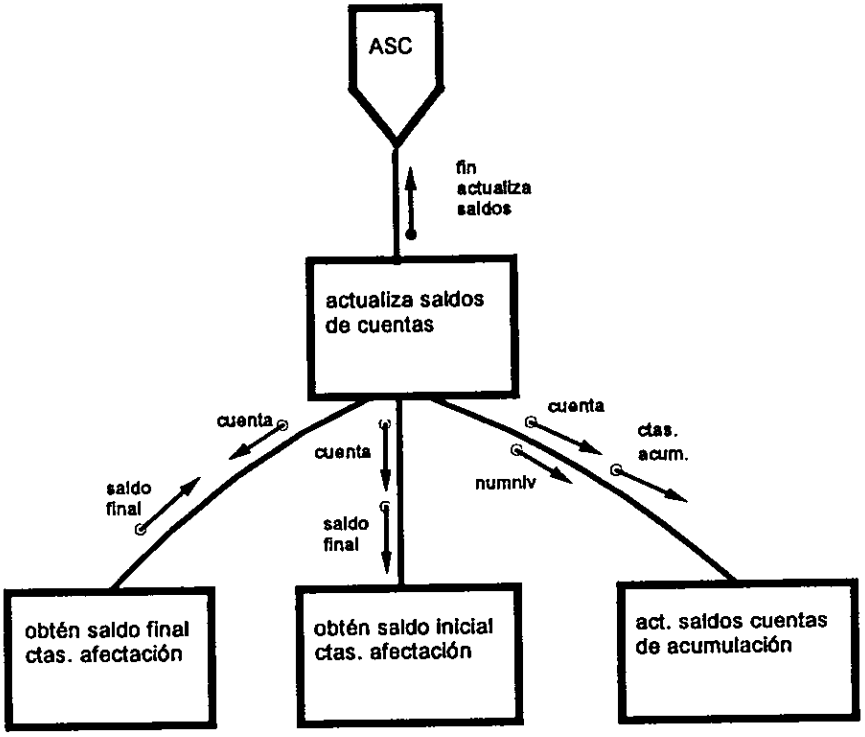
Para el desarrollo de las cartas de estructuras se parte de los diagramas resultantes del análisis realizado en el capítulo III. Las cartas de estructura muestran la jerarquía con la que se ejecutarán las operaciones para conseguir sus objetivos. Mientras más alta se encuentre una operación, mayor es su jerarquía para ejecutar a las operaciones que tiene subordinadas.



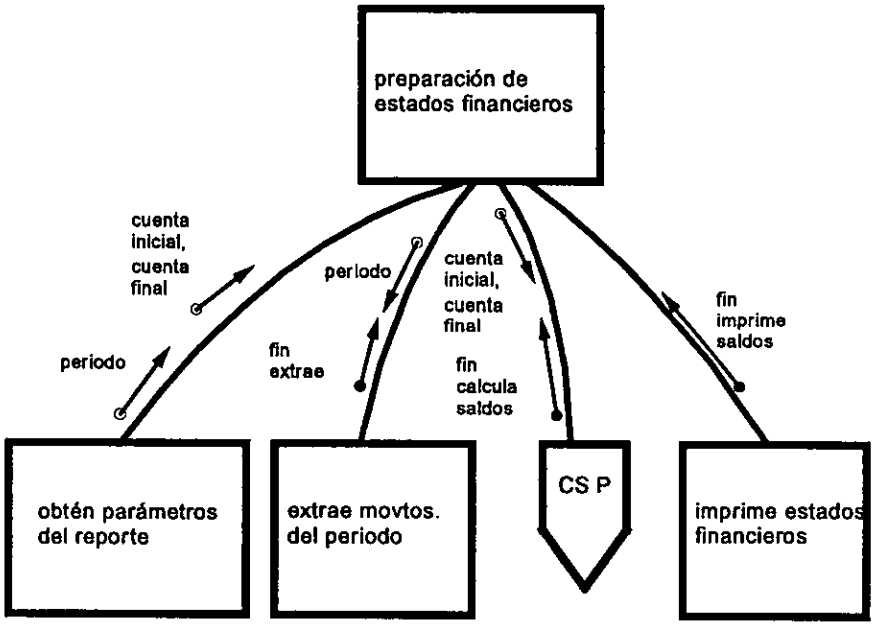

actualización de movimientos
fin actualiza
fin actualiza
fin actualiza
act. cargos y abonos por cta. y mes
actualiza saldos de cuentas
act. arch. movtos. y pólizas asentadas
saldo final
cuenta
mes
saldo final
mes
cuenta
fin lee sig. saldo
lee saldos movtos. por cuenta y mes
act. saldo final cta. meses afectados
lee sig. saldo movtos. por cuenta y mes

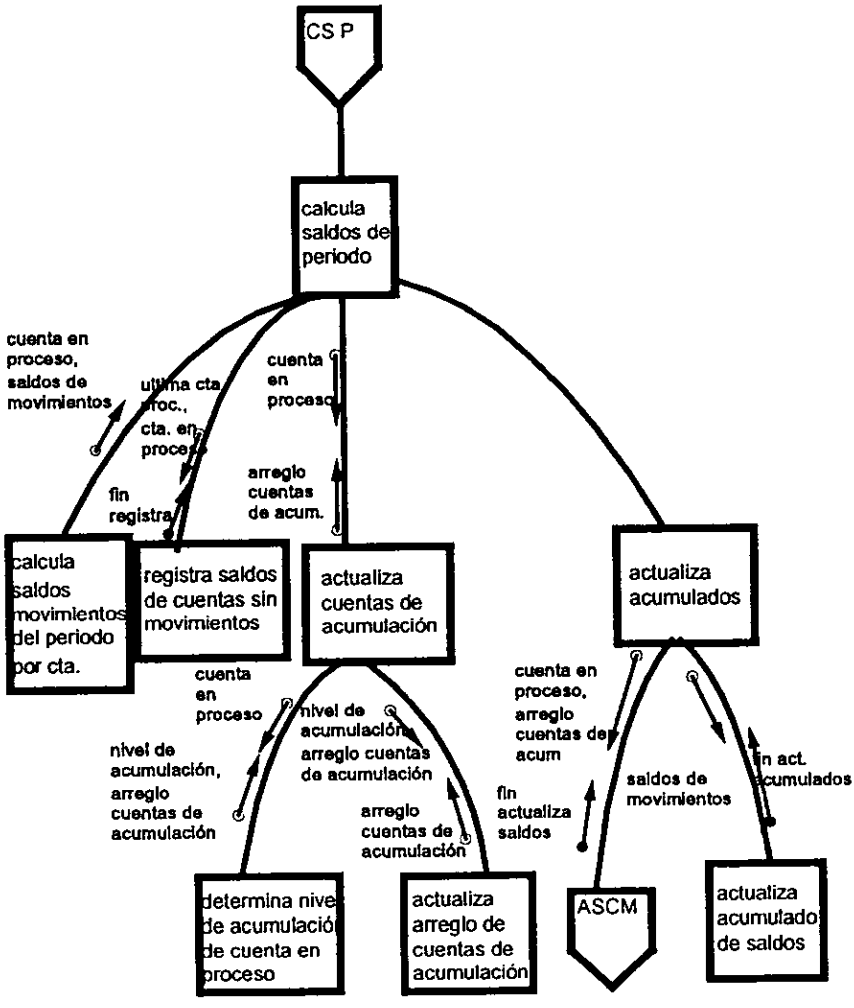

preparación de estados financieros
rango ctas.
periodo
periodo
periodo
rango ctas.
rango ctas.
fin calcula saldos
fin actualiza saldos
fin obtén totales
obtén parámetros del reporte
calcula saldos de movtos.
CSP
act. saldos ctas. acreedoras
obtén grandes totales
periodo
periodo
fin obtén movtos.
rango ctas.
rango ctas.
fin obtén saldos
obtén movtos. del periodo
obtén saldos movtos.

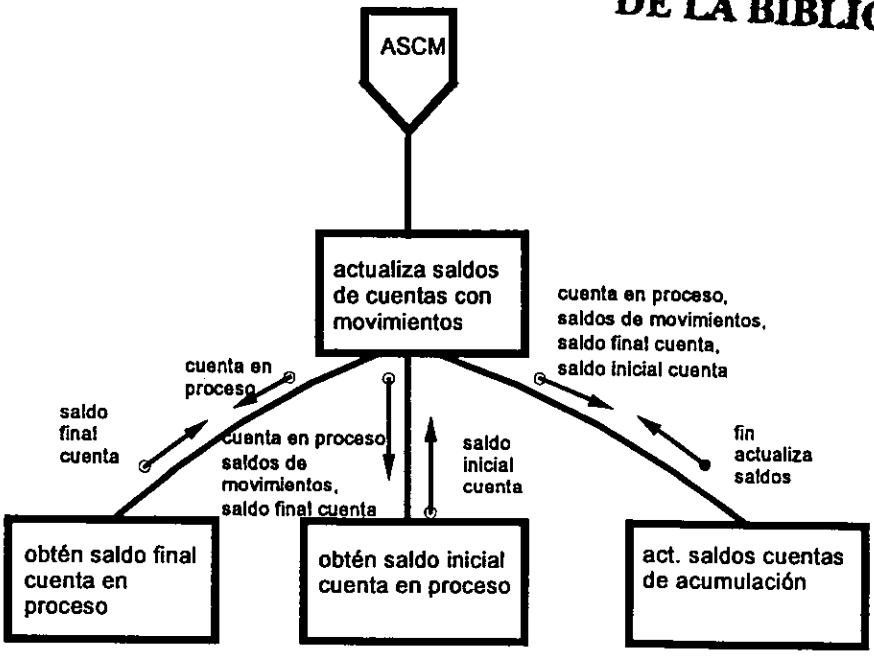

CSP
calcula saldos del periodo
estruct.
numniv
fin procesa cuentas
obtén estructura de cuenta
inicializa apuntadores ctas. de acumulac.
procesa cuentas de afectación
cuenta
numniv
estruct
ctas. acum.
ctas. acum.
numniv
fin lee cuenta
cuenta
ctas. acum.
numniv
fin archivo
actualiza apuntadores ctas. de acumulación
lee cuenta de afectación
ASC
procesa sig. cuenta de afectac.


ASC
fin
actualiza
saldos
actualiza saldos
de cuentas
cuenta
saldo
final
cuenta
saldo
final
cuenta
ctas.
acum.
numniv
obtén saldo final
ctas. afectación
obtén saldo inicial
ctas. afectación
act. saldos cuentas
de acumulación

IV.4 DISEÑO PARA UN LENGUAJE DE TERCERA GENERACION




CS P
calcula saldos de periodo
cuenta en proceso, saldos de movimientos
ultima cta proc., cta. en proceso
fin registra
cuenta en proceso
arreglo cuentas de acum.
calcula saldos movimientos del periodo por cta.
registra saldos de cuentas sin movimientos
actualiza cuentas de acumulación
actualiza acumulados
cuenta en proceso
nivel de acumulación, arreglo cuentas de acumulación
nivel de acumulación, arreglo cuentas de acumulación
arreglo cuentas de acumulación
cuenta en proceso, arreglo cuentas de acum
fin actualiza saldos
saldos de movimientos
fin act. acumulados
determina nivel de acumulación de cuenta en proceso
actualiza arreglo de cuentas de acumulación
ASCM
actualiza acumulados de saldos



IV.5 ESPECIFICACION DEL SISTEMA

En esta sección se detallan las operaciones que deben realizar los procesos que constituyen el sistema. Se indica la localización del proceso mediante la numeración que se le asignó durante el análisis del sistema.

Especificación del módulo Captura datos de póliza

Ubicación : Módulo 1.1.1

Propósito : Validar y almacenar la información relacionada a una póliza que se desea registrar en el sistema .

Recibe : Archivo de tipos de pólizas, archivo de pólizas registradas

Regresa : indicador fin de captura de los datos de la póliza

Detalles funcionales :

Validar tipo de póliza que se está capturando con el archivo de tipos de póliza

Si el tipo de póliza es válido entonces

Determina consecutivo para el tipo de póliza

Fin si

Captura datos restantes de la póliza

Especificación del módulo Captura de movimientos

Ubicación : Módulo 1.1.2

Propósito : registrar los movimientos que constituyen una póliza contable

Recibe : Archivo de movimientos registrados, archivo de cuentas

Regresa : indicador fin de captura de movimientos

Detalles funcionales :

Validar que la cuenta del movimiento exista en el catálogo de cuentas y que no sea una cuenta de acumulación

registrar descripción del movimiento y validar importe del movimiento

Especificación del módulo Valida Póliza

Módulo No. : 1.1.3

Propósito : Verificar que los saldos de cargos y abonos sean iguales para la póliza que se está capturando.

Recibe : Archivo de pólizas, Archivo de movimientos registrados

Regresa : estatus póliza

Detalles funcionales :

total cargos = 0.00

total abonos = 0.00

para todos los movimientos que correspondan al registro de póliza actual ejecuta

total cargos = total cargos + cargos

total abonos = total abonos + abonos

fin para

Si total cargos < > total abonos entonces

estatus póliza=incorrecto

Si no

estatus póliza=correcto

Fin si

Especificación del módulo Actualización consecutivo pólizas

Módulo : 1.1.4

Propósito : Actualizar el archivo de consecutivo de pólizas

Recibe : estatus póliza, archivo de pólizas,archivo de tipos de póliza

Regresa : indicador fín de captura de pólizas

Detalles funcionales :

Si estatus póliza= correcto entonces

si el número de póliza capturada = número de póliza del archivo de tipos de póliza entonces

incrementar en uno el número de póliza del archivo de tipos de póliza

Fin si

Fin si

Especificación del módulo actualización de cargos y abonos por cuenta y mes

Ubicación : módulo 1.3.1

Propósito : registrar en el archivo de saldos de movimientos los acumulados de cargos y abonos por cuenta y mes del archivo de movimientos pendientes de asentar

Recibe : Archivo de movimientos pendientes de asentar

Regresa : Archivo de saldos de movimientos por cuenta y mes actualizado

Detalles funcionales :

Ordena archivo de movimientos pendientes de asentar por cuenta y mes

ir al primer registro

mientras no eof(movtos pendientes)

total cargos = 0.00

total abonos = 0.00

cuenta proceso = cuenta

mes proceso = mes

mientras cuenta = cuenta proceso and mes = mes proceso

total cargos = total cargos + cargos

total abonos = total abonos + abonos

fin mientras

saldo final movtos = total cargos - total abonos

registra saldo final movtos, cuenta y mes en el archivo de saldos de movimientos por cuenta y mes

Especificación del módulo lee saldo de movimientos por cuenta y mes

Ubicación : Módulo 1.3.2

Propósito : lee registro del archivo de saldos de movimientos por cuenta y mes para actualizar el archivo de saldos

Detalles funcionales :

Lee cuenta,mes,saldo final movtos del registro en proceso

Especificación del módulo actualiza saldo final de la cuenta en los periodos afectados

Ubicación : Módulo 1.3.3

propósito : actualizar el saldo final de la cuenta en el mes en el que se realizaron movimientos, así como en los meses posteriores que constituyen el ejercicio contable.

Recibe :

Cuenta : Cuenta que registró movimientos para asentar

saldo final movimientos : saldo de los movimientos a actualizar

mes : mes a partir del cual se actualizarán los saldos

Detalles funcionales :

ir al primer registro del archivo de saldo final movimientos

mientras no eof(saldo final movimientos)

busca registro de saldos para la cuenta

si no existe registro de saldos para la cuenta entonces

crea registro de saldos para la cuenta

inicializa saldos para los periodos del ejercicio = 0.00

fin si

para los meses del ejercicio desde mes hasta el fin del ejercicio ejecuta

saldo final del mes = saldo final del mes+saldo final movimientos

fin para

fin mientras

Especificación del módulo procesa sig. saldo de movimientos por cuenta y mes

Ubicación : Módulo 1.3.4

Propósito : procesa sig. registro del archivo de saldos de movimientos por cuenta y mes para actualizar el archivo de saldos

Recibe : Archivo de saldos de movimientos por cuenta y mes

Regresa : indicador de fin de actualización de movimientos

Detalles funcionales :

avanza al siguiente registro de saldos de movimientos por cuenta y mes

Si ya estamos en el fin de archivo, fin de actualización de movimientos = cierto

Especificación del módulo actualiza archivos de movimientos y pólizas

Ubicación : Módulo 1.3.5

Propósito : Suprimir del archivo de movimientos y pólizas registradas, las pólizas y los movimientos que ya hayan sido asentados.

Recibe : Archivo de pólizas registradas

Archivo de movimientos registrados

Regresa : Archivo de pólizas registradas actualizado

Archivo de movimientos registrados actualizado

Detalles funcionales

Si ya se asentaron los movimientos registrados entonces

copiar pólizas registradas al archivo de pólizas asentadas

borrar las pólizas del archivo de pólizas registradas

copiar los movimientos registrados al archivo de movimientos asentados

borrar los movimientos del archivo de movimientos registrados

Especificación del módulo Obtén parámetros del reporte

Ubicación : Módulo 1.5.1

Propósito : Registrar el rango de cuentas (cuenta inicial, cuenta final) y período para el cual se desea imprimir un estado financiero

Recibe :

Regresa : archivo de parámetros actualizado

Detalles funcionales:

Captura cuenta inicial, cuenta final y período para el reporte

Registra datos en el archivo de parámetros para el reporte

Especificación del módulo obtén movimientos del período

Ubicación : Módulo 1.5.2

Propósito : extraer del archivo de movimientos asentados los movientos necesarios para generar el reporte financiero, de acuerdo a los parámetros capturados en el módulo 1.5.1

Recibe : Cuenta inicial, Cuenta final, Período

Regresa : Archivo de movimientos temporales

Detalles funcionales :

borra archivo de movimientos temporales

Para cada registro del archivo de movimientos asentados revisar

si fecha del movimiento pertenece al período del reporte entonces

si cuenta inicial < = cuenta del movimento y cuenta final > = cuenta del movimiento entonces

registra movimiento en el archivo de movimientos temporales

fin si

fin si

Especificación del módulo actualiza saldos de cuentas acreedoras

Ubicación : Módulo 1.5.5

Propósito : Actualizar los signos de las cuentas acreedoras considerando que disminuyen a las cuentas deudoras, cuando se presentan en un estado financiero

Recibe : archivo de saldos temporales, archivo de cuentas

Regresa : archivo de saldos temporales actualizados

Detalles funcionales :

ir al primer registro de saldos temporales

Para cada registro del archivo de saldos temporales ejecuta

Si la naturaleza de la cuenta del archivo de saldos temporales es acreedora entonces

Saldo inicial cuenta = saldo inicial de la cuenta * - 1

Saldo final cuenta = Saldo final de la cuenta * - 1

fin si

fin para

Especificación del módulo obtén grandes totales

Ubicación : Módulo 1.5.6

propósito : Obtener acumulados de saldos y movimientos para el reporte financiero

Recibe : archivo de saldos temporales

Regresa : archivo de grandes totales actualizado

Detalles funcionales

total inicial = 0

total final = 0

total cargos = 0

total abonos = 0

ir al primer registro del archivo de saldos temporales

mientras no eof(saldos temporales)

total inicial = total inicial + saldo inicial

total final = total final + saldo final

total cargos = total cargos + cargos

total abonos = total abonos + abonos

avanza al siguiente registro

fin mientras

registra grandes totales en el archivo de grandes totales

Especificación del módulo obtén estructura de cuenta

Ubicación : módulo 1.5.4.1

Propósito : almacenar la longitud de cada nivel de una cuenta en un arreglo, este es ocupado para determinar si una cuenta acumula a otra.

Recibe : archivo de estructura de cuenta

Regresa : estruct, arreglo de variables enteras que determina la longitud de cada nivel de una cuenta

Detalles funcionales :

ve al primer registro del archivo de estructura de cuenta

nivel = 1

mientras no eof(estructura de cuenta) ejecuta

estruct(nivel)=Long

Avanza al siguiente registro

fin mientras

Especificación del módulo inicializa apuntadores cuentas de acumulación

ubicación : Módulo 1.5.4.2

Propósito : iniciar el indicador de número de cuentas de acumulación

Recibe :

Regresa : numniv indicador de número de cuentas de acumulación

Detalles funcionales :

numniv = 0

Especificación del módulo actualiza apuntadores de cuentas de acumulación

Ubicación : módulo 1.5.4.4

Propósito : actualizar el arreglo de cuentas de acumulación y el contador de cuentas de acumulación de acuerdo al valor de la cuenta que se está procesando, las cuentas de acumulación que se tienen registradas y al criterio de determinación de cuentas de acumulación

recibe :

nivel_de_acumulación .- contador de cuentas de acumulación

cta_acum .- arreglo de cuentas de acumulación

cuenta .- cuenta de detalle que se está procesando

regresa :

```
    nivel_de_acumulación .- contador de cuentas de acumulación
actualizado
   cta_acum .- arreglo de cuentas de acumulación actualizado
Detalles funcionales
bandacumula=cierto
revniv=falso
Si nivel_de_acumulación =0 entonces
   nivel_de_acumulación =1
   bandacumula=falso
Finsi
cuenta_de_acumulación =1
mientras  bandacumula=cierto  y cuenta_de_acumulación<= nivel_de_acumulación ejecuta
  nivel_en_revisión =1
  nivel_diferente=falso
mientras bandacumula=cierto y nivel_en_revision < número_de_niveles  ejecuta
     Si nivel_en_revisión=1 y cuenta_de_acumulación=1 entonces
         Si cta_acum(1).nivel(1) < > cuenta.nivel(1) entonces
            Si Str$(cta_acum(1).nivel(1) ,i,1 ) < > "0", para cualquier i > 1 o
               Str$(cta_acum(1).nivel(1),1,1) < > Str$(cuenta.nivel(1),1,1) entonces
                 nivel_de_acumulación=1
                  bandacumula=falso
                 loop
              Si no
```

```
            nivel_en_revisión=2
        Fin si
    Fin si
    Si cta_acum(cuenta_de_acumulación).nivel(nivel_en_revisión) < >
cuenta.nivel(nivel_en_revisión) entonces
        Si Str$(cta_acum(cuenta_de_acumulación).nivel(nivel_en_revisión) ,i,1 ) < > "0",
para cualquier i  entonces
            nivel_de_acumulación = cuenta_de_acumulación
            bandacumula=falso
            loop
        Si no
            nivel_diferente= cierto
        Fin si
    Si no
        Si nivel_diferente= cierto entonces
            Si Str$(cta_acum(cuenta_de_acumulación).nivel(nivel_en_revisión) ,i,1 ) < > "0",
para cualquier i entonces
              nivel_de_acumulación=cuenta_de_acumulación
              bandacumula=falso
             loop
            fin si
        fin si
```

Fin si

nivel_en_revisión=nivel_en_revisión+1

fin mientras

cuenta_de_acumulación=cuenta_de_acumulación+1

fin mientras

Si bandacumula=cierto

nivel_de_acumulación=nivel_de_acumulación+1

Fin si

cta_acum(nivel_de_acumulación)=cuenta

Especificación del módulo obtén saldo inicial de la cuenta

Ubicación : Módulo 1.5.4.5.1

Propósito : Calcular el saldo inicial de una cuenta

Recibe : Cuenta, período,saldo final de la cuenta en el período, archivo de saldos de movimientos temporales

Regresa : Saldo inicial de la cuenta

Detalles funcionales

lee cargos y abonos para la cuenta en el periodo

saldo inicial = saldo final + abonos - cargos

Especificación del módulo obtén saldo final de la cuenta

Ubicación : módulo 1.5.4.5.2

Propósito : leer el saldo final de la cuenta para el período seleccionado

Recibe : cuenta y período

Regresa : saldo final de la cuenta

Detalles funcionales :

localiza registro de saldo para la cuenta

lee saldo final de la cuenta en el período seleccionado

Especificación del módulo actualiza saldos de cuentas de acumulación

Ubicación : Módulo 1.5.4.5.3

propósito : acumular los saldos de la cuenta de afectación que se está procesando a las cuentas de acumulación que le corresponden

Recibe : numniv .- indicador de número de cuentas de acumulación

cuentas de acumulación .- arreglo de cuentas de acumulación que le corresponden a la cuenta que se esta procesando.

saldos .- saldos de la cuenta de afectación que se está procesando

Regresa : indicador de actualización de saldos

Detalles funcionales

nivel = 1

mientras nivel < = numniv ejecuta

cuenta de acumulación(nivel).saldo final = cuenta de acumulación(nivel).saldo final + saldo final

cuenta de acumulación(nivel).saldo inicial = cuenta de acumulación(nivel).saldo inicial + saldo inicial

cuenta de acumulación(nivel).cargos = cuenta de acumulación(nivel).cargos + cargos

cuenta de acumulación(nivel).abonos = cuenta de acumulación(nivel).abonos + abonos

```
    nivel = nivel + 1
fin mientras
```

IV.6 CONCLUSIONES :

En el diseño estructurado observamos las mismas diferencias que se señalaron durante el análisis, lo cual es un resultado lógico al considerar que el diseño surge del resultado del análisis realizado.

Se observa en el diseño para un lenguaje de tercera generación un mayor acoplamiento entre las operaciones que constituyen el proceso, esto es debido a que se transmiten una mayor cantidad de datos las operaciones como resultado de cálculos intermedios. Esto dificulta el desarrollo y mantenimiento, pues para realizarlo deben tenerse en cuenta estos datos y su relación con otras operaciones, para no generar fallas de operación inesperadas como resultado de estas actividades.

En el diseño para un lenguaje de cuarta generación el acoplamiento entre las operaciones es menor, lo cual se refleja en una menor cantidad de datos que son transmitidos entre las operaciones. Esta característica, que es favorable pues da mayor independencia a las operaciones (facilitando por lo tanto el desarrollo y mantenimiento de éstas), es debida a que el uso del lenguaje de consulta (Query) substituirá el acoplamiento de datos por un acoplamiento de base de datos.

V. DESARROLLO DEL SISTEMA

En el presente capítulo se muestra la programación del sistema, para esto se desarrollan algunos módulos del sistema bajo estudio, el desarrollo del sistema corresponde a la conversión que se realiza de la especificación dada en la sección IV.5; las operaciones que constituyen al sistema son convertidas en programas. El desarrollo fue realizado con Paradox 3.5, se utilizan sus herramientas de cuarta generación. Se desarrollan a la vez algunas operaciones con su lenguaje de programación como lenguaje de tercera generación, para establecer las diferencias que surgen durante esta etapa.

V.1 DESARROLLO DE LA PROGRAMACION

Módulo 1.3.1 Actualiza cargos y abonos por cuenta y mes

```
Ordena archivo de movimientos pendientes de asentar por cuenta y mes
 ir al primer registro
 mientras no eof(movtos pendientes)
      total cargos = 0.00; total abonos = 0.00
      cuenta proceso = cuenta
      mes proceso = mes
      mientras cuenta = cuenta proceso and mes = mes proceso
         total cargos = total cargos + cargos
         total abonos  = total abonos + abonos
```

fin mientras

saldo final movtos = total cargos - total abonos

registra saldo final movtos, cuenta y mes en el archivo de saldos de movimientos por cuenta y mes

Este módulo puede ser totalmente desarrollado mediante un query de la siguiente forma :

Query

Movtos	Mes	Num-cta	Cargo	Abono
	Check	Check	CALC SUM AS Cargo	CALC SUM as Abono
Movtos				

Endquery

Este es un ejemplo de un query by example que toma como entrada de datos a la tabla Movtos, esto es indicado en el primer renglón del query, los nombres de los campos de esta tabla también se indican en la primera línea. La operación que se realiza con este query está indicada en el segundo renglón, el operador Check habilita a los campos Mes y Num-cta para aparecer en el resultado del query, mientras que el operador CALC SUM AS acumula los valores del campo en el cual está situado como función sumaria, la acumulación es realizada de acuerdo a los campos que tengan el operador Check. En este caso calculará los acumulados de cargos y abonos por mes y cuenta. Un comando SQL equivalente sería :

```
SELECT MES,NUM-CTA,SUM(CARGO),SUM(ABONO) FROM MOVTOS
        GROUP BY MES, NUM-CTA
```

El mismo módulo con un lenguaje de tercera generación :

```
Proc Acumula()
  MoveTo "Movtos"
  If IsEmpty("Movtos") Then
    Return
  EndIf
  Sort "movtos" on "Num-cta1","Num-cta2","Num-cta3","Fech1-pol"
  Home
  EditKey
  Num_cta=[Num-cta1]+[Num-cta2]+[Num-cta3]
  Fecha=Month([Fech-pol])
  While Not Eot()
     Saldt=0
     While Num_cta=[Num-cta1]+[Num-cta2]+[Num-cta3] And
      Fecha=Month([Fech-pol]) And Not Eot()
        If Abs([Cargo])>0.001 Then
           Saldt=saldt+[Cargo]
        Else
           Saldt=Saldt-[Abono]
        EndIf
        Skip 1
      EndWhile
```

```
        Act_sdo()
        If [Num-cta1]+[Num-cta2]+[Num-cta3]<>"" Then
           Num_cta=[Num-cta1]+[Num-cta2]+[Num-cta3]
           Fecha=Month([Fech-pol])
        EndIf
    EndWhile
    Do_it!
    Add "Movtos_a" "Movtos_f"
    Empty "Movtos_a"
EndProc
```

Modulo 1.3.3 Actualiza saldo final de la cuenta en los meses afectados

ir al primer registro del archivo de saldo final movimientos

mientras no eof(saldo final movimientos)

busca registro de saldos para la cuenta

si no existe registro de saldos para la cuenta entonces

crea registro de saldos para la cuenta

inicializa saldos para los periodos del ejercicio = 0.00

fin si

para los meses del ejercicio desde mes hasta el fin del ejercicio ejecuta

saldo final del mes = saldo final del mes+saldo final movimientos

fin para

fin mientras

Este módulo es desarrollado de la siguiente manera :

```
Tsald="Answer"
View "Saldos"
 View Tsald
 Home
 EditKey
While Not Eot()
   Num_cta=[Num-Cta]
   Fecha=[Mes]
   Saldt=0
   If Not IsBlank([Cargo]) Then
     Saldt=saldt+[Cargo]
   EndIf
   If Not IsBlank([Abono]) Then
     Saldt=Saldt-[Abono]
   EndIf
   Skip 1
   Act_sdo()
EndWhile
Do_it!
Add "Movtos" "Movtos_f"
Empty "Movtos"
```

```
  Add "Polizas" "Pol_Asen"
  Empty "Polizas"
EndProc
Proc Act_sdo()
  MoveTo "Saldos"
  MoveTo Field "Num-cta"
  Locate Num_cta
  If Not Retval  Then
    Crea_cta()
  EndIf
  Ac_mes()
  MoveTo TSald
EndProc
Proc Ac_mes()
  For Mes From Fecha To 13
     Mesf="Saldo"+Strval(Mes)
     MoveTo field Mesf
     []=[]+Saldt
  EndFor
Endproc
Proc Crea_cta()
  End
```

```
  If [Num-cta]<>"" Then
    Down
  EndIf
  [Num-cta]=Num_cta
EndProc
Play "PQSalCta"
Acumula()
Reset
{Scripts} {Play} {Qsalmcta} Do_It! Menu {Scripts} {End-Record}
```

El desarrollo de este módulo es basicamente el mismo utilizando un lenguaje de tercera o de cuarta generación, esto es debido a que las operaciones que se van a realizar no se ajustan a las herramientas del lenguaje de cuarta generación. La diferencia en el desarrollo entre los lenguajes es que, con el lenguaje de cuarta generación al utilizar un query para desarrollar el módulo 1.3.1. el resultado es una tabla, que será utilizada como entrada de datos para el módulo 1.3.3., esta tabla contiene los saldos de los movimientos totalizados por cuenta y mes. Con un lenguaje de tercera generación el desarrollo del módulo 1.3.1. requiere la programación de la operación para obtener el saldo de los movimientos por cuenta y mes; para que una vez obtenido éste sea transmitido como parte de los datos de entrada para ejecutar el módulo 1.3.3

Módulo 1.5.2. Obtén movimientos del período

Borra archivo de movimientos temporales

Para cada registro del archivo de movimientos asentados revisar

si fecha del movimiento pertenece al período del reporte entonces

si cuenta inicial < = cuenta del movimiento y cuenta final > = cuenta del movimiento entonces

registra movimiento en el archivo de movimientos temporales

fin si

fin si

Este módulo puede ser desarrollado mediante un query de la siguiente manera :

Query

T_ctasp | Cta_ini | Cta_fin | Periodo |

| _cta1 | _cta2 | _per |

Movtos_f | Mes | Num-cta | Cargo |

Check _per | Check >=_cta1,<=_cta2 | CALC SUM AS Cargo |

Movtos_f | Abono |

| CALC SUM AS Abono

Endquery

La entrada del query está dada por las tablas T_ctasp y Movtos_f, los campos de la tabla T_ctasp determinan el rango de las cuentas para el cual se genera el reporte (cuenta inicial, cuenta final) y el mes, la tabla Movtos_f contiene los movimientos que han sido asentados en el sistema, el guión bajo que se observa en las

etiquetas cta1, cta2 y per las hacen actuar como variables para determinar que registros se extraen de la tabla Movtos_f. Como se puede observar ningún campo de la tabla T_ctasp tiene el operador Check, por lo que de esta tabla no se obtendrá ningún valor para el resultado del query; de la tabla Movtos_f obtendremos los valores del resultado del query. El operador Check ocasiona la aparición del mes y el número de cuenta, en el campo mes se puede observar la aparición de la etiqueta _per; esto ocasiona una selección de los registros del archivo Movtos_f, esta selección incluye solo aquellos registros cuyo valor en el campo Mes sea igual al valor del campo Período de la tabla T_ctasp. En el campo Num-cta la etiqueta >=_cta1,<=_cta2 ocasiona otra condición, en este caso solo se incluyen aquellos registros cuyos valores en el campo Numcta estén dentro del rango cta-ini a cta-fin. El operador CALC SUM AS acumulará los cargos y abonos de los registros que sean seleccionados del archivo Movtos_f. Un comando SQL equivalente sería

SELECT MES,NUM-CTA,SUM(CARGOS),SUM(ABONOS) FROM MOVTOS_F, T_CTASP

WHERE NUM-CTA BETWEEN CTA-INI AND CTA-FIN AND MES=PERIODO

El mismo módulo con un lenguaje de tercera generación :

```
Proc Closed Mov_t(Mes)
  Array RegMov[15]
  View "Movtos_f"
  EditKey
  Scan For Month([Fech-Pol])=Mes
```

```
  CopyToArray RegMov
  MoveTo "Movtos_t"
  End
  If [Num-Cta1]+[Num-Cta2]+[Num-Cta3]<>""
    Then
      Down
  EndIf
  CopyFromArray RegMov
  MoveTo "Movtos_f"
EndScan
Do_it!
ClearImage
EndProc
```

Módulo 1.5.4.5.2. Obtén saldo final cuentas de afectación

Leer el saldo final de la cuenta para el período seleccionado

localiza registro de saldo para la cuenta

lee saldo final de la cuenta en el período seleccionado

Qctarpt.sc

Query

Cuentas | Num-cta |

| Check >=_cta1,<=_cta2 |

T_ctasp | Cve_cia | Cta_ini | Cta_fin |

| Check | _cta1 | _cta2 |

Endquery

Pqctarpt.sc

{Scripts} {Play} {Qctarpt} Do_It!

Copy "Answer" "t_ctas"

Este query extrae las cuentas para las cuales se genera la balanza, tomando como fuente de información la tabla Cuentas. Los campos de los que consta la tabla de resultados son cve_cia y Num-cta. Las cuentas que se extraen del archivo Cuentas dependen de los valores de cta1 y cta2 del archivo T_ctasp. Un comando SQL equivalente sería:

SELECT CUENTAS.NUM-CTA,T_CTASP.CVE_CIA FROM CUENTAS,T_CTASP WHERE

CUENTAS.NUM-CTA>=T_CTASP.CTA_INI AND

CUENTAS.NUM-CTA<=T_CTASP.CTA_FIN

El programa Pqctarpt.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla t_ctas.

Qsalcm01.sc

Query

T_ctasp | Cta_ini | Cta_fin | Periodo |

| _cta1 | _cta2 | _per |

Saldos | Num-cta | Saldo1 |

| Check >=_cta1,<=_cta2 | Check As Saldof |

Endquery

Pqsalm01.sc

{Scripts} {Play} {Qsalcm01} Do_It!

Copy "Answer" "Qsaldos"

Este query toma como entrada la tabla T_ctasp y al archivo de saldos, mediante este query se extrae el saldo del período para las cuentas que integran el reporte. La tabla de resultados consiste de los campos Num-cta y saldof, el último es definido como alias del campo Saldo1. Se ejecuta un query diferente dependiendo del período para el cual se esté elaborando el reporte, pues dependiendo del período se seleccionará un campo diferente (Saldo1, Saldo2, Saldo3, etc.). Sin embargo, al marcar este campo con el mismo alias en la tabla de resultados, obtendremos una tabla con la misma estructura, sin importar el período para el cual se haya ejecutado.

El programa Pqsalm01.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla Qsaldos.

Módulo 1.5.4.5.1. Obtén saldo inicial de la cuenta

lee cargos y abonos para la cuenta en el periodo

saldo inicial = saldo final + abonos - cargos

Qtsalf.sc

Query

T_salctr | Num-cta | Cve_cia | Cargo | Abono |

| Check _cta! | Check | Check _car | Check _abo |

Qsaldos | Num-cta | Saldof |

| _cta | Check _sal |

Endquery

Pqtsalf.sc

{Scripts} {Play} {Qtsalf} Do_It!

Copy "Answer" "t_qsalf"

Este query toma como entrada a las tablas T_salctr y Qsaldos, la tabla T_salctr contiene las cuentas comprendidas en el rango de cuentas solicitadas para el reporte, totales de cargos , abonos, y clave de la compañía para la que se está elaborando el reporte; la tabla Qsaldos contiene los saldos finales de las mismas cuentas. El query se encarga de generar una tabla que relacione los totales de los movimientos y el saldo final basado en el número de cuenta, el signo de admiración que aparece en el campo Num-cta de la tabla T_salctr habilita a aparecer en la tabla de resultados a las cuentas que se encuentren en la tabla T_salctr, aun aquellas que no tengan un registro en la tabla Qsaldos; es

decir las cuentas que no hayan tenido ningún movimiento durante el ejercicio.

El programa Pqtsalf.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla tqsalf.

Qtsald.sc

Query

T_qsalf | Num-cta | Cve_cia | Cargo | Abono | Saldof |

| Check | Check | Check _car | Check _abo | Check _salf,Calc _salf+_abo-_car As Saldoi |

Endquery

Pqtsald.sc

{Scripts} {Play} {Qtsald} Do_It! Copy "Answer" "t_qsald"

Este query toma como entrada a la tabla t_qsalf que contiene el saldo final, total de cargos y abonos de las cuentas de afectación necesarias para generar el reporte, el query calcula el saldo inicial para estas cuentas y lo agrega como un campo adicional a la tabla de resultados con el nombre saldoi.

El programa Pqtsald.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla tqsald.

Módulo 1.5.5. Actualiza saldos de cuentas acreedoras

ir al primer registro de saldos temporales

Para cada registro del archivo de saldos temporales ejecuta

Si la naturaleza de la cuenta del archivo de saldos temporales es acreedora entonces

Saldo inicial de la cuenta = saldo inicial de la cuenta * - 1

Saldo final de la cuenta = Saldo final de la cuenta * - 1

fin si

fin para

Este módulo podría ser implementado en un lenguaje de cuarta generación en esta forma :

Qtsala.sc

Query

T_qsald | Num-cta | Cve_cia | Cargo | Abono | Saldof |

| Check _cta | Check | Check | Check | _salf,Calc _salf*-1 As Saldof |

T_qsald | Saldoi |

| _sali,CALC _sali*-1 As Saldoi |

Cuentas | Num-cta | Nat-cta |

| _cta | A |

Endquery

Pqtsala.sc

{Scripts} {Play} {Qtsala} Do_It!

Copy "Answer" "t_qsala"

Este query toma como entrada a las tablas T_qsald y cuentas, entre estas tablas se establece una relación mediante la variable cta, sólo se consideran las cuentas que tengan valor "A" en el campo Nat-cta (Naturaleza acreedora), es decir, las cuentas de la tabla T_qsald que son de naturaleza acreedora. En el campo saldof y saldoi se observa cómo se define el valor del campo como una variable (_salf,_sali), mediante la operación (*-1) se le cambia el signo al campo.

El programa Pqtsala.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_qsala.

Qtsaldd.sc

Query

T_qsald | Num-cta | Cve_cia | Cargo | Abono | Saldof | Saldoi |

| Check _cta | Check | Check | Check | Check | Check |

Cuentas | Num-cta | Nat-cta |

| _cta | D |

Endquery

Pqtsaldd.sc

Menu {Scripts} {Play} {Qtsaldd} Do_It!

Copy "Answer" "t_qsald"

Este query toma como entrada a las tablas T_qsald y cuentas, entre estas tablas se establece una relación mediante la variable cta, sólo se considerarán las cuentas que tengan valor "D" en el campo Nat-cta (Naturaleza deudora), es decir, las cuentas de la tabla T_qsald que son de naturaleza deudora. Los signos de los saldos de estas cuentas son correctos, por lo cual no tiene que aplicárseles ninguna operación.

El programa Pqtsaldd.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_qsald.

Módulo 1.5.6. Obtén grandes totales

total inicial = 0

total final = 0

total cargos = 0

total abonos = 0

ir al primer registro del archivo de saldos temporales

mientras no eof(saldos temporales)

    total inicial = total inicial + saldo inicial

    total final = total final + saldo final

    total cargos = total cargos + cargos

    total abonos = total abonos + abonos

    avanza al siguiente registro

fin mientras

registra grandes totales en el archivo de grandes totales

Este módulo puede ser implementado en un lenguaje de cuarta generación en esta forma :

Qtsabo.sc

Query

T_qsala | Num-cta | Cve_cia | Saldof |

| _cta | Check | Calc SUM As TotAbo |

Cuentas | Num-cta | Acum |

| _cta | N |

Endquery

Pqtsabo.sc

{Scripts} {Play} {Qtsabo} Do_It!

Empty "t_qacabo"

Add "Answer" "t_qacabo"

Este query toma como entrada a las tablas T_qsala y cuentas, entre estas tablas se establece una relación mediante la variable cta, sólo se consideran las cuentas que tengan valor "N" en el campo Acum (cuentas de afectación) y se acumula el saldo final de estas cuentas.

El programa Pqtsabo.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_qacabo.

Qtscar.sc

Query

T_qsald | Num-cta | Cve_cia | Saldof |

| _cta | Check | Calc Sum As TotCar |

Cuentas | Num-cta | Acum |

| _cta | N |

Endquery

Pqtscar.sc

Menu {Scripts} {Play} {Qtscar} Do_It!

Empty "t_qaccar"

Add "Answer" "t_qaccar"

Este query toma como entrada a las tablas T_qsald y cuentas, entre estas tablas se establece una relación mediante la variable

cta, sólo se consideran las cuentas que tengan valor "N" en el campo Acum (cuentas de afectación) y se acumula el saldo final de estas cuentas.

El programa Pqtscar.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_qaccar.

Qasfc.sc

Query

T_qsald | Num-cta | Cve_cia | Cargo | Abono |

| _cta | Check | Calc Sum As Cargo | Calc Sum As Abono |

Cuentas | Num-cta | Acum |

| _cta | N |

Endquery

Pqasfc.sc

{Scripts} {Play} {Qasfc} Do_It!

Empty "T_tcarab"

Add "Answer" "T_tcarab"

Este query toma como entrada a las tablas T_qsald y cuentas, entre estas tablas se establece una relación mediante la variable cta, sólo se consideran las cuentas que tengan valor "N" en el campo Acum (cuentas de afectación) y se acumulan cargos y abonos de estas cuentas.

El programa Pqasfc.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_tcarab.

Qacsal.sc

Query

T_qacabo | Cve_cia | TotAbo |

| Check _cia | Check |

T_qaccar | Cve_cia | TotCar |

| _cia | Check |

T_tcarab | Cve_cia | Cargo | Abono |

| _cia | Check | Check |

Endquery

Pqacsal.sc

{Scripts} {Play} {Qacsal} Do_It! Menu {Scripts} {End-Record}

Copy "Answer" "t_saltac"

Este query toma como entrada a las tablas T_qacabo, T_qaccar, T_tcarab y las une basado en el campo Cve_cia, como resultado se crea una sola tabla que contenga los acumulados de saldos finales de cargos, saldos finales abonos, así como acumulados de cargos y abonos del periodo.

El programa Pqacsal.sc es una macro generada para ejecutar el query y copiar el resultado a la tabla T_saltac.

V.2 PANTALLAS DEL SISTEMA

Pantalla para Pólizas

Changing F form for Polizas Form 1/1
< 1, 1>

C Y M Asesores en Soluciones Computacionales S.A. de C.V.

Av. Santa Ursula 177 T 2 - 102 Col. santa Ursula Xitla R.F.C. 911022-CMA

C A P T U R A D E P O L I Z A S

Fecha _____ Tipo _ No. Pol ____ Mes ___ Ref. _____
Importe _____ Descripción _____

Ref	Cuenta	Descripción Movimiento	Movimientos Cargo	Abono
___	_____	_____	_____	_____
___	_____	_____	_____	_____
___	_____	_____	_____	_____
___	_____	_____	_____	_____

La pantalla se desarrolla mediante el generador de formas, en este caso se relaciona las tablas pólizas y movimientos. Al momento de especificar las tablas se define cual es la maestra mediante los campos que establecen la relación entre ellas, esta relación determina también la cardinalidad de la relación. En esta pantalla la relación la establecen los campos No. de póliza,

tipo de póliza y mes de la póliza que constituye la llave de la tabla pólizas; estos campos forman parte también de la llave de la tabla movimientos. La tabla de movimientos posee además como campos llave al número de movimiento y al número de cuenta, en base a esto se determina una relación 1:n entre las tablas. Al momento de la corrida la forma se ve así :

C Y M Asesores en Soluciones Computacionales S.A. de C.V.

Av. Santa Ursula 177 T 2 - 102 Col. santa Ursula Xitla R.F.C. 911022-CMA

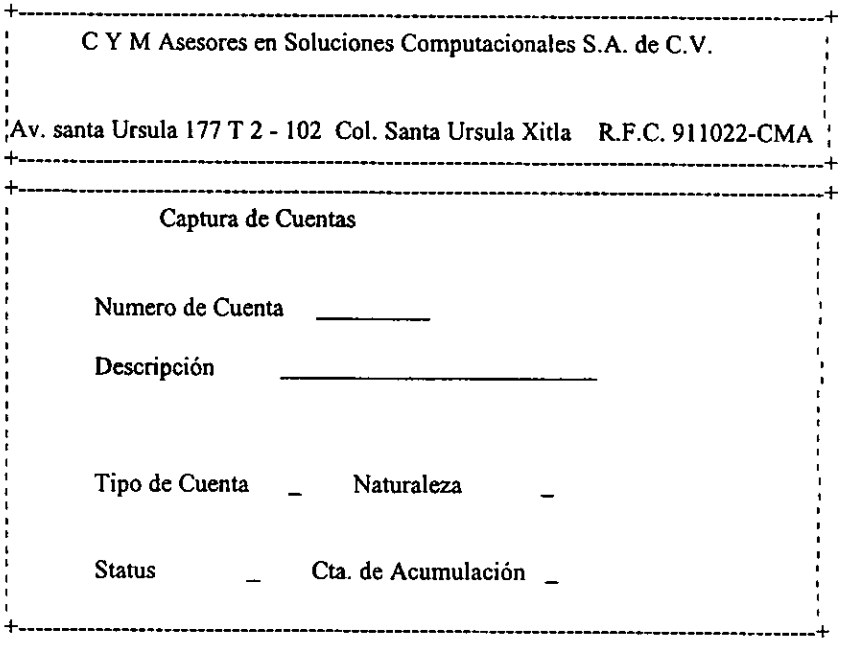
C A P T U R A D E P O L I Z A S

Fecha 25/8/93 Tipo D No. Pol 4 Mes 8 Ref. d004
Importe 100 Descripción abono I.V.A. agosto

Ref	Cuenta	Descripción Movimiento	Movimientos Cargo	Abono
1	110-01-01	pago I.V.A agosto Cta. Cheques Serfin	100	0
2	110-04-00	pago I.V.A agosto I.V.A. Acreditable	0	100

Pantalla para cuentas

Changing F1 form for Cuentas Form 1/1
< 1, 1>



Esta pantalla sólo consiste de la tabla cuentas, en este caso para el desarrollo de la pantalla unicamente se requiere especificar la tabla que se actualiza, situar los campos en la forma, anexar etiquetas y líneas para obtener la forma. En operación la forma se ve así :

C Y M Asesores en Soluciones Computacionales S.A. de C.V.

Av. santa Ursula 177 T 2 - 102 Col. Santa Ursula Xitla R.F.C. 911022-CMA

Captura de Cuentas

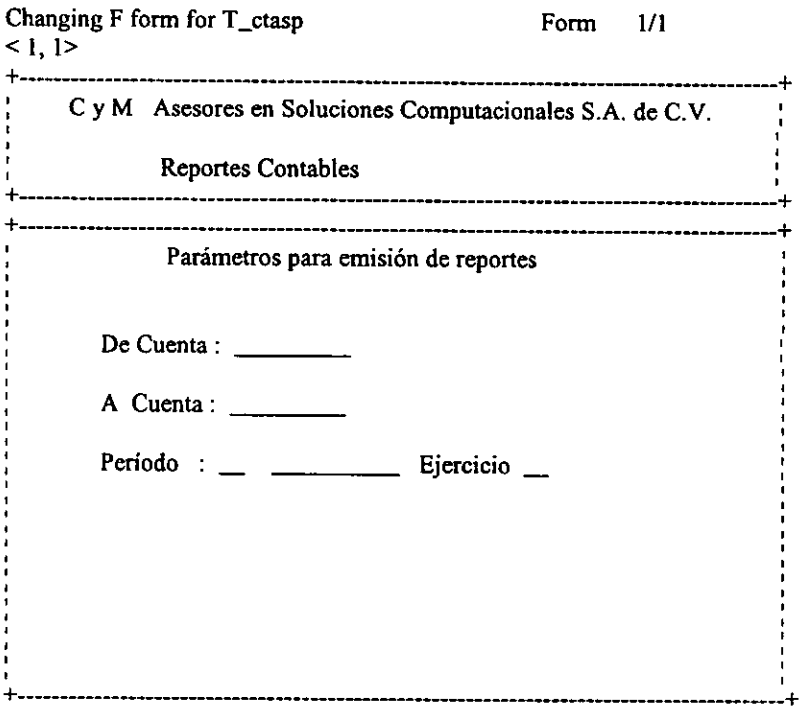
Número de Cuenta 110-04-00

Descripción I.V.A. Acreditable

Tipo de Cuenta A Naturaleza D

Status A Cta. de Acumulación N

Pantalla para reportes contables



Esta pantalla se desarrolla para obtener los parámetros para la emisión del reporte, se establece relación con la tabla cuentas para que el usuario la pueda accesar durante su ejecución, de esta forma se permite al usuario accesar la tabla cuentas para definir el rango de cuentas. Se establece relación al mismo tiempo con una tabla de nombres de meses para describir el mes para el que se generará el reporte, en operación la pantalla se ve así :

C y M Asesores en Soluciones Computacionales S.A. de C.V.

Reportes Contables

Parámetros para emisión de reportes

De Cuenta : 100-00-00

A Cuenta : 460-02-00

Periodo : 8 Agosto Ejercicio 94

V.3 REPORTES DEL SISTEMA

```
Changing report R1 for T_rbal table                Report  1/1
Report Header
....+...10....+...20....+...30....+...40....+...50....+...60....+...70....+...80....+...90....+...100....+...110....+...120....+...130.*

-page-------------------------------------------------------------------------------------------------------------------------------

Month dd, yyyy          AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA                    Pag.999
                          Balanza de Comprobación
                        del mes de  AAAAAAAAAA de 99

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAA AAAAAAAAAA A
=========================================================================================================

                                                    Movimientos del período     Saldos del período
Cuenta        Descripción                  Nat  Saldo Inicial  Deudor  Acreedor   Deudor   Acreedor
=========================================================================================================
---group Cve_cia-------------------------------------------------------------------------------------------

+table----------------------------------------------------------------------------------------------------

AAAAAAAAA  AAAAAAAAAAAAAAAAAAAAAAAAAA  A  999999999.99 999999999.99 999999999.99 999999999.99 999999999.99
A

+table----------------------------------------------------------------------------------------------------
                        TOTALES            9999999999.99 999999999.99 999999999.99 999999999.99
---group Cve_cia-------------------------------------------------------------------------------------------
```

Este reporte se genera indicando la tabla a la que está asociado (T_rbal), se situan los campos de la tabla en el esquema (los conjuntos de 9´s y de A´s que se observan en el reporte), así como las etiquetas para hacer comprensibles las columnas y de los encabezados para darle presentación al reporte. Los resultados de la ejecución del reporte son los siguientes :

Balanza de Comprobación
del mes de Agosto de 94

Av. Sta. ursula 177 T 2 - 102 Sta. Ursula Xitla 14420 D. F. Mexico

Cuenta	Descripción	Nat	Saldo Inicial	Movimientos del período Deudor	Acreedor	Saldos del periodo Deudor	Acreedor
100-00-00	Activo	D	0.00	2200.00	2200.00	0.00	0.00
110-00-00	Activo Circulante	D	0.00	1100.00	2200.00	0.00	1100.00 *
110-01-00	Bancos	D	0.00	1100.00	2200.00	0.00	1100.00 *
110-01-01	Cta. Cheques Serfin	D	0.00	1100.00	2200.00	0.00	1100.00 *
110-02-00	Fondo Fijo caja	D	0.00	0.00	0.00	0.00	0.00
110-03-00	Clientes	D	0.00	0.00	0.00	0.00	0.00
110-03-01	Clientes de mostrador	D	0.00	0.00	0.00	0.00	0.00
110-03-02	Servicios Corporativos	D	0.00	0.00	0.00	0.00	0.00
110-03-03	Eurística Ingeniería y si	D	0.00	0.00	0.00	0.00	0.00
110-04-00	I.V.A. Acreditable	D	0.00	0.00	0.00	0.00	0.00
110-05-00	Impuestos Anticipados	D	0.00	0.00	0.00	0.00	0.00
110-05-01	I.S.R.	D	0.00	0.00	0.00	0.00	0.00
110-05-02	Otros imptos. Anticipad	D	0.00	0.00	0.00	0.00	0.00
110-06-00	Anticipos a Proveedores	D	0.00	0.00	0.00	0.00	0.00
110-07-00	Accionistas	D	0.00	0.00	0.00	0.00	0.00
110-08-00	Deudores Diversos	D	0.00	0.00	0.00	0.00	0.00
110-08-01	Notario Alfredo Ruiz	D	0.00	0.00	0.00	0.00	0.00
120-00-00	Activo Fijo	D	0.00	1100.00	0.00	1100.00	0.00
120-01-00	Eq. de Cómputo	D	0.00	0.00	0.00	0.00	0.00
120-02-00	Depr. Eq. de Cómputo	A	0.00	1100.00	0.00	1100.00	0.00 *
120-03-00	Mob. y Eq. de Ofna.	D	0.00	0.00	0.00	0.00	0.00
120-04-00	Depr. Mob. Y Equipo	A	0.00	0.00	0.00	0.00	0.00
120-05-00	Eq. de Transporte	D	0.00	0.00	0.00	0.00	0.00
120-06-00	Depr. de Eq. de Transport	A	0.00	0.00	0.00	0.00	0.00
130-00-00	Activo Diferido	D	0.00	0.00	0.00	0.00	0.00
130-01-00	Depositos en Garantía	D	0.00	0.00	0.00	0.00	0.00
130-02-00	Gastos de Instalación	D	0.00	0.00	0.00	0.00	0.00
130-03-00	Amort. Gastos de Instalac	A	0.00	0.00	0.00	0.00	0.00
200-00-00	Pasivo	A	0.00	0.00	0.00	0.00	0.00
210-00-00	Pasivo Circulante	A	0.00	0.00	0.00	0.00	0.00
210-01-00	Proveedores	A	0.00	0.00	0.00	0.00	0.00
210-01-01	M.P.S. Mayorista	A	0.00	0.00	0.00	0.00	0.00
210-01-02	Vital Electrónica S.A.	A	0.00	0.00	0.00	0.00	0.00
210-01-03	Dimensión Digital S.A.	A	0.00	0.00	0.00	0.00	0.00
210-01-04	O.C.I. S.A.	A	0.00	0.00	0.00	0.00	0.00
210-01-05	Computel S.A.	A	0.00	0.00	0.00	0.00	0.00
210-02-00	Acreedores Diversos	A	0.00	0.00	0.00	0.00	0.00
210-02-01	Marco Aurelio Reyna P	A	0.00	0.00	0.00	0.00	0.00
210-02-02	Roberto Aguilar Esquivel	A	0.00	0.00	0.00	0.00	0.00
210-02-03	Telmex S.A.	A	0.00	0.00	0.00	0.00	0.00
210-03-00	Impuestos por pagar	A	0.00	0.00	0.00	0.00	0.00
210-03-01	I.V.A. por pagar	A	0.00	0.00	0.00	0.00	0.00
210-03-02	10% Sobre honorarios	A	0.00	0.00	0.00	0.00	0.00
210-04-00	Anticipos de Clientes	A	0.00	0.00	0.00	0.00	0.00

Balanza de Comprobación
del mes de Agosto de 94

Av. Sta. ursula 177 T 2 - 102 Sta. Ursula Xitla 14420 D. F. Mexico

Cuenta	Descripción	Nat	Saldo Inicial	Movimientos del período Deudor	Movimientos del período Acreedor	Saldos del período Deudor	Saldos del período Acreedor
210-04-01	Clientes Varios	A	0.00	0.00	0.00	0.00	0.00
300-00-00	Capital Contable	A	0.00	0.00	0.00	0.00	0.00
300-01-00	Capital Social	A	0.00	0.00	0.00	0.00	0.00
300-02-00	Resultado Ej. Anterior	A	0.00	0.00	0.00	0.00	0.00
300-03-00	Resultados del Ejercicio	A	0.00	0.00	0.00	0.00	0.00
400-00-00	Resultados	D	0.00	0.00	0.00	0.00	0.00
410-00-00	Ingresos	A	0.00	0.00	0.00	0.00	0.00
410-01-00	Ventas	A	0.00	0.00	0.00	0.00	0.00
410-02-00	Ingresos por servicios	A	0.00	0.00	0.00	0.00	0.00
410-03-00	Productos Financieros	A	0.00	0.00	0.00	0.00	0.00
420-00-00	Desctos. y Devoluciones	D	0.00	0.00	0.00	0.00	0.00
430-00-00	Compras	D	0.00	0.00	0.00	0.00	0.00
430-01-00	Computadoras	D	0.00	0.00	0.00	0.00	0.00
430-02-00	Dispositivos	D	0.00	0.00	0.00	0.00	0.00
430-03-00	Consumibles	D	0.00	0.00	0.00	0.00	0.00
440-00-00	Devoluciones y Desctos	A	0.00	0.00	0.00	0.00	0.00
450-00-00	Gastos Generales	D	0.00	0.00	0.00	0.00	0.00
450-01-00	Papeleria y Artículos de	D	0.00	0.00	0.00	0.00	0.00
450-02-00	Honorarios	D	0.00	0.00	0.00	0.00	0.00
450-03-00	Transportes	D	0.00	0.00	0.00	0.00	0.00
450-04-00	Sueldos y salarios	D	0.00	0.00	0.00	0.00	0.00
450-05-00	Viáticos	D	0.00	0.00	0.00	0.00	0.00
450-06-00	Energia eléctrica	D	0.00	0.00	0.00	0.00	0.00
450-07-00	Impuestos	D	0.00	0.00	0.00	0.00	0.00
450-08-00	Manto. Automóvil	D	0.00	0.00	0.00	0.00	0.00
450-09-00	Teléfono	D	0.00	0.00	0.00	0.00	0.00
450-10-00	Depr eq. de computo	D	0.00	0.00	0.00	0.00	0.00
450-11-00	Depr. Mob. y eq. de Ofna	D	0.00	0.00	0.00	0.00	0.00
450-12-00	Varios	D	0.00	0.00	0.00	0.00	0.00
450-13-00	Arrendamiento de oficina	D	0.00	0.00	0.00	0.00	0.00
450-14-00	Atención a Clientes	D	0.00	0.00	0.00	0.00	0.00
450-15-00	Amort. Gtos. De Instalac.	D	0.00	0.00	0.00	0.00	0.00
460-00-00	Gastos Financieros	D	0.00	0.00	0.00	0.00	0.00
460-01-00	Comisiones Bancarias	D	0.00	0.00	0.00	0.00	0.00
460-02-00	Comisiones Cheques de	D	0.00	0.00	0.00	0.00	0.00
	TOTALES		2200.00	2200.00	-1100.00	-1100.00	

```
Changing report R for Cuentas table                    Report   1/2
Report Header
....+...10....+...20....+...30....+...40....+...50....+...60....+...70....+...8*

-page-------------------------------------------------------------------------

dd/mm/yy                    Catálogo de Cuentas                    Pag. 999

+ta-le------------------------------------------------------------------------

 No. Cta.   Tipo Acum Nat     D e s c r i p c i ó n                      Stat.

 AAA AA AA  A   A   A AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA A
+ta-le------------------------------------------------------------------------


-page-------------------------------------------------------------------------

-------------------------------------------------------------------------------
```

Este reporte se genera indicando la tabla a la que está asociado (T_rbal), se situan los campos de la tabla en el esquema (los conjuntos de 9´s y de A´s que se observan en el reporte), así como las etiquetas para hacer comprensibles las columnas y los encabezados. El resultado de la ejecución del reporte es el siguiente :

No. Cta.	Tipo	Acum	Nat	Descripción	Stat.
100-00-00	A	S	D	Activo	A
110-00-00	A	S	D	Activo Circulante	A
110-01-00	A	S	D	Bancos	A
110-01-01	A	N	D	Cta. Cheques Serfin	A
110-02-00	A	N	D	Fondo Fijo caja	A
110-03-00	A	S	D	Clientes	A
110-03-01	A	N	D	Clientes de mostrador	A
110-03-02	A	N	D	Servicios Corporativos Fo	A
110-03-03	A	N	D	Eurística Ingeniería y si	A
110-04-00	A	N	D	I.V.A. Acreditable	A
110-05-00	A	S	D	Impuestos Anticipados	A
110-05-01	A	N	D	I.S.R.	A
110-05-02	A	N	D	Otros imptos. Anticipados	A
110-06-00	A	S	D	Anticipos a Proveedores	A
110-07-00	A	N	D	Accionistas	A
110-08-00	A	S	D	Deudores Diversos	A
110-08-01	A	N	D	Notario Alfredo Ruiz	A
120-00-00	A	S	D	Activo Fijo	A
120-01-00	A	N	D	Eq. de Cómputo	A
120-02-00	A	N	A	Depr. Eq. de Cómputo	A
120-03-00	A	N	D	Mob. y Eq. de Ofna.	A
120-04-00	A	N	A	Depr. Mob. Y Equipo	A
120-05-00	A	N	D	Eq. de Transporte	A
120-06-00	A	N	A	Depr. de Eq. de Transport	A
130-00-00	A	S	D	Activo Diferido	A
130-01-00	A	N	D	Depositos en Garantía	A
130-02-00	A	N	D	Gastos de Instalación	A
130-03-00	A	N	A	Amort. Gastos de Instalac	A
200-00-00	P	S	A	Pasivo	A
210-00-00	P	S	A	Pasivo Circulante	A
210-01-00	P	S	A	Proveedores	A
210-01-01	P	N	A	M.P.S. Mayorista	A
210-01-02	P	N	A	Vital Electrónica S.A.	A
210-01-03	P	N	A	Dimensión Digital S.A.	A
210-01-04	P	N	A	O.C.I. S.A.	A
210-01-05	P	N	A	Computel S.A.	A
210-02-00	P	S	A	Acreedores Diversos	A
210-02-01	P	N	A	Marco Aurelio Reyna Padil	A
210-02-02	P	N	A	Roberto Aguilar Esquivel	A
210-02-03	P	N	A	Telmex S.A.	A
210-03-00	P	S	A	Impuestos por pagar	A
210-03-01	P	N	A	I.V.A. por pagar	A
210-03-02	P	N	A	10% Sobre honorarios	A
210-04-00	P	S	A	Anticipos de Clientes	A
210-04-01	P	N	A	Clientes Varios	A
300-00-00	C	S	A	Capital Contable	A
300-01-00	C	N	A	Capital Social	A

No. Cta.	Tipo	Acum	Nat	Descripción	Stat.
300-02-00	C	N	A	Resultado Ej. Anterior	A
300-03-00	C	N	A	Resultados del Ejercicio	A
400-00-00	R	S	D	Resultados	A
410-00-00	R	S	A	Ingresos	A
410-01-00	R	N	A	Ventas	A
410-02-00	R	N	A	Ingresos por servicios	A
410-03-00	R	N	A	Productos Financieros	A
420-00-00	R	N	D	Desctos. y Devoluciones s	A
430-00-00	R	S	D	Compras	A
430-01-00	R	N	D	Computadoras	A
430-02-00	R	N	D	Dispositivos	A
430-03-00	R	N	D	Consumibles	A
440-00-00	R	N	A	Devoluciones y Desctos s/	A
450-00-00	R	S	D	Gastos Generales	A
450-01-00	R	N	D	Papeleria y Artículos de	A
450-02-00	R	N	D	Honorarios	A
450-03-00	R	N	D	Transportes	A
450-04-00	R	N	D	Sueldos y salarios	A
450-05-00	R	N	D	Viáticos	A
450-06-00	R	N	D	Energia eléctrica	A
450-07-00	R	N	D	Impuestos	A
450-08-00	R	N	D	Manto. Automóvil	A
450-09-00	R	N	D	Teléfono	A
450-10-00	R	N	D	Depr eq. de computo	A
450-11-00	R	N	D	Depr. Mob. y eq. de Ofna.	A
450-12-00	R	N	D	Varios	A
450-13-00	R	N	D	Arrendamiento de oficina	A
450-14-00	R	N	D	Atención a Clientes	A
450-15-00	R	N	D	Amort. Gtos. De Instalac.	A
460-00-00	R	S	D	Gastos Financieros	A
460-01-00	R	N	D	Comisiones Bancarias	A
460-02-00	R	N	D	Comisiones Cheques devuel	A

Parte del código necesario para realizar la impresión de la balanza de comprobación mediante un lenguaje de tercera generación es el siguiente :

```
Proc Closed ImprRtro()

  UseVars ContLin,IndlMov,Saldof,Saldoi,Nopag,AuxCta,DetMesr,IndRAux,RegSdot,MesRpt

  Print "\n"
  IndCont=ContLin+1
  SaltP()
  ContLin=Contlin+1
  If not IndLMov
    Then
      MoveTo "Sald_t"
      MoveTo Record RegSdot
      DetCta()
      IndCont=ContLin+1
      SaltP()
      ContLin=ContLin+1
      Print "\n"
      Saldoi=[Saldo]+[Abono]-[Cargo]
  EndIf
```

```
MoveTo "Pol_Asen"
IndPol=False
Scan For [No-Pol]=[Movtos_t->No-Pol] And [Tipo-pol]=[Movtos_t->Tipo-Pol] And
Month([fech-pol])=MesRpt
  IndPol=True
  QuitLoop
EndScan
MoveTo "Movtos_t"
Print Fill(" ",3)
Print Format ("d5",[Fech-pol])
Print " "+[Tipo-pol]+"-"
Print Format("w4, ez",[No-Pol])
If Indpol
  then
    Print Fill(" ",8)
    Print Format("w35",[Pol_Asen->Descrip])
  Else
    Print Fill(" ",43)
EndIf
If Not IndlMov
  Then
    SdoFMov()
```

```
    Print "  "+Format("w15.2, ec, s-",Saldof)
  Else
    Print Fill(" ",17)
 EndIf
 Movtos()
 Saldoi=Saldoi-[Abono]+[Cargo]
 SdoFMov()
 Print "  "+Format("w15.2, ec, s-",Saldof)
EndProc


Proc Closed Imprsbt()

 Usevars NumNiv,IndNCta,ContLin,IndNAcum,NoPag,AuxCta,DetMesr,IndRAux,RegSdot

 If NumNiv>=IndNcta
   Then
    return
 EndIf
 IndCont=ContLin+(IndNcta-NumNiv)*2
 SaltP()
 ContLin=ContLin+(IndNcta-NumNiv)*2
```

```
  For VImpSNiv From IndNcta-1 to NumNiv Step -1
    MoveTo Record IndNacum[VImpSNiv]
    Car="-"
    Linea()
    Print "\n"
    Print Fill(" ",27)+"Totales :"+Fill(" ",43)
    Movtos()
  EndFor
  Car="="
  Linea()
  MoveTo Record Regsdot

EndProc

Proc Closed Linea()

  UseVars Car
  CarLin=Fill(Car,15)
  Print "\n",Fill(" ",81)+CarLin+"  "+CarLin

EndProc
```

```
Proc Closed DetCta()

  Print Fill(" ",2)+[Num-cta1]+[Num-cta2]+[Num-cta3]+Fill("
",18)+Format("w35",[Cuentas->Desc-cta])

 Endproc

Proc Closed ImprSdo()

  UseVars IndLMov,ContLin,RegSdot,SaldIni,Saldf,NoPag,AuxCta,DetMesr,IndRAux

  MoveTo "Sald_t"
  MoveTo Record RegSdot
  If IndLmov
    Then
      IndCont=ContLin+2
      SaltP()
      ContLin=ContLin+2
      Car="-"
      Linea()
```

```
      Print "\n"
      Print Fill(" ",62)
    Else
      IndCont=ContLin
      Saltp()
      ContLin=ContLin+1
      Print "\n"
      DetCta()
  Endif
  Saldos()
  Print "  "+Format("w15.2, ec, s-",SaldIni)
  Movtos()
  Print "  "+Format("w15.2, ec, s-",Saldf)

Endproc

Proc Closed ImprSdob()

  UseVars IndCont,SaldIni,Saldf,NoPag,AuxCta,DetMesr,IndRAux,ContLin

  Saldos()
  Arteris=" "
```

```
Abonof=0
Cargof=0
If [Cuentas->Nat-cta]="A"
  Then
    If Saldf<-0.00001
      Then
        Cargof=Saldf*-1
        Arteris="*"
      Else
        Abonof=Saldf
    EndIf
  Else
    If Saldf<-0.00001
      Then
        Abonof=Saldf*-1
        Arteris="*"
      Else
        Cargof=Saldf
    EndIf
EndIf
IndCont=ContLin+1
SaltP()
```

```
  Print "\n"
  Print " "+[Cuentas->Num-cta1]+[Cuentas->Num-cta2]+[Cuentas->Num-cta3]
  Print " ",Format("W35",[Cuentas->Desc-cta])," ",[Cuentas->Nat-cta]," "
  Print Format("W15.2",Saldini)," ",Format("W15.2",[Cargo])," ",Format("W15.2",[Abono])," "
  Print Format("W15.2",Cargof)," ",Format("W15.2",Abonof)," ",Arteris
  ContLin=Contlin+1

EndProc

Proc Closed Movtos()
  Print "  "+Format("w15.2, ec, s-",[cargo])
  Print "  "+Format("w15.2, ec, s-",[Abono])
Endproc

Proc Closed EncCta()

  UseVars AuxCta
  AuxCta="A U X I L I A R - Cuenta :
"+[Cuentas->Num-cta1]+[Cuentas->Num-cta2]+[Cuentas->Num-cta3]+"  "+[Cuentas->Desc-cta]

EndProc
```

```
Proc Closed Encabdo()

  UseVars NoPag,IndRAux,AuxCta,DetmesR,ContLin

  NoPag=NoPag+1
  Print "\n",Fill(" ",5),Format("d5",Today()),Fill(" ",42)
  Print "C y M  A.S.C.  S. A.",Fill(" ",44),"Pag. ",Format("W3",NoPag)
  Print "\n"
  If IndRAux
    Then
      Marg=Int((132-len(AuxCta))/2)
      Print Fill(" ",Marg),AuxCta,Fill(" ",Marg-22),"R.F.C. CMA911022C99","\n"
    Else
      Print Fill(" ",54),"Balanza de Comprobación ","\n"
  EndIf
  Marg1=Int((132-len(DetMesR))/2)
  Print Fill(" ",Marg1),DetMesR,"\n"
  Print "   "+"Av. Santa Ursula 177 II-102 Col. Sta. Ursula Xitla   Tlalpan  "
  Print "Mexico D.F.    C.P. 14420","\n"
  Print Fill("=",132),"\n"
  If IndRAux
    Then
```

```
      Print "   "+"Cuenta"+Fill(" ",18)+"Descripción de la Cuenta","\n"
      Print "   "+"Fecha"+"   "+" Póliza Refer."+Fill(" ",10)+"C o n c e p t o"
      Print Fill(" ",13)+"Saldo Inicial    D e b e      H a b e r    Saldo Final","\n"
    Else
      Print " "+"Cuenta          Descripción          Nat",Fill(" ",19)
      Print "Movimientos del Ultimo Mes          Saldo del Mes","\n"
      Print Fill(" ",49)," Saldo Inicial D e u d o r  A c r e e d o r  D e u d o r  A c r e e d o
r","\n"
  EndIf
  Print Fill("=",132)
  ContLin=8

EndProc
Proc Closed SaltP()
  UseVars IndCont,NoPag,AuxCta,DetMesr,IndRAux,ContLin
  If IndCont>59
    Then
      Print "\f"
      Encabdo()
  EndIf
EndProc
```

```
Proc Closed ImprGTot()

  UseVars
IndCont,NoPag,NoPag,AuxCta,DetMesr,IndRAux,ContLin,GTot_d,GTot_h,Debe_sf,Haber_sf

  IndCont=ContLin+2
  Saltp()
  ContLin=ContLin+2
  Print"\n","\n"
  Print Fill(" ",40),"T O T A L E S","          "
  Print Format("W15.2",GTot_d)," ",Format("W15.2",GTot_h)," ",Format("W15.2",Debe_sf),"
",Format("W15.2",Haber_sf)

EndProc
```

El resultado de este código es el siguiente :

Balanza de Comprobación
del mes de Agosto 94

Av. Sta. ursula 177 T 2 - 102 Col. Sta. Ursula Xitla Tlalpan Mexico D.F. C.P. 14420

Cuenta	Descripción	Nat	Saldo Inicial	Movimientos del periodo Deudor	Movimientos del periodo Acreedor	Saldos del periodo Deudor	Saldos del periodo Acreedor
100-00-00	Activo	D		0.00	2200.00	2200.00	0.00	0.00
110-00-00	Activo Circulante	D		0.00	1100.00	2200.00	0.00	1100.00 *
110-01-00	Bancos	D		0.00	1100.00	2200.00	0.00	1100.00 *
110-01-01	Cta. Cheques Serfin	D		0.00	1100.00	2200.00	0.00	1100.00 *
110-02-00	Fondo Fijo caja	D		0.00	0.00	0.00	0.00	0.00
110-03-00	Clientes	D		0.00	0.00	0.00	0.00	0.00
110-03-01	Clientes de mostrador	D		0.00	0.00	0.00	0.00	0.00
110-03-02	Servicios Corporativos	D		0.00	0.00	0.00	0.00	0.00
110-03-03	Eurística Ingeniería y si	D		0.00	0.00	0.00	0.00	0.00
110-04-00	I.V.A. Acreditable	D		0.00	0.00	0.00	0.00	0.00
110-05-00	Impuestos Anticipados	D		0.00	0.00	0.00	0.00	0.00
110-05-01	I.S.R.	D		0.00	0.00	0.00	0.00	0.00
110-05-02	Otros imptos. Anticipad	D		0.00	0.00	0.00	0.00	0.00
110-06-00	Anticipos a Proveedores	D		0.00	0.00	0.00	0.00	0.00
110-07-00	Accionistas	D		0.00	0.00	0.00	0.00	0.00
110-08-00	Deudores Diversos	D		0.00	0.00	0.00	0.00	0.00
110-08-01	Notario Alfredo Ruiz	D		0.00	0.00	0.00	0.00	0.00
120-00-00	Activo Fijo	D		0.00	1100.00	0.00	1100.00	0.00
120-01-00	Eq. de Cómputo	D		0.00	0.00	0.00	0.00	0.00
120-02-00	Depr. Eq. de Cómputo	A		0.00	1100.00	0.00	1100.00	0.00 *
120-03-00	Mob. y Eq. de Ofna.	D		0.00	0.00	0.00	0.00	0.00
120-04-00	Depr. Mob. Y Equipo	A		0.00	0.00	0.00	0.00	0.00
120-05-00	Eq. de Transporte	D		0.00	0.00	0.00	0.00	0.00
120-06-00	Depr. de Eq. de Transport	A		0.00	0.00	0.00	0.00	0.00
130-00-00	Activo Diferido	D		0.00	0.00	0.00	0.00	0.00
130-01-00	Depositos en Garantía	D		0.00	0.00	0.00	0.00	0.00
130-02-00	Gastos de Instalación	D		0.00	0.00	0.00	0.00	0.00
130-03-00	Amort. Gastos de Instalac	A		0.00	0.00	0.00	0.00	0.00
200-00-00	Pasivo	A		0.00	0.00	0.00	0.00	0.00
210-00-00	Pasivo Circulante	A		0.00	0.00	0.00	0.00	0.00
210-01-00	Proveedores	A		0.00	0.00	0.00	0.00	0.00
210-01-01	M.P.S. Mayorista	A		0.00	0.00	0.00	0.00	0.00
210-01-02	Vital Electrónica S.A.	A		0.00	0.00	0.00	0.00	0.00
210-01-03	Dimensión Digital S.A.	A		0.00	0.00	0.00	0.00	0.00
210-01-04	O.C.I. S.A.	A		0.00	0.00	0.00	0.00	0.00
210-01-05	Computel S.A.	A		0.00	0.00	0.00	0.00	0.00
210-02-00	Acreedores Diversos	A		0.00	0.00	0.00	0.00	0.00
210-02-01	Marco Aurelio Reyna P	A		0.00	0.00	0.00	0.00	0.00
210-02-02	Roberto Aguilar Esquivel	A		0.00	0.00	0.00	0.00	0.00
210-02-03	Telmex S.A.	A		0.00	0.00	0.00	0.00	0.00
210-03-00	Impuestos por pagar	A		0.00	0.00	0.00	0.00	0.00
210-03-01	I.V.A. por pagar	A		0.00	0.00	0.00	0.00	0.00
210-03-02	10% Sobre honorarios	A		0.00	0.00	0.00	0.00	0.00
210-04-00	Anticipos de Clientes	A		0.00	0.00	0.00	0.00	0.00
210-04-01	Clientes Varios	A		0.00	0.00	0.00	0.00	0.00
300-00-00	Capital Contable	A		0.00	0.00	0.00	0.00	0.00
300-01-00	Capital Social	A		0.00	0.00	0.00	0.00	0.00
300-02-00	Resultado Ej. Anterior	A		0.00	0.00	0.00	0.00	0.00
300-03-00	Resultados del Ejercicio	A		0.00	0.00	0.00	0.00	0.00

Balanza de Comprobación
del mes de Agosto 94

Av. Sta. ursula 177 T 2 - 102 Col. Sta. Ursula Xitla Tlalpan Mexico D.F. C.P. 14420

Cuenta	Descripción	Nat	Saldo Inicial	Movimientos del periodo Deudor	Movimientos del periodo Acreedor	Saldos del periodo Deudor	Saldos del periodo Acreedor
400-00-00	Resultados	D		0.00	0.00	0.00	0.00
410-00-00	Ingresos	A		0.00	0.00	0.00	0.00
410-01-00	Ventas	A		0.00	0.00	0.00	0.00
410-02-00	Ingresos por servicios	A		0.00	0.00	0.00	0.00
410-03-00	Productos Financieros	A		0.00	0.00	0.00	0.00
420-00-00	Desctos. y Devoluciones	D		0.00	0.00	0.00	0.00
430-00-00	Compras	D		0.00	0.00	0.00	0.00
430-01-00	Computadoras	D		0.00	0.00	0.00	0.00
430-02-00	Dispositivos	D		0.00	0.00	0.00	0.00
430-03-00	Consumibles	D		0.00	0.00	0.00	0.00
440-00-00	Devoluciones y Desctos	A		0.00	0.00	0.00	0.00
450-00-00	Gastos Generales	D		0.00	0.00	0.00	0.00
450-01-00	Papeleria y Articulos de	D		0.00	0.00	0.00	0.00
450-02-00	Honorarios	D		0.00	0.00	0.00	0.00
450-03-00	Transportes	D		0.00	0.00	0.00	0.00
450-04-00	Sueldos y salarios	D		0.00	0.00	0.00	0.00
450-05-00	Viáticos	D		0.00	0.00	0.00	0.00
450-06-00	Energia eléctrica	D		0.00	0.00	0.00	0.00
450-07-00	Impuestos	D		0.00	0.00	0.00	0.00
450-08-00	Manto. Automóvil	D		0.00	0.00	0.00	0.00
450-09-00	Teléfono	D		0.00	0.00	0.00	0.00
450-10-00	Depr eq. de computo	D		0.00	0.00	0.00	0.00
450-11-00	Depr. Mob. y eq. de Ofna	D		0.00	0.00	0.00	0.00
450-12-00	Varios	D		0.00	0.00	0.00	0.00
450-13-00	Arrendamiento de oficina	D		0.00	0.00	0.00	0.00
450-14-00	Atención a Clientes	D		0.00	0.00	0.00	0.00
450-15-00	Amort. Gtos. De Instalac.	D		0.00	0.00	0.00	0.00
460-00-00	Gastos Financieros	D		0.00	0.00	0.00	0.00
460-01-00	Comisiones Bancarias	D		0.00	0.00	0.00	0.00
460-02-00	Comisiones Cheques de	D		0.00	0.00	0.00	0.00
	TOTALES			2200.00	2200.00	-1100.00	-1100.00

Parte del código necesario para realizar la impresión del reporte de pólizas capturadas mediante un lenguaje de tercera generación es el siguiente :

```
View "Pol_asen"
View "Pol_t"
View "Movtos_f"
View "Movtos_t"
View "Cuentas"
Empty ("Pol_t")
Empty ("Movtos_t")
Arch_pol="Pol_t"
Arch_mov="movtos_t"
Tip_rep="Reporte de Polizas Asentadas"
Det_cta=""
ReadLib "xtr_trns" Extrae

Val1=DateVal("12/01/94")
Val2=DateVal("12/31/94")

Extrae (Val1,Val2,"Movtos_f","Movtos_t","Fech-pol",NFields("Movtos_f"))
Extrae (Val1,Val2,"Pol_asen","Pol_t","Fech-pol",NFields("Movtos_f"))
Release Procs Extrae
```

```
ReadLib "LibRpol" EncDet,DetRpol,GranTot,ImprDpol,DescCta,RepDMov,ImprMov

ReadLib "LibUtlrp" Linea,Encabdo,Saltp


DetrPol(Arch_pol,Arch_mov)

Reset


CreateLib "LibRPol"


Proc  EncDet(Tip_rep,Det_cta)


  MargI=Int((132-Len(Tip_rep))/2)

  Print Fill(" ",MargI),Tip_rep,"\n"

  Print "\n",Fill("=",132),"\n"

  Print "   No Pol   Tipo        D e s c r i p c i ó n        Fecha   Referencia","\n"

  Print "          No. Cta.      D e s c r i p c i ó n          C a r g o       A b o n o
",Format("W20",Det_cta)

  Print "\n",Fill("=",132),"\n"

  ContLin=ContLin+6


EndProc
```

```
Proc DetRPol(Arch_pol,Arch_mov)

  MoveTo Arch_pol
  Home
  NoPag=0
  GTot_Ab=0
  GTot_Ca=0
  Encabdo()
  While Not Eot()
   ImprDpol()
   Tot_Ab=0
   Tot_ca=0
   RepDMov(Arch_pol,Arch_mov)
   GTot_Ab=GTot_Ab+Tot_Ab
   Gtot_Ca=GTot_Ca+Tot_ca
   MoveTo Arch_pol
   Skip 1
  EndWhile
  GranTot(Gtot_ca,GTot_Ab,"GRAN TOTAL")

EndProc
```

```
Proc GranTot(Cargo,Abono,Det_tot)

  IndCont=ContLin+4
  SaltP()
  Linea(61,2,15,"=")
  Print Fill(" ",20),Format("w20",Det_tot),Fill(" ",21),Format("w15.2, ec, s-",Cargo),"
",Format("w15.2, ec, s-",Abono),"\n"
  ContLin=ContLin+3

EndProc

Proc ImprDPol()

  IndCont=ContLin+6
  SaltP()
  Print "\n\n","    ",Format("W4.0",[No-pol]),"      ",[Tipo-Pol],"    "
  Print Format("W35",[Descrip]),"    ",Format("D10",[Fech-pol]),"    ",[Referencia],"  ","\n\n"
  ContLin=ContLin+4

EndProc
```

```
Proc DescCta()

    NCta1=[Num-cta1]
    NCta2=[Num-Cta2]
    NCta3=[Num-Cta3]
    MoveTo "Cuentas"
    MoveTo Field "Num-Cta1"
    Locate NCta1,NCta2,NCta3
    DescCta=[Desc-cta]

EndProc

Proc RepDMov(Arch_pol,Arch_Mov)

  MNopol=[No-pol]
  MTipol=[Tipo-pol]
  MFepol=[Fech-pol]
  MoveTo Arch_mov
  Scan For [No-pol]=MNopol And [tipo-pol]=MTipol And [Fech-pol]=MFepol
    DescCta()
    MoveTo Arch_Mov
    ImprMov()
```

```
    Tot_Ab=Tot_Ab+[Abono]
    Tot_ca=Tot_ca+[Cargo]
  EndScan
  GranTot(Tot_Ab,Tot_Ca,"Total Póliza")

EndProc

Proc ImprMov()

  IndCont=ContLin+1
  SaltP()
  Print "\n","          ",[Num-Cta1],[Num-Cta2],[Num-Cta3],"   ",Format("W35",DescCta),"
",Format("w15.2, ec, s-",[Cargo])
  Print "  ",Format("w15.2, ec, s-",[Abono])
  ContLin=ContLin+1

EndProc
```

El resultado de este código es el siguiente :

Reporte de Polizas Asentadas

No Pol	Tipo / No. Cta.	Descripcion / Descripcion	Fecha / Cargo	Referencia / Abono
1	D	Poliza Ap. Ej. 97	1/01/97	d 001
	1100102	Cta. Cheques Bital	730.00	0.00
	1100200	Fondo Fijo caja	500.00	0.00
	1100301	Clientes de mostrador	0.00	176.81
	1100302	CORSERFI	280.36	0.00
	1100400	I.V.A. Acreditable	3,111.63	0.00
	1200100	Eq. de CÃmputo	19,386.60	0.00
	1200200	Depr. Eq. de CÃmputo	0.00	3,958.10
	1200300	Mob. y Eq. de Ofna.	1,815.36	0.00
	1200400	Depr. Mob. Y Equipo	0.00	680.55
	1300200	Gastos de InstalaciÃn	1,649.00	0.00
	1300300	Amort. Gastos de InstalaciÃn	0.00	219.87
	2100201	Marco Aurelio Reyna Padilla	0.00	46,425.73
	3000100	Capital Social	0.00	10,000.00
	3000200	Resultado Ej. Anterior	35,778.88	0.00
	3000300	Resultados del Ejercicio	0.00	1,790.77
		Total Poliza	63,251.83	63,251.83
		GRAN TOTAL	63,251.83	

Av. Santa Ursula 177 II-102 Col. Sta. Ursula Xitla Tlalpan Mexico D.F. C.P. 14420

Reporte de Polizas Asentadas

No Pol	Tipo	Descripcion	Fecha	Referencia		
	No. Cta.	Descripcion		Cargo	Abono	
1.	D	Telefono Febrero	20/02/97	D 001		
	1100400	I.V.A. Acreditable		20.39	0.00	
	2100201	Marco Aurelio Reyna Padilla		0.00	156.35	
	4500900	Telfono		135.96	0.00	
		Total Poliza		156.35	156.35	
		GRAN TOTAL		156.35	15	

CONCLUSIONES

Algunas de las diferencias entre el desarrollo de sistemas con un lenguaje de tercera generación y un lenguaje de cuarta generación son las siguientes :

En el Capítulo 3 y 4 se realizó la fase de análisis y diseño del sistema, para estas actividades la técnica estructurada es igualmente válida para desarrollar un sistema con un lenguaje de tercera, así como con un lenguaje de cuarta generación. Sin embargo, debido a las herramientas del lenguaje de cuarta generación (lenguaje de consulta Query), se observa que hay una reducción en el acoplamiento entre los procesos, llegando en algunos casos a anular la transmisión de datos entre estos. En el caso del proceso CALCULA SALDOS DEL PERIODO participan 15 datos en el análisis para un lenguaje de cuarta generación y 17 en el análisis para uno de tercera. En el proceso ACTUALIZA SALDOS DE CUENTAS participan 10 en el análisis para un lenguaje de cuarta generación y 12 en el análisis para un lenguaje de tercera, constituyendo una reducción en favor del lenguaje de cuarta generación del 12 % y 17 % respectivamente.

Este atributo que es considerado favorable para el desarrollo de un sistema - Puesto que mientras menor sea el acoplamiento, más fácil será realizar el desarrollo y mantenimiento del sistema - es debido a que mediante un comando query podemos realizar operaciones tales como actualización de acumulados, conteos, recuperación o borrado de los registros de una tabla, pudiendose considerar todos los registros de la tabla o bien sólo un conjunto de ellos. El uso de estas operaciones elimina la necesidad de que el programador controle el proceso del archivo.

Estas operaciones en el caso de un lenguaje de tercera generación requieren más procesamiento intermedio y transmisión de resultados entre los procesos, razón por la cual el acoplamiento entre los procesos es más grande, dificultándose como resultado de ésto el desarrollo y mantenimiento del sistema.

Durante el desarrollo del sistema (Capítulo V) es donde se encuentran los mayores beneficios, durante esta fase se aplican las herramientas que constituyen el lenguaje de cuarta generación, observándose como resultado directo de la aplicación de estas herramientas una substancial reducción en el código, implicando una gran reducción en el tiempo empleado para realizar el desarrollo del sistema (ahorro aproximado de un 80% de tiempo en el módulo estudiado).

En el caso del módulo 1.3.1 este fue desarrollado con el lenguaje de cuarta generación con un solo comando Query, mientras que con el lenguaje de tercera generación fuerón necesarias 23 instrucciones. El módulo 1.5.2 fue desarrollado con un comando Query con el lenguaje de cuarta generación, con el lenguaje de tercera generación fueron nencesarias 12 instrucciones, esto constituye una reducción mayor al 80% en el código efectuado para el lenguaje de cuarta generación.

Esta reducción en la necesidad de generar código acelera el proceso de desarrollo, pues simplifica enormemente las pruebas que se realizan sobre el sistema y por lo compacto que son los comandos Query, garantiza la correcta ejecución de los procesos.

En el desarrollo de pantallas (Sección V.2) podemos observar que el generador de pantallas automatiza las funciones de despliegue

y actualización de la información, garantizando aspectos tales como la integridad referencial de la información. El programador se desocupa de estas operaciones y centra su atención en las que son propias del sistema, que exceden las funciones del desarrollador de formas; como son actualización de contadores, revisión de condiciones, cálculos elaborados, etc. El ahorro de tiempo derivado de esto es muy alto comparado contra el de un lenguaje de tercera generación, puesto que con el 4 GL sólo es necesario generar el esquema incluyendo las tablas que son requeridas para la forma, sin ninguna línea de código. En el caso estudiado el ahorro en tiempo de desarrollo de formas es superior al 80%.

Durante la producción de reportes (Sección V.3) el generador de reportes automatiza funciones como salto de hoja, impresión de encabezados y pies de página de las secciones que constituyen el reporte, presentación y formateo de la información contenida en las tablas de la base de datos en el reporte. Al programador unicamente le corresponde la responsabilidad de señalar las tablas que son necesarias para elaborar el reporte y especificar las dependencias existentes entre ellas.En el caso del reporte para generar la balanza de comprobacion, el código realizado para generar el reporte mediante un lenguaje de tercera generación excede las 300 lineas, mientras que con el lenguaje de cuarta generación se desarrollo mediante un conjunto de comandos query (20 comandos aproximadamente) y el diseño del reporte, con un ahorro en el tiempo de desarrollo superior al 80% comparado con el lenguaje de tercera generación.

Tenemos así que con los generadores de reportes y de pantallas se automatizan gran parte de las operaciones de entrada y salida de la información, operaciones cuyo desarrollo con los lenguajes de generaciones anteriores dependían totalmente del programador, consumiendo por lo tanto gran parte del tiempo de desarrollo. En conjunción a estas herramientas el lenguaje de consulta (Query) optimiza el desarrollo de procesos para realizar cálculos y actualizaciones sobre la información.

El uso de estas herramientas hace que el desarrollo de sistemas sea más dinámico dándole posibilidad al desarrollador de prestar más atención a la comprensión del sistema bajo estudio, así como elaborar modelos funcionales en un corto período de tiempo para determinar si el sistema será aceptado o rechazado por el usuario.

BIBLIOGRAFIA

Meilir Page - Jones. 1988. The practical guide to structured system design, 2a. Ed., New Jersey, N.Y.

P.E. Avison. 1985 Information Systems Development. Addison Wesley, Inglaterra

James Martin. 1977. Organización de las Bases de Datos. 1a. Ed. Prentice Hall

Linda Weiser Friedman. 1991. Comparative Programming Languages ; generalizing the programming function. Prentice Hall, New Jersey

Peter Bishop. 1991. Conceptos de Informática. Ediciones Anaya Multimedia,Madrid

Raymond A. Lorie y Jean-Jacques Daudenarde. 1991. SQL and its applications. Prentice Hall, New Jersey

Naphtali Rishe. 1992.Database design. Mc Graw Hill, New York

Benet Campderrich. 1978. Técnicas de bases de datos. 1978. Editores técnicos asociados, Barcelona

Richard Frost. 1989. Bases de Datos y Sistemas expertos. Ediciones Díaz de Santos S. A.,Madrid

Misbah Deen. 1990. Fundamentos de los sistemas de bases de datos. Gustavo Gili. Barcelona

Tsichritzis Dionysios C. 1977. Database managment systems. Academic Press, N.Y.

APENDICE A : GLOSARIO DE TERMINOS

Algoritmo : un conjunto de pasos ejecutados con un orden para conseguir un resultado.

Atributo : una propiedad o característica de un objeto.

Base de Datos : una colección de datos que son almacenados y administrados electrónicamente

Base de datos relacional : una base de datos organizada fisicámente como un conjunto de tablas. Cualquier vista lógica de los datos puede ser formada por la unión de las tablas en diferentes formas.

Clave : dato que contiene información a cerca de una entidad para identificarla.

Compilador : un programa que traduce un lenguaje de programación en un lenguaje máquina comprensible para una computadora.

Concurrencia : la capacidad de un programa de ejecutar dos o más operaciones a la vez.

Diagrama de flujo : Es una representación gráfica para la definición, análisis, o método de solución de un problema, en el cual se utilizan símbolos que representan operaciones, datos, flujo, equipo, etc.

Diccionario de datos : almacén de datos donde se registra toda la información que describe a una base de datos

Integridad de los datos : el estado de los datos que han sido protegidos de borrados accidentales o cambios no controlados

Llamada : una instrucción en un programa que referencia a una subrutina o programa independiente. Una vez que éste termina, regresa al programa que la llamó.

Lenguaje concurrente : un lenguaje que soporta la ejecución simultánea de dos o más programas.

Lenguajes de alto nivel : son lenguajes de programación que son independientes de las computadoras. Estos lenguajes son opuestos a los lenguajes máquina y ensamblador, que son considerados de bajo nivel.

Lenguaje de consulta : lenguaje generalizado que permite a un usuario obtener información de una base de datos.

Lenguaje noprocedural : un lenguaje de programación que genera la lógica del programa directamente de la descripción del usuario del problema.

Lenguaje procedural : un lenguaje de programación tal como COBOL. FORTRAN, BASIC, etc. que se basan en el uso de un orden apropiado de acciones y en el conocimiento de operaciones de procesamiento de datos y técnicas de programación.

Macro : un pequeño programa usado para automatizar una serie de procesos o comandos.

Normalizar : el agrupamiento de datos en registros para un procesamiento eficiente en un sistema de base de datos relacional.

Preprocesador : un programa que ejecuta algún procesamiento preliminar sobre la entrada, antes de que sea procesado por el programa principal.

Procedimiento : una acción particular solicitada por otro procedimiento.

Programa : una colección de instrucciones que le dicen a la computadora qué hacer. Un programa es escrito en un lenguaje de programación y es convertido en un lenguaje de máquina mediante el uso de ensambladores y compiladores.

Programación estructurada : una filosofía de programación diseñada para manejar la complejidad formalizando y estandarizando la metodología de la programación. La programación estructurada corresponde al tipo top-down.

Query : especificación a alto nivel de la información que se desea recuperar.

Recursión : La capacidad de una subrutina o programa de ejecutarse a sí misma.

Sistema de administración de base de datos : un conjunto de programas que controlan la organización, almacenamiento y recuperación , seguridad e integridad de una base de datos.

Sistema de diccionario de datos : un conjunto de programas que controlan la organización, almacenamiento y recuperación , seguridad e integridad de un diccionario de datos.

Sistema operativo : programa que controla la calendarización y ejecución de múltiples programas. Controla los recursos del

sistema de cómputo y define estándares para el resto de los programas

Transacción : unidades de trabajo que, cuando se les permite ejecutarse concurrentemente, está garantizado que producirán resultados equivalentes a los producidos mediante una ejecución serial.

Trigger : Procedimiento que es ejecutado cuando surge una condición predefinida en la base de datos.

Usuario final : el operador de una computadora o estación de trabajo o un usuario de los resultados de un sistema.

Variable global : una variable accesible a todos los módulos de un programa.