

56  
2y



Universidad Nacional Autónoma  
de México

---

---

Facultad de Ingeniería

**NOMINA DE CONSEJERIAS COMERCIALES**

**TESIS PROFESIONAL**

Que para obtener el título de

**Ingeniero en computación**

P r e s e n t a :

**FRANCISCO JAVIER HERNANDEZ HERNANDEZ**



*Director de Tesis*  
*Ing. Luis G. Cordero Borboa*

México, D.F.

1996

**TESIS CON  
FALLA DE ORIGEN**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**A MIS PADRES:**

**ENRIQUE Y AMPARO**

*POR HABERME INDICADO EL CAMINO  
A SEGUIR Y SU APOYO INCONDICIONAL  
A LO LARGO DE ESTE RECORRIDO.*

**A MI ESPOSA:**

**VERÓNICA**

*POR LA MOTIVACIÓN, EL APOYO Y EL  
TIEMPO CEDIDO PARA LOGRAR ESTA META  
POR QUE ESTE LOGRO TAMBIÉN ES SUYO.*

**A MI HIJA:**

**MARÍA FERNANDA**

*POR LO QUE SIGNIFICA PARA MI Y POR QUE  
QUIERO SER UN EJEMPLO PARA ELLA, COMO  
MIS PADRES LO SON PARA MI.*

**A MIS HERMANOS**

**ENRIQUE, AMPARO, ALEJANDRO Y MANUEL**

*POR TENER UNA FAMILIA UNIDA, QUE ME  
APOYÓ EN TODO MOMENTO Y ME PERMITIÓ  
CONCENTRARME PARA LOGRAR ESTE OBJETIVO.*

## INDICE

Objetivo	1
Introducción	2

### CAPITULO 1

## CONCEPTOS TEÓRICOS

### 1.1 Teoría de Diseño Estructurado

1.1.1 Reseña histórica	3
1.1.2 Conceptos básicos	4
1.1.3 Análisis y Diseño de procesos	9
1.1.4 Herramientas de Análisis	10
1.1.4.1 Análisis de flujo de datos	12
1.1.4.2 Diagramas entidad relación	16
1.1.4.3 Relaciones	19
1.1.4.4 Indicadores asociativos de tipo de objeto	21
1.1.4.5 Indicadores de subtipo/supertipo	23

1.1.4.6	Diccionario de datos	26
1.1.4.7	Especificaciones de proceso	28
1.1.4.8	Diagrama de transición de estados	29
1.1.4.9	El diagrama de estructuras	31
1.1.5	Diseño estructurado	32
<b>1.2 Antecedentes de los sistemas de administración de bases de datos</b>		
1.2.1	Definiciones de base de datos	34
1.2.2	Evolución de las bases de datos	35

## **CAPITULO 2**

### **PLANTEAMIENTO DE LA PROBLEMÁTICA**

<b>2.1</b>	<b>Antecedentes</b>	40
<b>2.2</b>	<b>Requerimientos de usuario</b>	43

<b>2.3 Estrategia de Solución</b>	45
<b>2.4 Análisis</b>	46
2.4.1 Diagrama de flujo de datos	46
2.4.1.1 Diagrama de contexto	46
2.4.1.2 Diagrama cero	47
2.4.1.3 Procesos primitivos	48
2.4.2 Especificaciones de proceso	65
2.4.3 Diccionario de datos	81
2.4.4 Diagrama entidad relación	85

## **CAPITULO 3**

### ***DESARROLLO DEL SISTEMA***

#### **3.1 DISEÑO**

3.1.1 Especificaciones de diseño	89
3.1.1.1 Diseño estructurado	89

3.1.1.2 Diseño centrado en la de transformación	90
3.1.1.3 Diseño centrado en la de transacción	90
3.1.2 Diagramas de estructura	90
3.1.3 Diseño Conceptual	100
3.1.4 Diseño e implementación de los diversos módulos de programación	107
<b>CONCLUSIONES</b>	116
<b>BIBLIOGRAFÍA</b>	117

## ***Objetivo***

El objetivo de este trabajo es la realización de un sistema que permita automatizar la nómina de Consejerías Comerciales de el Banco Nacional de Comercio Exterior S.N.C., y de esta manera reducir considerablemente el tiempo y el costo que requiere este proceso. Además pretende mostrar en el sistema un ejemplo de la técnica de análisis y diseño estructurado.

## ***Introducción***

La elaboración de este trabajo, parte de la necesidad de automatizar el proceso de Nómina de Consejerías Comerciales del Banco Nacional de Comercio Exterior S.N.C..

Tomando como base la premisa anterior, se propuso el análisis y diseño de el sistema que nos permitiera la automatización.

Dentro del planteamiento inicial del problema se observó que era necesario un sistema que incluyera algunas interfaces con otros sistemas o áreas, las cuales están contempladas en este trabajo.

El tema se compone de tres capítulos, en ellos se muestra la teoría que soporta el desarrollo del sistema y las etapas del mismo.

El capítulo uno, presenta la teoría del análisis y diseño estructurado y algunos conceptos sobre sistemas.

En el capítulo dos, se muestran los requerimientos de usuario, así como el análisis y las herramientas de análisis estructurado.

El capítulo tres, presenta el diseño y desarrollo en pseudocódigo del sistema.

Finalmente, se incluyen las conclusiones y la bibliografía.

Finalmente se incluyen las conclusiones y la bibliografía.

## CAPÍTULO I

### CONCEPTOS TEÓRICOS

#### 1.1 TEORÍA DE DISEÑO ESTRUCTURADO

##### *1.1.1 Reseña Histórica*

Hasta fines de los años 70, el analista escribía lo que entendía de los requerimientos del usuario en un enorme documento que consistía primeramente en una narrativa. Los primeros autores de textos de "análisis estructurado", sobre todo (De Marco 1978), (Gane y Sarson 1977) y (Weinberg, 1978), señalaron que estos pesados tomos (a menudo conocidos como "especificación funcional") se veían afectados por diversos problemas importantes:

- Eran monolíticos
- Eran redundantes
- Eran ambiguos
- Eran imposibles de mantener

Mientras que se debatían todos estos problemas, ya se estaba adoptando un conjunto complementario de ideas en el área de programación y diseño. Estas ideas, normalmente conocidas como diseño y programación estructurados, prometían grandes mejoras en la organización, codificación, prueba y mantenimiento de los programas de computadora. Y, de hecho, si han demostrado ser útiles, aunque cada vez más organizaciones de procesamiento de datos han empezado gradualmente a darse cuenta de que no tenía caso

escribir programas brillantes y diseñar sistemas altamente modulares si nadie sabía realmente que era lo que se suponía que el sistema debería hacer. En realidad, se podría argumentar que el diseño y la programación estructurados les permitían a algunos equipos llegar a un desastre más rápidamente que antes, al construir una brillante solución al problema equivocado.

Como resultado, ha habido un movimiento gradual (puesto que aceptarlo le ha llevado a la profesión de desarrollo de sistemas al rededor de diez años) tendiente a hacer especificaciones funcionales que sean :

- gráficas
- participadas
- mínimamente redundantes

Este enfoque, al que por lo general se conoce como análisis estructurado, se utiliza ahora en la mayoría de las organizaciones de desarrollo de sistemas orientados a los negocios, al igual que en gran número de las orientadas hacia la ingeniería. Se pueden encontrar aun algunas organizaciones que produzcan especificaciones tipo novela victoriana, pero son minoría y, como los dinosaurios se extinguirán.

### ***1.1.2 Conceptos básicos***

Para el desarrollo de un sistema se deben identificar dos áreas principales, el área del problema y el área de solución. Dentro de el área de el problema participan directamente el

usuario y el analista. El usuario participa exponiendo la situación actual , así como los problemas que presenta. El analista se encarga de revisar, asesorar y proponer una solución al usuario.

Dentro de el área de solución participan los diseñadores quienes se encargan de realizar un diseño de acuerdo a los requerimientos hechos por el usuario y captados por el analista.

Es importante que un sistema tenga un buen diseño ya que de este dependen que el sistema se fácil de :

- ENTENDER
- MODIFICAR
- MANTENER

Existen dos tipos de diseño, el diseño externo y el diseño interno.

### ***Diseño Externo***

podemos definir como diseño externo todo aquello que el sistema presenta a el usuario, por ejemplo pantallas y reportes. Debido a esto, para un usuario la calidad de un sistema es medida por el diseño externo.

### ***Diseño Interno***

El diseño interno son todos aquellos procesos que el sistema realiza internamente y que pasan desapercibidos para el usuario.

Los objetivos de el diseño interno son los siguientes.

- lograr una máxima independencia entre un modulo y otro
- mejorar la calidad de los sistemas
- simplificar el problema, evitando la complejidad
- lograr que algún cambio en el sistema tenga el impacto deseado en el mismo.
- lograr que cada modulo sea funcional
- que se logre un entendimiento completo sobre el diseño

### **Métodos de arriba hacia abajo A-B**

El método A-B es un método en el que se sigue un proceso analítico en donde existen varias metodologías A-B

- .diseño orientado a funciones
- .diseño orientado a datos
- .diseño orientado a objetos

Un diseño A-B permite contar con un gran soporte de documentación, incluyendo diagramas estructurados y documentación de programas. Además cuenta con soporte electrónico como son las herramientas de diseño de sistemas.

El diseño A-B presenta las siguientes características:

- . Se logran desarrollos fáciles de verificar
- . Al diseñar estructuradamente un sistema, este es factible de modificar
- . En caso de alguna modificación se identifica fácilmente el impacto que puede tener
- . Los procesos complejos son mas fáciles de entender
- . en un diseño A-B se logra una independencia entre módulos
- . Cada módulo tiene una función específica

### **Conceptos sobre sistemas**

Un sistema es un conjunto de componentes interrelacionados dentro de una frontera definida; Un sistema interactúa con el medio ambiente a través de la frontera del sistema. Se dice que las relaciones entre los elementos de un sistema, no es otra cosa que la estructura del sistema; además podemos analizar a cada elemento del sistema como otro sistema.

Podemos entender un sistema de dos formas diferentes, la primera como si fuera una “caja negra” y la segunda llamada de la “caja blanca”. La “caja negra” también es llamada “vista desde el exterior” y la blanca es llamada “vista desde el interior”.

## CAJA NEGRA

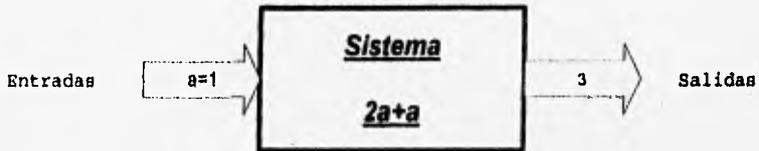


En la caja negra el sistema es definido funcionalmente es decir ilustra que se hace y no como se hace. Indica la o las entradas, una transformación y la o las salidas.

La transformación puede ser descrita por diagrama de flujo de datos, pares Ordenados ,una ecuación o una definición Funcional.

En la caja blanca el sistema se define en base a su comportamiento es decir, describe los procesos necesarios para transformar las entradas y obtener la salida deseada .

## CAJA BLANCA



### *1.1.3 Análisis y Diseño de procesos*

Para llevar un buen desarrollo estructurado es necesario llevar a cabo los siguientes pasos:

- Identificación de necesidades del usuario
- Elaboración del diseño funcional
- Diseño estructural
- Diseño procedural
- Programación

### ***1.1.4 Herramientas de análisis***

Gran parte de la labor que desempeñara como analista involucra el modelado del sistema que desea el usuario. Hay muchos tipos diferentes de modelos que podemos elaborar, así como hay muchos modelos diferentes que puede hacer de una casa nueva un arquitecto.

El termino “modelo” pudiera parecer algo formal y atemorizante, pero representa un concepto que usted ha manejado durante la mayor parte de su vida. Considere los siguientes tipos de modelos:

- Mapas: modelos bidimensionales de nuestro mundo en que vivimos.
- Globos terráqueos: modelos tridimensionales de nuestro mundo.
- Diagramas de flujo: Representaciones esquemáticas de un edificio, o de un puente, etcétera.
- Partituras musicales: representaciones gráficas y textuales de notas musicales y tiempos de una pieza musical.

Aunque uno no sepa leer un modelo arquitectónico, el concepto de dicho modelo no debería asustarle; no es demasiado difícil imaginarse que pudiera aprender a leer y entender tales modelos, aun si jamas piensa crear uno usted mismo. Los usuarios con los que trabaja podrán ciertamente leer los modelos (con una pequeña ayuda inicial) y pudieran incluso ser capaces de crearlos.

¿Porque construimos modelos? ¿Por que no se construye simplemente el sistema mismo? La respuesta es que podemos construir modelos de manera tal que enfatizamos ciertas propiedades críticas del sistema, mientras que simultáneamente desacentuamos otros de sus aspectos. Esto nos permite comunicarnos con el usuario de una manera enfocada, sin distraernos con asuntos y características ajenas al sistema. Y si nos damos cuenta de que nuestra comprensión de los requerimientos del usuario no fue la correcta (o de que el usuario cambio de parecer acerca de sus requerimientos ), podemos hacer cambios en el modelo o desecharlo y hacer uno nuevo, de ser necesario. La alternativa es tener algunas reuniones preliminares con el usuario y luego construir todo el sistema; desde luego, existe el riesgo de que el producto no sea aceptable, y pudiera ser excepcionalmente costoso hacer un cambio a esas alturas.

Por esta razón, el analista hace uso de herramientas de modelado para:

- Concentrarse en las propiedades importantes del sistema y al mismo tiempo restar atención a otras menos importantes.
- Discutir cambios y correcciones de los requerimientos del usuario, a bajo costo y con el riesgo mínimo.
- Verificar que el analista comprenda correctamente el ambiente del usuario y que lo haga respaldado con información documental para que los diseñadores de sistemas y los programadores puedan construir el sistema.

No todas las herramientas de modelado cumplen con estos propósitos: por ejemplo, una descripción narrativa de 500 páginas de los requerimientos del usuario (que es, grosso modo, un modelo) podría 1) Contribuir a oscurecer todas las propiedades del sistema, 2) Costar más en su elaboración que el sistema mismo y 3) No verificar si el analista comprendió o no las necesidades reales del usuario.

Ahora presentaremos y discutiremos brevemente dos herramientas de modelado de sistemas importantes: el diagrama de flujo de datos y el diagrama de entidad relación.

El diagrama de flujo de datos ilustra las funciones que el sistema debe realizar; los diagramas de entidad relación hacen énfasis en las relaciones entre los datos.

Aunque el diagrama de flujo de datos proporciona una visión global bastante conveniente de los componentes funcionales del sistema no da detalles de estos. Para mostrar detalles acerca de que información se transforma y de como se transforma, se ocupan dos herramientas textuales de modelado adicionales: El diccionario de datos y las especificaciones de procesos.

#### ***1.1.4.1 Análisis de flujo de datos***

Se cuenta con una herramienta muy importante para la documentación del análisis estructurado, esta herramienta es el diagrama de flujo de datos.

Un viejo adagio de la profesión de desarrollo de sistemas dice que un sistema de proceso de datos involucra tanto los datos como el proceso, y no se puede construir un sistema exitoso sin considerar ambos componentes. El aspecto de proceso de un sistema ciertamente es algo importante de modelar y de verificar con el usuario. El modelado que

llevamos a cabo puede describirse en una variedad de maneras:

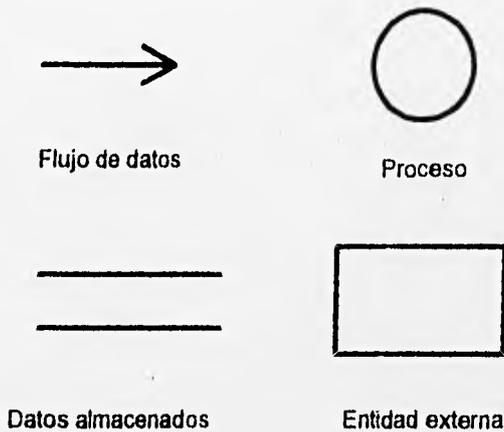
- ¿Con que funciones debe desempeñar el sistema? ¿Cuales son las interacciones entre dichas funciones?
- ¿Que transformaciones debe llevar a cabo el sistema? ¿Que entradas se transforman en que salidas?
- ¿Que tipo de labor debe realizar el sistema ? ¿De donde obtiene la información para llevar a cabo dicha labor? ¿Donde entrega los resultados de su labor?

La herramienta de modelado que utilizamos para describir la transformación de entradas a salidas es un diagrama de flujo de datos

Los diagramas de flujo de datos consisten en procesos, agregados de datos, flujos y terminadores:

- Los procesos se representan por medio de círculos, o “burbujas”, en el diagrama . Representan las diversas funciones individuales que el sistema lleva a cabo. Las funciones transforman entradas en salidas.
- Los flujos se muestran por medio de flechas. Son las conexiones entre los procesos (funciones del sistema) y representan la información que dichos procesos requieren como entrada o la información que generan como salida .
- Los agregados de datos se representan por medio de dos líneas paralelas o mediante una elipse. Muestran colecciones (o agregados) de datos que el sistema debe almacenar por un periodo de tiempo. Cuando los diseñadores de sistemas y los programadores terminan de construir el sistema, los agregados existirán como archivos o bases de datos.

- Los terminadores muestran las entidades externas con las que el sistema se comunica. Típicamente se trata de individuos o grupos de personas (por ejemplo, otro departamento o división dentro de la organización), sistemas de cómputo externos y organizaciones externas.



Aunque el diagrama de flujo de datos proporciona una visión global bastante conveniente de los componentes funcionales del sistema, no da detalles de estos. Para mostrar detalles acerca de que información se transforma y como se transforma, se ocupan dos herramientas textuales de modelado adicionales: El diccionario de datos y la especificación de procesos.

Una parte muy importante dentro del diseño, es el diseño detallado; nos permite describir un proceso paso a paso en diferentes etapas, es decir desde el lenguaje natural hasta código de programa.

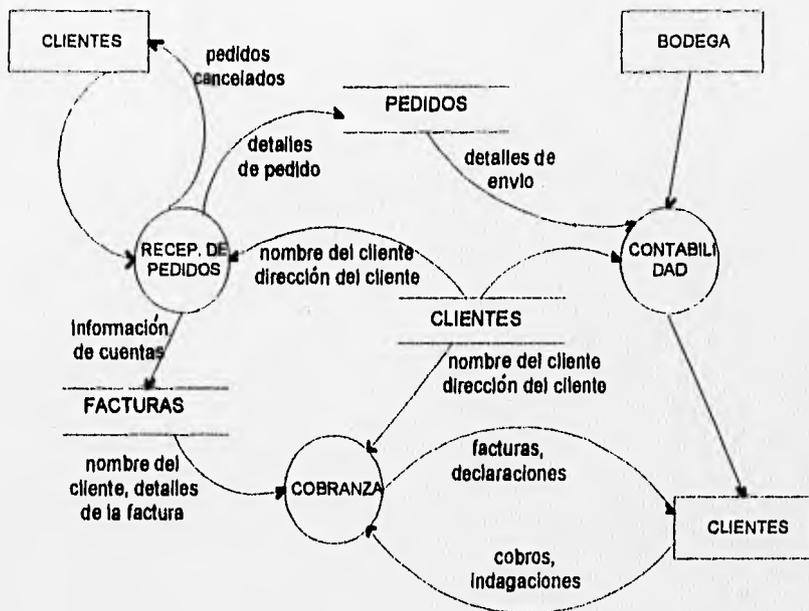


Diagrama de flujo de datos

#### **1.1.4.2 Diagramas entidad-relación**

El diagrama entidad-relación (También conocido como DER, o diagrama E-R) es un modelo de datos que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema. Es muy diferente del DFD, que modela las funciones que lleva a cabo un sistema; y es diferente del diagrama de transición de estados, que modela el comportamiento dependiente del tiempo de un sistema.

A menudo el grupo de administración de datos (grupo AD) es responsable de administrar y controlar la información esencial de un negocio; cuando se comience a construir un nuevo sistema de información se requerirá hablar con estas personas para poder coordinar la información del sistema con el modelo global, corporativo, de información.

El diagrama de entidad-relación es una herramienta útil para llevar a cabo esta conversación.

A menudo existe otro grupo dentro de la administración con un nombre similar: el grupo de administración de bases de datos (a veces conocido como grupo de ABD). Se suele localizar dentro del departamento de proceso de datos (mientras que el de administración de datos no necesariamente está ahí), y su labor es asegurar que las bases de datos computarizadas se organicen, administren y controlen de manera eficiente. Así que suele ser el equipo de implantación responsable de tomar el modelo esencial (independiente de alguna tecnología específica) y deducirlo a un modelo de base de datos eficaz para algún sistema de base de datos. El diagrama entidad relación es una herramienta efectiva de modelado para comunicarse con el grupo de administración de base de datos. Basándose en la información presentada por el DER, el grupo de administración de bases de datos puede ver el tipo de claves o índices o apuntadores que se necesitaran para llegar de manera eficiente a los registros de bases de datos

Para el analista , el DER representa un gran beneficio también: enfatiza las relaciones entre almacenes de datos en el DFD que de otra forma se hubieran visto solo en la especificación del proceso.

Considere un caso extremo: ¿Que tal si no están realizando funciones ? ¿ Que tal si el propósito del sistema que se esta construyendo no es hacer algo, sino meramente ser el recipiente de una gran cantidad de información interesante ? Tal sistema pudiera llamarse sistema de consultas adhoc, o sistema de apoyo a decisiones. En tal tipo de sistema, pudiéramos concentrarnos por completo en el modelo de datos, y ni siquiera preocuparnos por construir el modelo DFD, que esta orientado a las funciones. Desde luego, esto es claramente una situación rara: la mayoría de los sistemas si efectúan funciones; a menudo encontramos que construir primeramente el modelo de datos hace mas fácil descubrir cuales son las funciones requeridas

### ***Componentes de un DER***

Hay cuatro componentes principales en un diagrama de entidad relación

- Tipos de objetos
- Relaciones
- Indicadores asociativos de tipo objeto
- Indicadores de supedito/subtipo

### ***Tipos de objetos***

El tipo de objeto se representa en un diagrama de entidad-relación por medio de una caja rectangular. Representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros individuales (o instancias) tienen las siguientes características:

- Cada una puede identificarse de manera única por algún medio. Existe alguna forma de diferenciar entre instancias individuales del tipo de objeto. Por ejemplo, si se tiene un tipo de objeto conocido como **cliente**, debemos ser capaces de distinguir uno de otro (tal vez por un número de cuenta, por su apellido o por un número de seguro social). Si todos los clientes son iguales (si hay un negocio en el que solo entes sin cara y sin nombre que entran a la tienda a comprar cosas), entonces **cliente** no sería un tipo de objeto con significado.



- Cada uno juega un papel necesario en el sistema que se construye. Es decir, para que el tipo de objeto sea legítimo, debe poder decirse que el sistema no puede operar sin tener acceso a esos miembros.

- Cada uno puede describirse por uno o mas datos. Es decir, un **cliente** puede describirse por medio de datos tales como nombre, domicilio, límite de crédito y número telefónico. Muchos textos sobre bases de datos describen esto como "asignar datos a un tipo de objeto".

El objeto es algo material del mundo real, y tipo de objeto es su representación en el sistema. Sin embargo un objeto también pudiera ser algo no material: por ejemplo, horarios, planes, estándares, estrategias y mapas.

#### **1.1.4.3 Relaciones**

Los objetos se conectan entre si mediante relaciones. Una relación representa un conjunto de conexiones entre objetos, y se representa por medio de un rombo. A continuación se presenta una relación sencilla que pudiera existir entre dos o mas objetos.



Es importante reconocer que la relación representa un conjunto de conexiones. Cada instancia de la relación representa una asociación entre cero o mas ocurrencias de un objeto

y cero o mas ocurrencias del otro. Así en la figura anterior la relación etiquetada como compras puede contener las siguientes instancias individuales:

- Instancia 1: el cliente 1 compra el artículo 1
- Instancia 2: el cliente 2 compra los artículos 2 y 3
- Instancia 3: el cliente 3 compra el artículo 4
- Instancia 4: el cliente 4 compra los artículos 5,6 y 7
- Instancia 5: el cliente 5 no compra ningún artículo
- Instancia 6: los clientes 6 y 7 compran el artículo 8
- Instancia 7: los clientes 8,9 y 10 compran los artículos 9,10 y 11
- etc.

Como puede verse, entonces, una relación puede conectar dos o mas instancias del mismo objeto.

Nótese que la relación representa algo que debe ser recordado por el sistema: algo que no pudo haberse calculado ni derivado mecánicamente. Así, el modelo de datos de la figura anterior indica que existe alguna razón relacionada con el usuario para recordar el hecho de que el cliente 1 compra el artículo 1, etc. Y la relación también indica que no existe nada a priori que hubiera permitido determinar que el cliente 1 compro el artículo 1 y nada mas. La relación representa la memoria del sistema.

#### ***1.1.4.4 Indicadores asociativos de tipo de objeto***

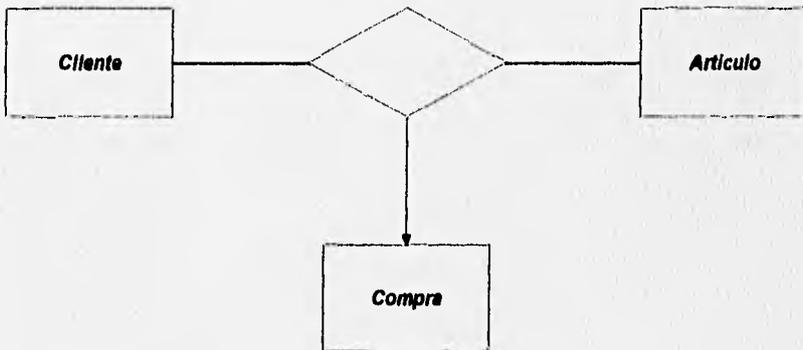
Una notación especial en el DER es el indicador asociativo de tipo de objeto; representa algo que funciona como objeto y como relación. Otra manera de ver esto es considerar que el tipo asociativo de objeto representa una relación acerca de la cual se desea mantener alguna información.

Considere por ejemplo el caso sencillo de un cliente que adquiere un artículo (o artículos) como se muestra en la siguiente figura. Sin tener en cuenta si se incluye o no la notación de tipo procedimiento, el punto principal es que la relación de compra no hace más que asociar un cliente con uno o más artículos.





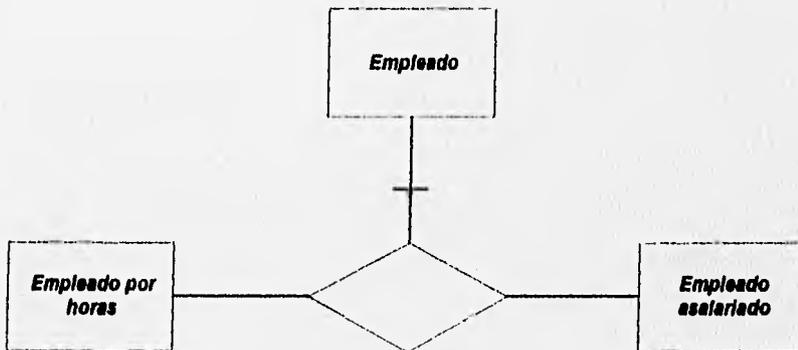
Pero suponga que existen datos que deseamos recordar a cerca de cada instancia de una compra (por ejemplo, a que hora del día se hizo). ¿Donde se podría almacenar dicha información? 'Hora del día' definitivamente no es un atributo de cliente, ni de articulo. Mas bien, se asocia 'hora del día' con la compra misma, y esto se muestra en el siguiente diagrama.



Nótese que compra ahora se escribe dentro de una caja rectangular conectada, por medio de líneas dirigidas, a un rombo de relación sin nombre. Esto pretende indicar que compra funciona como un tipo de objeto y una relación.

#### 1.1.4.5 Indicadores de subtipo/supertipo

Los tipos de objeto de subtipo/supertipo consisten en tipos de objeto de una o mas subcategorías, conectados por una relación. En la siguiente figura se muestra un subtipo/supertipo típico: la categoría general es empleado y las subcategorías son empleado asalariado y empleado por horas. Nótese que los subtipos se conectan al supertipo por medio de una relación sin nombre; note también que el supertipo se conecta a la relación con una línea que contiene una barra.



En esta notación el supertipo se describe por datos que se aplican a todos los subtipos. Por ejemplo, en la figura anterior se podría imaginar que todos los empleados se describen por hechos tales como:

- Nombre
- Años de servicio
- Domicilio particular
- Nombre del supervisor

Sin embargo, cada subtipo se describe por medio de datos diferentes; de otro modo, no tendría caso hacer distinción entre ellos. Por ejemplo, se podría imaginar que un empleado asalariado se describe por cosas tales como:

- Salario mensual
- Porcentaje anual adicional
- Aportación para coche de la empresa

el empleado por horas por medio de cosas como:

- Paga por hora
- Cantidad por tiempo extra
- Hora de comienzo

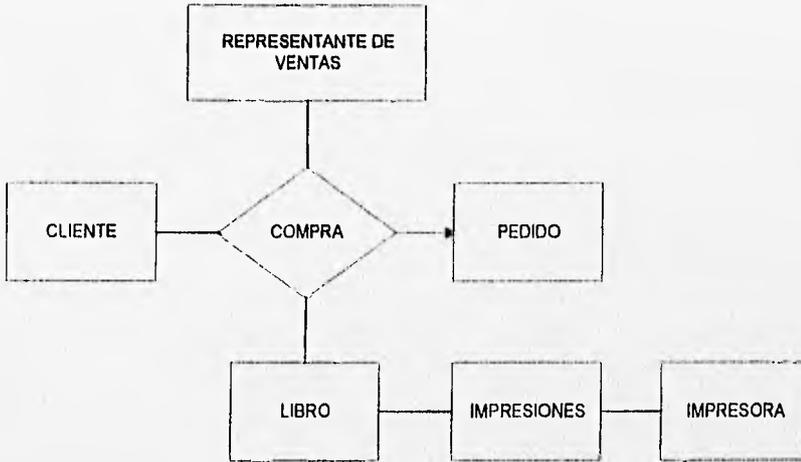


Diagrama de entidad-relacion típico

Cada una de las cajas rectangulares corresponde a un almacén de datos en un DFD, y puede verse que hay relaciones (conexiones) que normalmente no se aprecian en un DFD. Esto se debe a que el DFD enfoca la atención del lector a las funciones que el sistema efectúa no a los datos que ocupa.

#### **1.1.4.6 El diccionario de datos**

Otra herramienta de modelado importante que discutiremos es el diccionario de datos; aunque no tiene la presencia y el atractivo gráfico de los DFD, los diagramas de entidad-relación y los diagramas de transición de estados, es crucial. Sin los diccionarios de datos, el modelo de los requerimientos del usuario no puede considerarse completo; todo lo que se tendría es un borrador rudimentario.

La importancia del diccionario de datos a menudo se pasa de largo, sin el diccionario de datos en el análisis de sistemas el usuario no podrá estar seguro de que entendió los detalles de la aplicación.

El diccionario de datos es un listado organizado de todos los datos pertinente al sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento de todas las entradas, salidas, componentes de almacenes y cálculos intermedios. El diccionario de datos contiene lo siguiente:

- El significado de los flujos y almacenes que se muestran en los DFD.
- La composición de agregados de paquetes de datos que se mueve a la largo de los flujos, es decir, paquetes complejos (por ejemplo el domicilio de un cliente), que pueden descomponerse en unidades mas elementales (como ciudad, estado y código postal).
- La composición de los paquetes de datos en los almacenes.
- Los valores y unidades relevantes de piezas elementales de información en los flujos de datos y los almacenes de datos.
- Los detalles de las relaciones entre almacenes que se enfatizan en un diagrama de entidad-relación.

En la mayoría de los sistemas reales con los que se trabajan, los paquetes o elementos de datos serán lo suficientemente complejos como para que se necesite describirlos en términos de otras cosas. Los elementos complejos de datos se definen en términos de elementos mas sencillos, y los sencillos en términos de los valores y unidades legítimos que puedan asumir.

### **Notación del diccionario de datos.**

Existen muchos esquemas de notación comunes utilizados por el analistas de sistemas. El que se muestra a continuación es de los mas comunes y utiliza varios símbolos sencillos:

= esta compuesto

+ y

( ) optativo (puede estar presente o ausente)

{ } iteración

[ ] seleccionar una de varias alternativas

\*\* comentario

@ identificador (campo clave) para un almacén.

| separa opciones alternativas en la construcción

Por ejemplo se puede definir el nombre así:

Nombre = título de cortesía + nombre + (segundo nombre) + apellido

Título de cortesía = [ Sr. | Srta. | Sra. | Dr. | Profesor ]

Nombre = { carácter legal }

Segundo nombre = { carácter legal }

Apellido = { carácter legal }

Carácter legal = [ A-Z|a-z|0-9|'|\_| ]

#### ***1.1.4.7 Especificaciones de proceso***

Como se menciono anteriormente, la mayoría de los analistas usan el lenguaje estructurado como método favorito para escribir especificaciones de proceso. Tal vez sea mas importante señalar que la mayoría de los analistas de las organizaciones utilizan una herramienta para escribir todas sus especificaciones. Esto es, en mi opinión, un gran error; usted debe sentirse libre para utilizar una combinación de herramientas de especificación según a) las preferencias de usuario, b) sus propias preferencias y, c) la naturaleza propia de los propios procesos.

A pesar de que la mayoría de los analistas están a favor del lenguaje estructurado, debe recordar que se puede usar cualquier método mientras satisfaga dos requerimientos cruciales:

- La especificación del proceso debe expresarse de una manera que puedan verificar tanto el usuario como el analista.
- El proceso debe especificarse de una forma que pueda ser comunicada efectivamente al público involucrado.

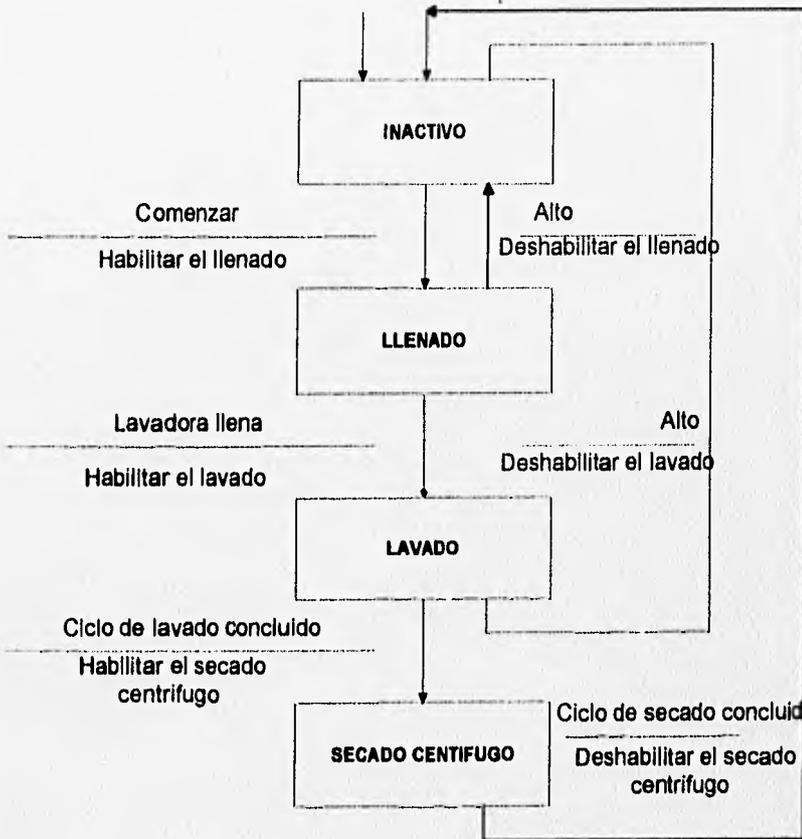
Una buena herramienta de especificación de proceso debe también tener una tercera característica: no debe imponer (o implicar) decisiones de diseño e implantación arbitrarias. A menudo esto es muy difícil, pues el usuario, de quien depende la función que realizará cada burbuja en el DFD, suele escribirla en los términos en los que la lleva a cabo en la actualidad. Su trabajo como analista consiste en destilar de esto la esencia de lo que dicha función es y no como se lleva a cabo hoy en día.

#### ***1.1.4.8 Diagrama de transición de estados.***

El comportamiento del programa depende del tiempo, es decir, la secuencia con la cual se hará el acceso a los datos y se ejecutarán las funciones, es un aspecto de muchos sistemas complejos. Para algunos sistemas computacionales de empresas este aspecto no es importante, puesto que la secuencia es esencialmente trivial. Así, en muchos sistemas computacionales ( Aquellos que ni son de tiempo real, ni están en línea ), La función N no puede realizar su labor hasta que recibe la entrada que requiere; y esta entrada se produce como salida de una función N-1, y así sucesivamente.

Sin embargo, en muchos sistemas de línea y tiempo real tanto en el campo de los negocios, en el de la ciencia y la ingeniería tienen complejas relaciones en el tiempo que deben modelarse tan cuidadosamente como las funciones y las relaciones entre datos. Por ejemplo, muchos sistemas de tiempo real deben responder dentro de un margen breve, posiblemente de tan solo unos microsegundos, a ciertas entradas provenientes del ambiente exterior. y deben estar preparados para recibir diversas combinaciones y secuencias de entradas a las cuales se debe responder adecuadamente.

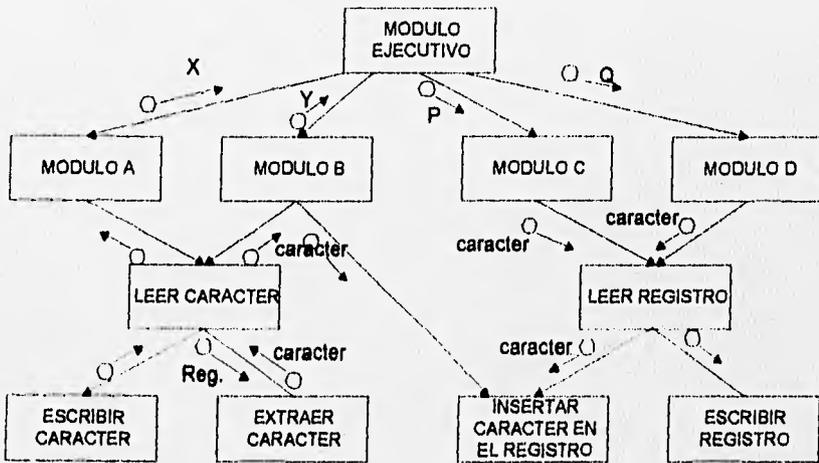
La herramienta de modelado que utilizamos para describir este aspecto del comportamiento de un sistema es el diagrama de transición de estados.



*Un diagrama de transición de estados de una lavadora automática*

**1.1.4.9 El diagrama de estructuras**

Los diseñadores de sistemas suelen utilizar los diagramas de flujo de datos, diccionarios de datos, especificaciones de procesos y diagrama de entidad-relación y de transición de estados creados por el analista para crear una arquitectura de programas, es decir, una jerarquía de módulos (los que a veces se conocen como subrutinas o procedimientos) para realizar los requerimientos del sistema. Una herramienta gráfica de modelado comúnmente utilizada para representar tal jerarquía de programas es el diagrama de estructuras.



### **1.1.5 Diseño estructurado**

#### **Objetivos del diseño**

Además de lograr los objetivos que se especifican en el modelo de implantación del usuario, el diseñador también se ocupa de la calidad global del diseño. La capacidad que los programadores exhiban para implantar un sistema de alta calidad y libre de errores depende en gran medida de la naturaleza del diseño; de manera similar, la capacidad de los programadores de mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación depende de la calidad del diseño.

El campo del diseño estructurado ofrece guías para ayuda al diseñador a determinar los módulos, y sus interconexiones, que mejor realizarán los requerimientos especificados por el analista. Las dos reglas más importantes son las referentes al acoplamiento y la cohesión; a continuación se discuten estas y algunas otras reglas comunes.

- **Cohesión.** Grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que conforman un módulo) son necesarios y suficientes para llevar a cabo una sola función bien definida. Esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos, y también debe asegurarse de no juntar procesos no relacionados.
- **Acoplamiento.** Grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación o cambio y modificación de algún módulo o módulos. En la práctica,

esto significa que cada modulo debe tener interfaces sencillas y limpias con otros, y que se debe compartir un numero mínimo de datos entre módulos. También significa que un módulo dado no debe modificar la lógica interna o los datos de algún otro módulo.

- **Tamaño del módulo.** De ser posible, cada módulo debe ser lo suficientemente pequeño como para caber en una sola pagina (o para que pueda desplegarse en una sola pantalla). Desde luego, a veces no es posible determinar que tan grande va a ser un módulo hasta haberlo escrito, pero las actividades iniciales de diseño a menudo darán al diseñador una buena pista de que el módulo va a ser grande y complejo. Si es así, debe partirse en uno o mas niveles de submodulos.
- **Alcance del control.** El numero de subordinados inmediatos que un módulo administrador puede llamar se cono como el alcance del control. Un módulo no debe llamar a mas de una media docena de módulos de nivel inferior.
- **Alcance del efecto/Alcance del control.** Esta regla sugiere que cualquier módulo afectado por el resultado de alguna decisión debe ser subordinado (aunque no necesariamente un subordinado inmediato) del módulo que toma la decisión .

Existe otra razón para tener interés en el diseño de sistemas: tal vez le toque hacerlo. Sobre todo en los sistemas pequeños y medianos, a menudo se espera que el mismo individuo documente los requerimientos del usuario y además desarrolle el diseño. Por ello, se puede esperar que decida sobre la mejor manera de asociar el modelo de los requerimientos del usuario en diferentes configuraciones de procesadores; tal vez tenga que decidir como implantar de la mejor manera el modelo lógico de datos y tal vez tenga que decidir como asignar a las funciones del sistema las distintas tareas dentro de cada procesador.

## 1.2 ANTECEDENTES DE LOS SISTEMAS DE ADMINISTRACIÓN DE BASES DE DATOS.

### 1.2.1 Definiciones de Base de Datos.

En forma simplificada, una base de datos es un conjunto de archivos que contengan datos.

En forma práctica este término hace referencia a muchos otros conceptos; una base de datos presupone o implica la existencia de un programa que pueda manipular los datos contenidos en el archivo o los archivos existentes; de esta manera, se debe hablar de un Sistema de Base de Datos, porque se manejan programas y archivos formando un sistema para lograr un objetivo coherente. De una manera más formal: "Una base de datos (o un Sistema de Base de Datos) es una colección de información usualmente organizada y ordenada para ser utilizada para ser utilizada en una aplicación o con algún objetivo en particular" . La estructura básica de una base de datos relacional, entendida como un sistema de archivos de información, es la siguiente: el componente básico son los archivos, que en terminología de bases son llamados tablas. Una tabla es una colección de información organizada en renglones (registros) y columnas (campos). Para que la recopilación de información se haga en forma coherente, los datos que se capturan y que pertenecen a una sola entidad se guardan en un renglón de la tabla con una relación lógica llamada eneada. Es decir, una eneada toma al conjunto de renglones en una o varias tablas como una serie de elementos dato relacionados y heterogéneos (una eneada es capaz de reconocer un conjunto de renglones de varias tablas debido a que establece vínculos lógicos entre los elementos).

Un Sistema de Base de Datos se basa, fundamentalmente, en dos cosas: una serie de archivos que contienen los datos y un programa que sepa como acceder esos datos. El programa en cuestión debe saber como leer los datos, cómo escribirlos, la forma en que los

va a presentar al usuario, etc. Un programa con estas características es llamado Administrador, Manejador o Gestor de Bases de datos; en forma más general, un Sistema Manejador de Bases de Datos (en adelante llamado DBMS, Data Base Management System) es la combinación de programas y tablas que se utilizan conjuntamente para el manejo de datos. Este es el concepto que engloba en realidad el término 'base de datos'. Para que un DBMS pueda ser considerado como administrador conveniente para el almacenamiento de grandes cantidades de datos, debe mostrar las siguientes características:

- Rápido acceso para la recuperación de información.
- Actualización conveniente.
- Economía de almacenamiento.

La capacidad para representar estructuras de información del mundo real, la confiabilidad, la protección de la privacidad y el mantenimiento de integridad son criterios importantes que se deben evaluar en la organización de cualquier sistema de archivos.

### ***1.2.2 Evolución de las bases de datos.***

Los antecesores de los manejadores de bases de datos eran simples rutinas generales de archivos. Estos sistemas de manejo de archivos eran capaces de desempeñar discretamente algunas operaciones comunes como la ordenación, mantenimiento de archivos y generación de reportes.

Durante las décadas de los 50's y los 60' se empezaron a desarrollar algunos productos de definición de datos; estos productos permitían definir la estructura de base de datos que se podían acceder por múltiples usuarios.

En 1969, se propuso una definición para un Lenguaje de Descripción de Datos (DDL - Data Description Language) y un Lenguaje de manipulación de Datos (DML - Data Manipulation Language) para bases de datos, esto sirvió como base de los sistemas de redes para administración de bases de datos.

Los primeros sistemas para la administración de bases de datos se basaban en la teoría de relaciones jerárquicas y de redes entre los datos.

El modelo de red permite que un tipo de registro tenga más de una relación. Un modelo menos general plantea un tipo de relaciones jerárquicas entre registros, estructuradas en árbol; este tipo de relaciones es la base para el modelo jerárquico. El modelo de datos jerárquico permite que un tipo de registro tenga solamente una relación como "padre" y solamente una relación como "hijo".

Los modelos de datos jerárquico y de redes harán principalmente "havegacionales": un usuario empieza desde un registro padre o propietario y navega a través de los miembros de una relación a través de las construcciones Set next, Set first o set last (construcciones ir al próximo, ir al principio o ir al final). Además, las relaciones propietario/miembro (para el modelo de red) o la relación padre/hijo (para el modelo jerárquico) se almacenaban en forma explícita en los registros de la base de datos. En forma más específica, las implantaciones de DBMS jerárquicas y de redes no tenían una independencia física de los datos. Esto significaba que el punto de vista del usuario de las bases de datos reflejaban la manera en que estaban organizados, almacenados y la forma en que se accedían los datos según fuera

el medio físico de almacenamiento. En algunos casos, se requería que el usuario o el administrador del sistema de manejo de las bases de datos (Data Base management system Administrator, DBA) especificara los detalles para el emplazamiento de registros, áreas de almacenamiento, ordenación de registros, localizaciones de registro, etc. Además del problema que implicaba tener que dar especificaciones técnicas (y asegurarse que las especificaciones dadas eran correctas), este enfoque limitaba seriamente la extensibilidad, la facilidad de mantenimiento, la reusabilidad y la portabilidad de las aplicaciones desarrolladas a partir de los sistemas de manejo de bases de datos basados en estos modelos.

A principio de la década de los 70's, Ted Codd propuso el modelo relacional de datos para proporcionar más flexibilidad en la organización de bases de datos grandes, y para paliar algunos problemas, de los modelos anteriores. El modelo relacional para DBMS prevalece sobre el uso de los otros dos modelos principales debido a varias ventajas inherentes. La primera de esas ventajas es que los lenguajes relacionales de consultas como el SQL (Structured Query Lenguaje - Lenguaje Estructurado para Consultas) son mucho más declarativos que los lenguajes navegacionales de los primeros modelos; esto significa que el usuario especifica lo que se quiere de una base de datos en un estilo declarativo de alto nivel de programación diciendo qué es lo que se va a acceder de la base de datos, y no cómo accederlo. El modelo relacional es simple y elegante.

Las bases de datos relacionales siguen expandiendo su popularidad y su uso. Aún así, han habido varias propuestas alternativas sobre modelaje de base de datos; muchos de estos modelos post-relacionales de datos fueron desarrollados como prototipos en laboratorios de investigación y nunca fueron comercializados.

Una de las primeras propuestas alternativas para el modelaje de base de datos fue el modelo semántico. La motivación para el modelo semántico (y para la mayoría de los modelos de datos) es muy similar al objetivo de la orientación a objetos: modelar el mundo real tan cercanamente como sea posible. El enfoque de relaciones entre entidades es una técnica ampliamente aceptada para el manejo de datos. Una entidad es un objeto o cosa que existe y que puede distinguirse de otras entidades; una entidad puede ser una persona, una institución, un vuelo, etc. Las entidades se describen a través de atributos o propiedades.

En términos de construcciones regulares de bases de datos, las entidades conforman registros mientras sus atributos se representan mediante campos de esos registros.

Los modelos semánticos se usan principalmente como herramientas de diseño de bases de datos; cuando está listo este esquema semántico, se mapea en un esquema relacional usando el lenguaje de definición de datos de alguna base de datos relacional (tal como el DDL de SQL). El usuario puede entonces recuperar y actualizar los datos almacenados por medio del esquema usando un lenguaje relacional de manipulación de datos (tal como el DML de SQL). Pero esto es inconveniente y no natural; han habido intentos de incorporar capacidades de manipulación de datos en el modelo semántico, a través de las relaciones funcionales.

El enfoque semántico para modelaje de datos no fue el único que trato de agregar más semántica a los modelos tradicionales; hubo mas modelos de datos que han tratado de extender incrementalmente el modelo relacional para permitir una mayor flexibilidad, mientras se conserva una sólida fundamentación.

El espacio de los objetos en el modelo relacional consiste de una colección de tablas planas. Cada tabla es un conjunto de filas o renglones (tuplas); los valores en columnas de

cada fila (atributos de las tuplas) solamente pueden ser instancias de tipos atómicos de datos tales como enteros, reales o cadenas de caracteres. La representación de tablas planas es conocida como la primera forma normal. Otro modelo post-relacional es el llamado modelo de objetos complejos. Los modelos de objetos complejos intentan relajar las restricciones de la primera forma normal; los modelos de objetos complejos son conocidos también modelos, de no-primer forma normal, modelos de relaciones anidadas, modelo de conjunto y tuplas o modelos de datos basados en la lógica con términos complejos.

Con un modelo relacional anidado, el usuario podrá tener atributos valuados en la relación (esto es, basado en un conjunto de tuplas) y, por tanto, podrá representar, almacenar o recuperar directamente un conjunto de tuplas. Se pueden hacer modelos más generales para objetos complejos (incluyendo el anidamiento de objetos arbitrarios) constituyendo espacios de objetos encima de una colección tipos atómicos básicos, usando dos constructores de objetos; conjuntos y tuplas.

Los modelos de objetos complejos post-relacionales mencionados hasta ahora, permiten solamente espacios de objetos estructurados en árbol; aunque esto está bien fundamentado conceptualmente, estos modelos no permiten que un objeto sea un sub-objeto de padres múltiples, tampoco soportan espacios de objetos estructurados en grafos. Para que se presenten estas características los modelos deben soportar el concepto orientado a objetos de identidad de objetos.

## CAPÍTULO 2

# PLANTEAMIENTO DE LA PROBLEMÁTICA

### 2.1 Antecedentes

Actualmente existen empresas mexicanas que cuentan con oficinas en el extranjero ya sean representaciones comerciales, o representaciones financieras. Uno de los procesos administrativos que representa mayor dedicación y en consecuencia tiempo es el cálculo de la nómina de dichas oficinas, el envío de fondos y los registros que se originan de este proceso (aplicaciones contables, registro de movimientos en bancos, concertaciones de divisas y transferencia de fondos en forma electrónica vía swift) son muy especiales, esto se debe en primer lugar al manejo de una divisa diferente en cada oficina así como el manejo de sueldos en moneda extranjera tomando en cuenta los movimientos generados de el proceso de nómina en moneda nacional.

En las Representaciones Comerciales Existen dos tipos de personal .

El personal de base, es personal generalmente mexicano contratado en las oficinas ubicadas en México.

El personal Local, es personal generalmente de el país en donde se encuentra la oficina y son contratados por honorarios.

Los sueldos del personal local son sueldos en la moneda local. El personal de base tiene dos sueldos, un sueldo en pesos mexicanos y un sueldo en moneda local. El sueldo en pesos mexicanos es el que se les asigna en el momento que son contratados en México, y en caso de que regresaran a trabajar a las oficinas en México ese sería su sueldo. El sueldo en moneda local de el personal de base es un sueldo que se les asigna en moneda extranjera considerando el costo de vida de el país en el que se encuentran ubicadas las oficinas.

Es importante mencionar que para una sola oficina pueden existir cuatro diferentes monedas:

Moneda Nacional

Moneda Local

Moneda de Concertación

Moneda de envío

Todo esto es lo que hace complejo el proceso de nómina de representaciones comerciales y que evidentemente requiere de un Sistema de computo para realizarlo en mucho menos tiempo y sin errores.

A continuación se mencionan las principales actividades que se realizan en una nómina de representaciones comerciales.

Consulta tipos de cambio

Concertación de Compra de Divisas

Registro en Control de Cuentas Bancarias

Envío de Fondos Via SWIFT

Contabilización en diferentes tipos de moneda

Aparentemente el generar una nómina en moneda extranjera sería sencillo, si tomamos los sueldos en moneda extranjera y los convertimos a pesos según el tipo de cambio del día, pero esto únicamente aplica para los empleados locales o de honorarios ya que para los empleados de base se deben de tomar en cuenta sus movimientos generados por la nómina en moneda nacional.

Debido a los registros que se originan de el proceso de nómina de representaciones comerciales, es necesario establecer interfaces con los siguientes sistemas.

Sistema de Control de Divisas

Sistema de Bancos

Sistema de Contabilidad Integral

Sistema de Nómina en moneda nacional

Interface con el SWIFT.

## 2.2 Requerimientos de usuario

Normalmente en el proceso de nómina de consejerías comerciales se llevan a cabo las siguientes actividades principales.

- Obtención de movimientos generados en el proceso de nómina en moneda nacional
- Obtención de tipos de cambio
- Cálculo de el ajuste por costo de vida
- Generación de recibos de pago con netos en moneda extranjera y su equivalente en pesos
- Generación de telex para consejerías
- Elaborar concertación de divisas
- Elaborar concertación en bancos
- Generación de órdenes de pago
- Contabilización de la nómina

### Movimientos en moneda nacional

Existe un proceso previo al de la nómina de consejerías comerciales, este proceso es llamado nómina en moneda nacional y realiza el cálculo de la nómina de todos los empleados del banco, sin importar si son empleados de consejerías comerciales, oficinas regionales u oficina matriz, pero sin considerar a los empleados de honorarios.

### Cálculo de el ajuste por costo de vida

Uno de los aspectos más importantes dentro del proceso de nómina de consejerías comerciales es el cálculo del concepto denominado "Ajuste por costo de vida".

Todo empleado de consejerías comerciales tiene un sueldo en moneda extranjera y un sueldo en moneda nacional, el sueldo en moneda extranjera debe ser recibido completamente cada quincena sin descontar impuesto alguno, pero considerando las deducciones generadas en el proceso de nómina en moneda nacional como son prestamos, pensiones etc.

Ejemplo:

	Dólares	Pesos
Sueldo en moneda extranjera	2,000	
Tipo de cambio	5.0	
Equivalente en pesos sueldo en moneda extranjera		10,000
Sueldo en moneda nacional		6,000
Impuesto descontado por el sueldo en moneda nacional		1,800

La cantidad que debe cobrar cada quincena el empleado de la consejería comercial es 10,000 pesos, pero la nómina en moneda nacional únicamente le genera un movimiento por el importe registrado en su sueldo en moneda nacional, en este ejemplo son 6,000 pesos. La diferencia es de 4,000 pesos, pero también se le descontaron 1,800 pesos de impuesto

correspondientes al sueldo en moneda nacional, los cuales es necesario regresárselos, es decir, el ajuste por costo de vida es igual a la diferencia de sueldos (considerando el tipo de cambio del día) + la devolución del impuesto descontado, para que finalmente nos de un monto total de 10,000 pesos

### ***2.3 Estrategia de solución***

La solución al requerimiento de automatizar la nómina de consejerías comerciales, es el desarrollo de un sistema dentro de el equipo AS400, incluyendo interfaces con otros sistemas como son el de tipos de cambio, de bancos y contable.

Para la elaboración del sistema se utilizarán técnicas de análisis y diseño estructurado.

Dicha técnica cuenta con una serie de herramientas, de las cuales, para la realización de este trabajo, se emplean la siguientes

**Diagramas de flujo de datos**

**Diagramas de entidad relación**

**Diccionario de datos**

**Miniespecificaciones**

**Diagrama de estructura**

## 2.4 Análisis

El análisis estructurado es una técnica de desarrollo de sistemas, la cual permite examinar en forma detallada los componentes de un sistema. En el se presentan las relaciones y flujos de información entre los procesos de un sistema.

Se emplea el análisis estructurado como herramienta de apoyo en el estudio general de el sistema, especificando las interrelaciones de información entre los procesos, así como los datos de entrada y salida.

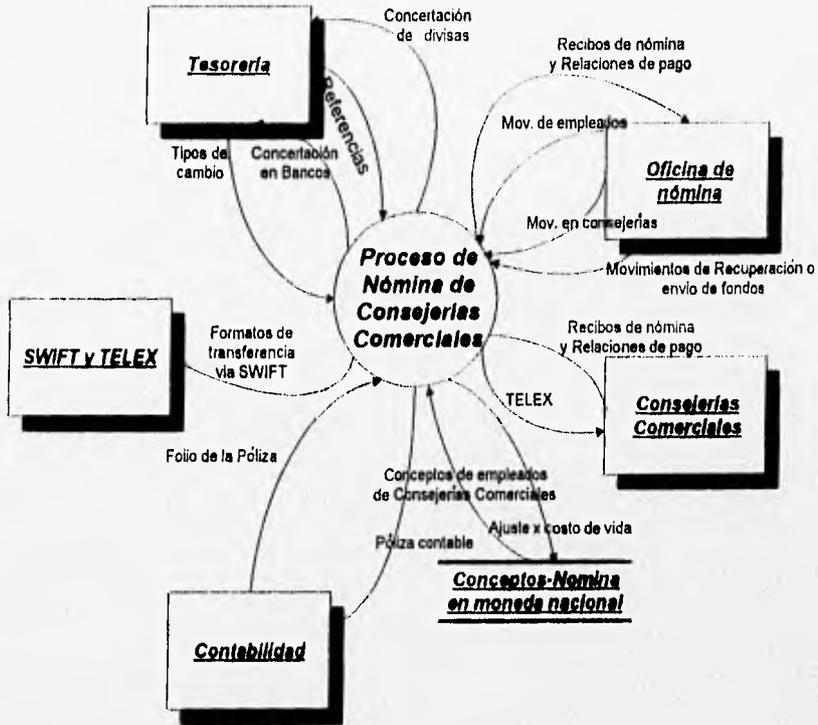
### 2.4.1 Diagramas de flujo de datos

Una de las herramientas más importantes de la etapa de análisis son los diagramas de flujo de datos, los cuales se presentan adelante.

#### 2.4.1.1 Diagrama de contexto

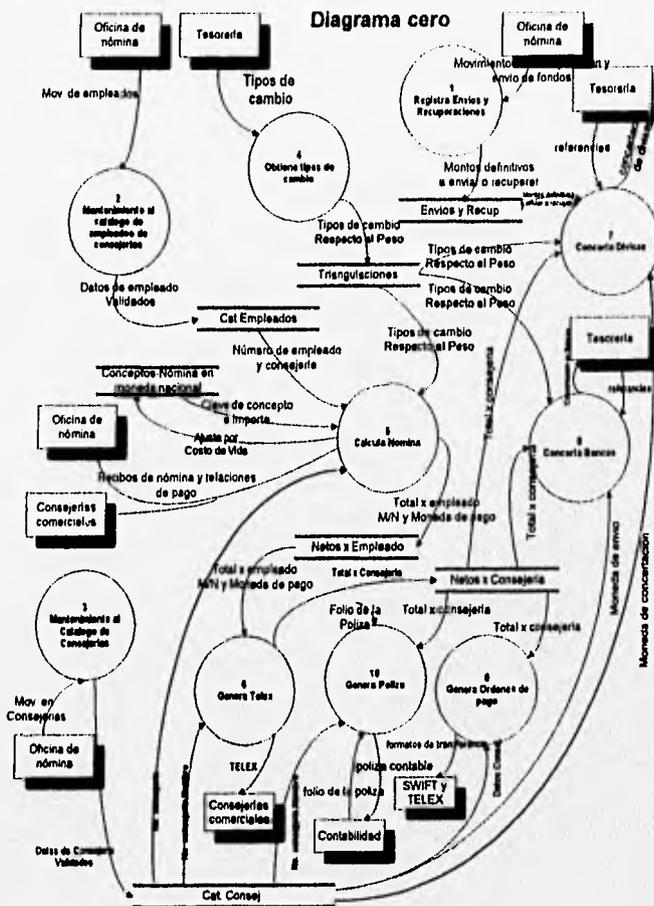
De acuerdo con la técnica de análisis estructurado, a continuación se presenta el diagrama de contexto de la nómina de consejerías comerciales, en el se delimita el alcance del estudio efectuado.

Asimismo, se muestran los nexos de información que existen con cada una de las áreas involucradas en la nómina de consejerías comerciales.



### 2.4.1.2 Diagrama cero

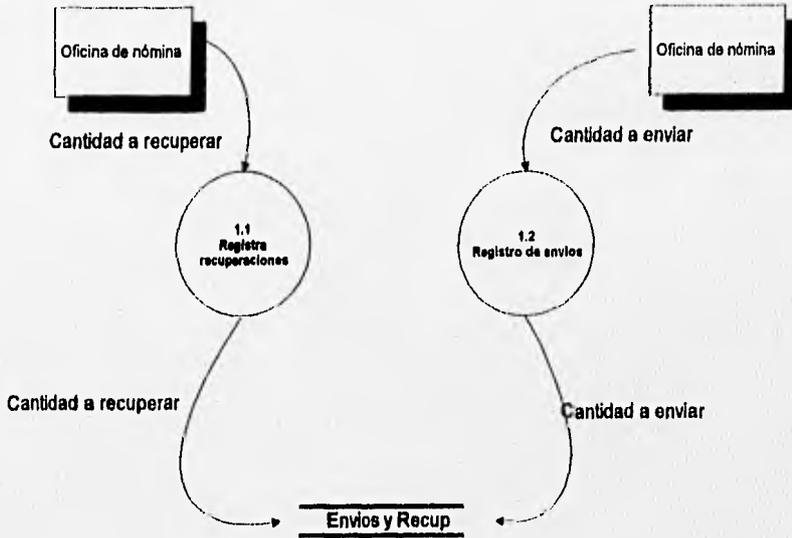
El diagrama cero corresponde al primer nivel de elementos que conforman el sistema, entendiéndose como "elementos", los procesos y flujos de información que lo constituyen. Es importante hacer notar, que este diagrama representa la primera partición del diagrama de contexto.



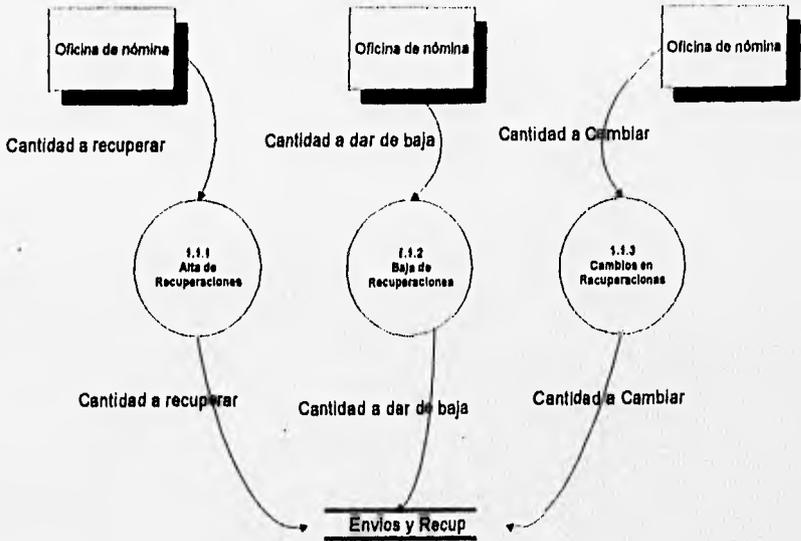
### 2.4.1.3 Procesos primitivos

En seguida aparecen los procesos resultantes de la partición de cada uno de los procesos, hasta llegar a un nivel en el que ya no sea posible dividirlos más, por lo que desde ese momento son considerados primitivos.

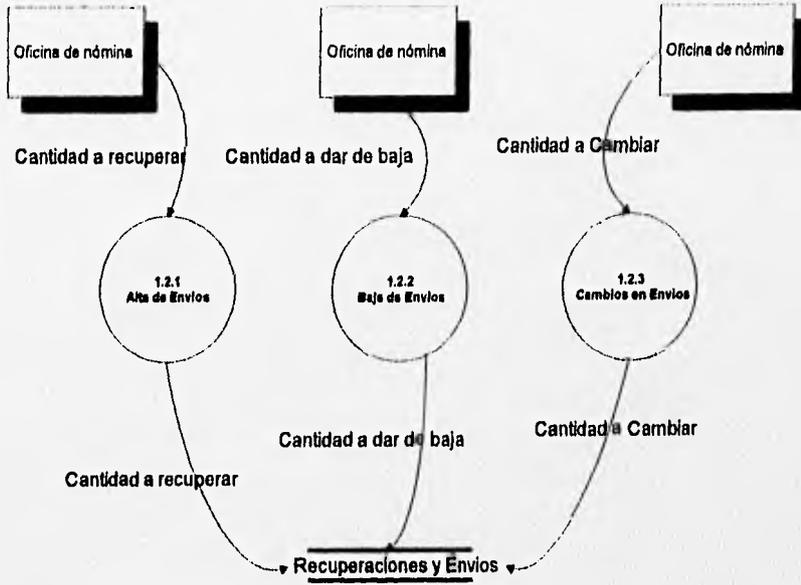
**Diagrama 1**  
**Registra envíos y recuperaciones**



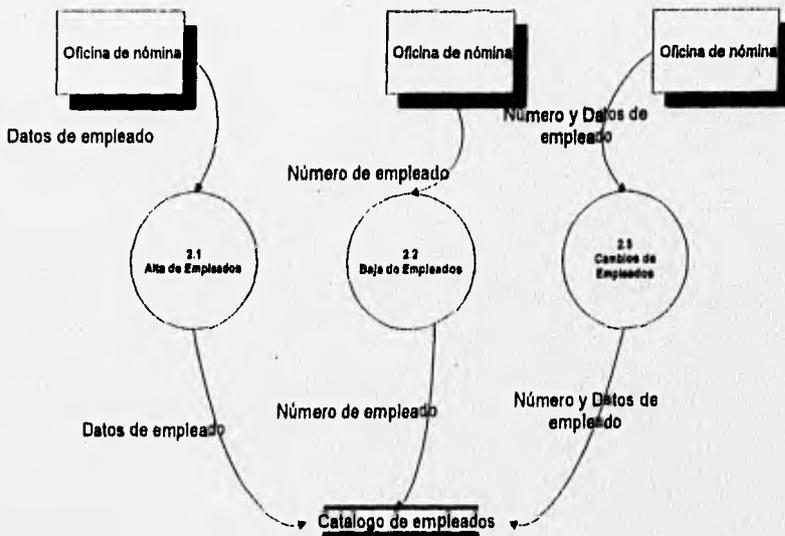
**Diagrama 1.1**  
**Registra**  
**recuperaciones**



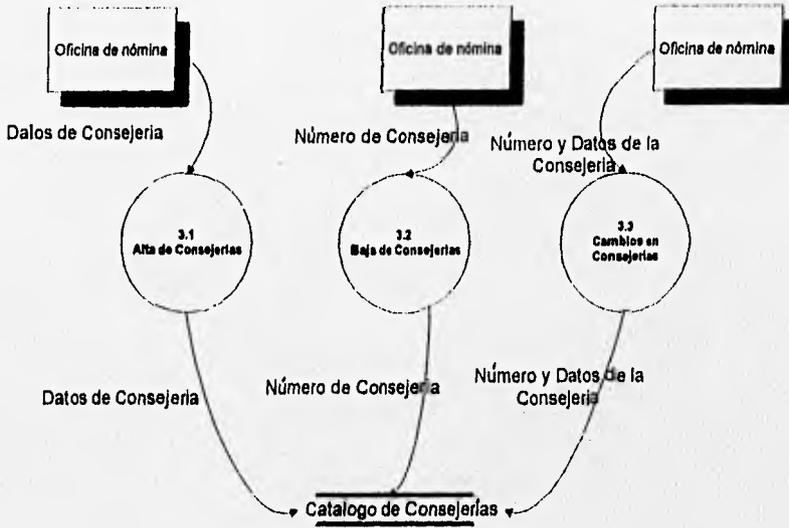
**Diagrama 1.2**  
**Registro de envíos**



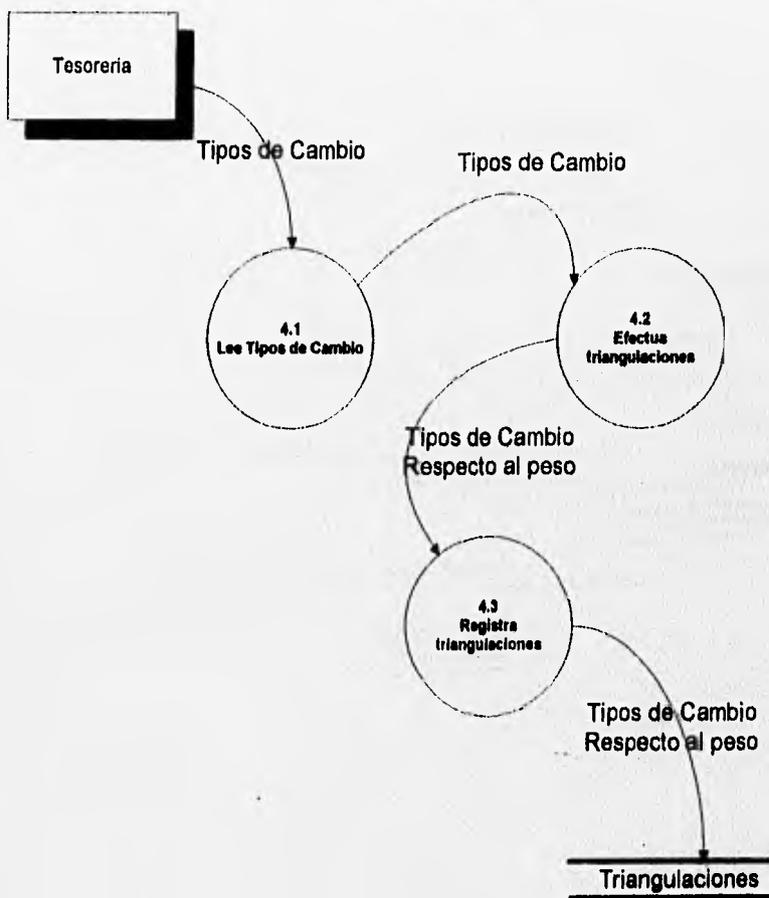
**Diagrama 2**  
**Mantenimiento al catálogo de empleados**  
**de consejerías**



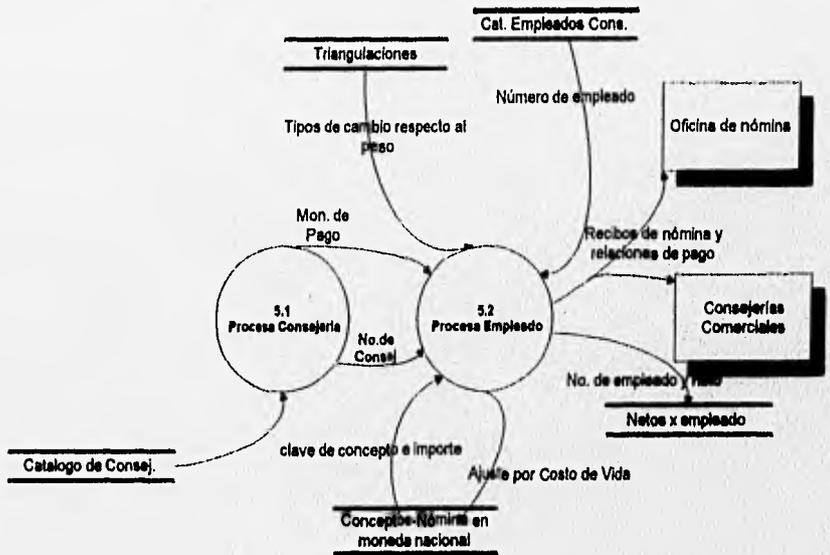
**Diagrama 3**  
**Mantenimiento al catalogo**  
**de consejerías**



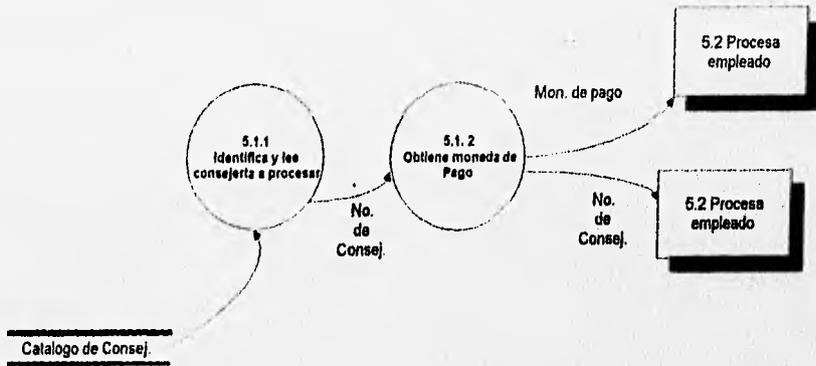
**Diagrama 4**  
**Obtiene tipos de cambio**



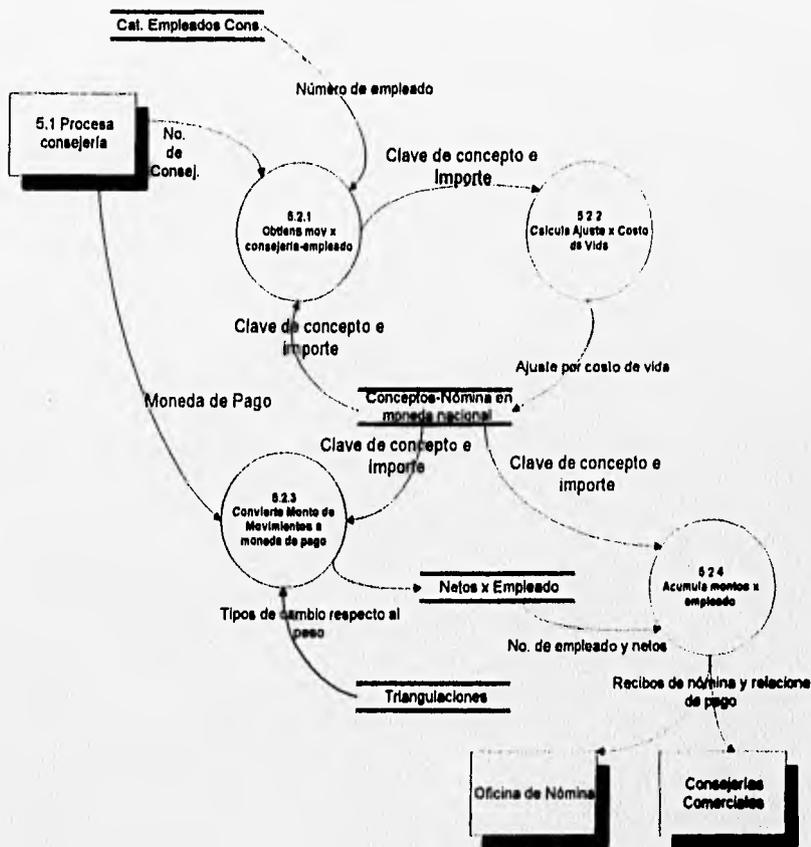
**Diagrama 5**  
**Cálculo nómina**



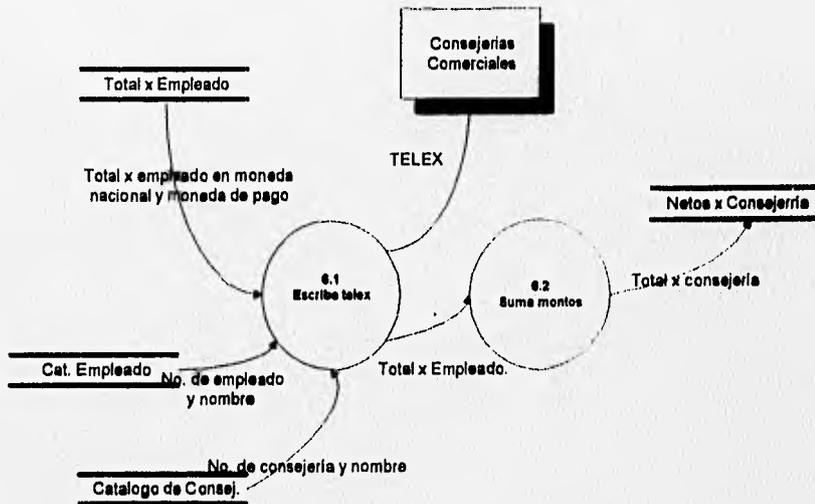
**Diagrama 5.1**  
**Procesa consejería**



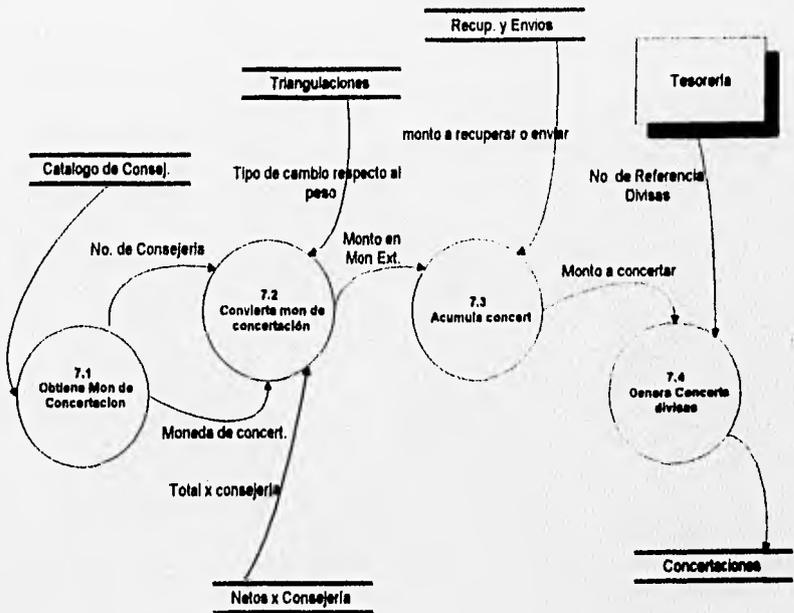
**Diagrama 5.2**  
**Proceso empleado**



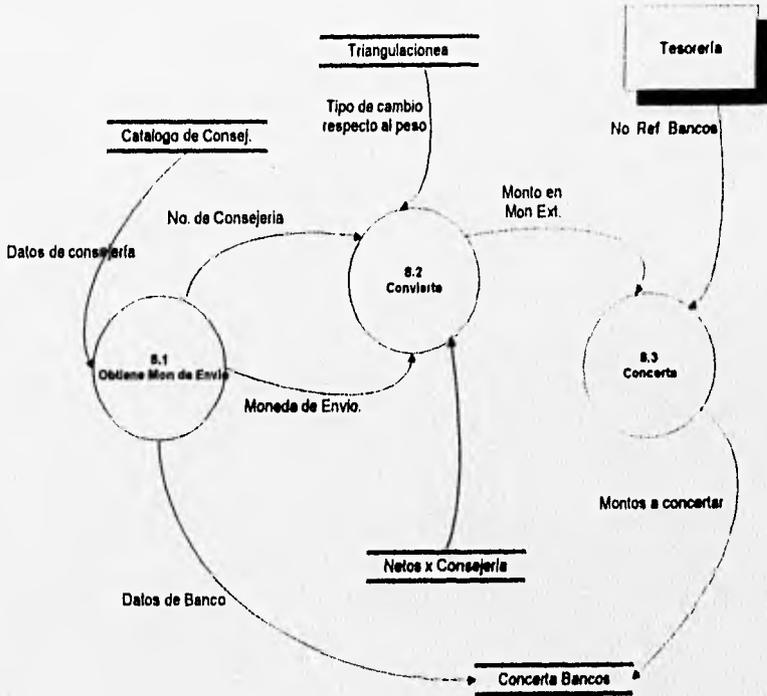
**Diagrama 6**  
**Genera TELEX**



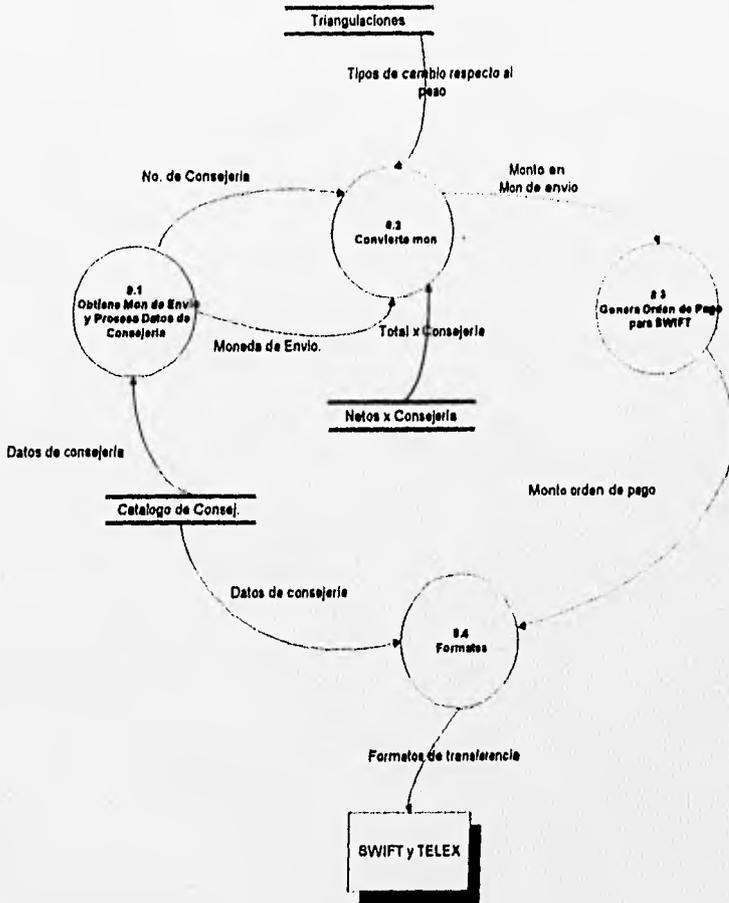
**Diagrama 7**  
**Concerta Divisas**



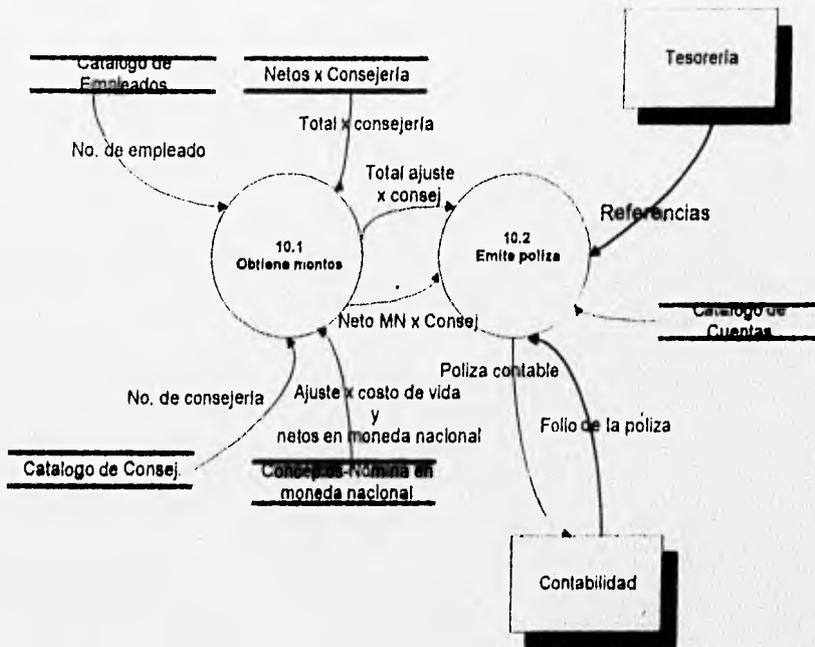
**Diagrama 8**  
**Concerta Bancos**



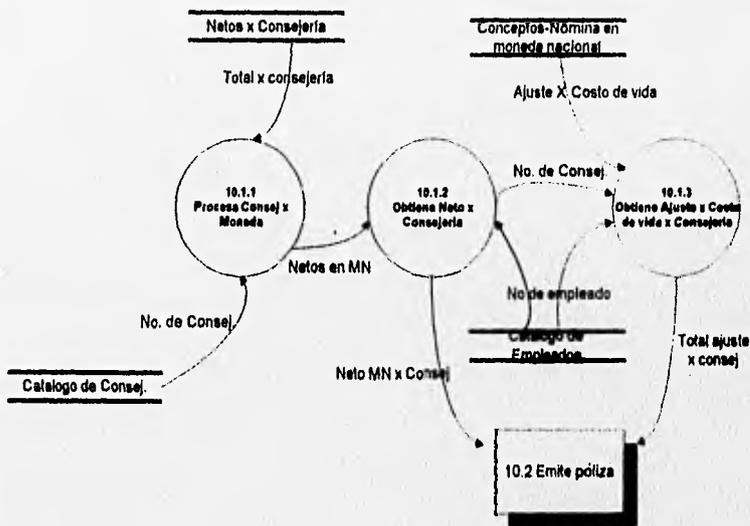
**Diagrama 9**  
**Genera ordenes de pago**



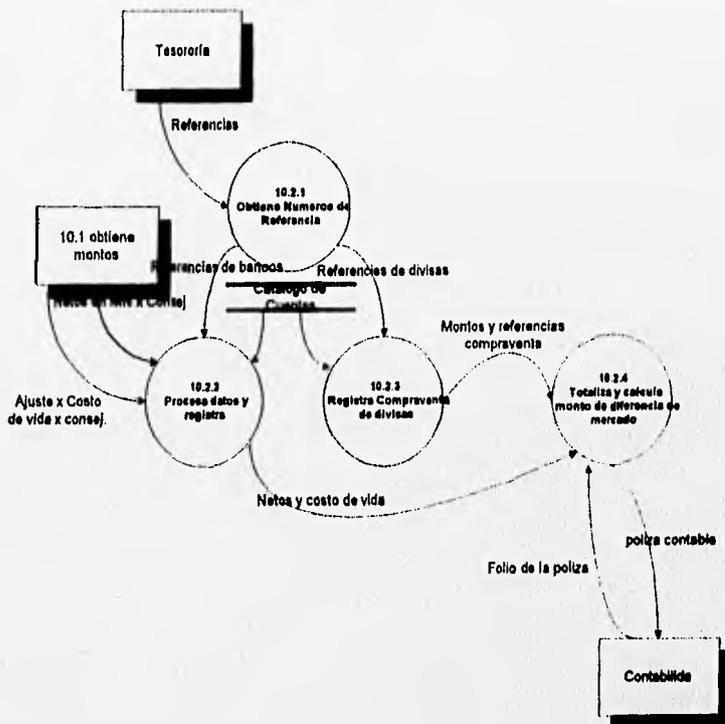
**Diagrama 10**  
**Genera póiza**



**Diagrama 10.1**  
**Obtiene montos**



**Diagrama 10.2**  
**Emite póliza**



### **2.4.2 Especificaciones de proceso**

Es necesario documentar los procesos que quedaron definidos en los diagramas de flujo de datos, por lo cual se debe hacer una descripción clara y estructurada de cada uno de ellos.

#### **1.1.1 Alta de Recuperaciones**

Lee datos de recuperación

Valida monto, consejería y periodo

Si los datos son correctos

Graba Nueva Recuperación

fin datos correctos

#### **1.1.2 Baja de Recuperaciones**

Lee datos de Baja recuperación

Valida periodo y consejería

Si Existe Registro

Pregunta para confirmar

Borra registro

fin existe registro

### **1.1.3 Cambios en recuperaciones**

Lee datos de recuperación

Valida consejería y periodo

Si existe registro

    Lee nuevos datos

**Graba Recuperación modificada**

fin existe registro

### **1.2.1 Alta de envíos**

Lee datos de Envío

Valida monto, consejería y periodo

Si los datos son correctos

**Graba Nuevo Envío**

fin datos correctos

### **1.2.2 Baja de envíos**

Lee datos de Baja de envío

Valida periodo y consejería

Si Existe Registro

Pregunta para confirmar

Borra registro

fin existe registro

### **1.2.3 Cambios en envíos**

Lee datos de Envíos

Valida consejería y periodo

Si existe registro

Lee nuevos datos

Graba Envío modificado

fin existe registro

## **2.1 Alta de Empleados**

Lee datos de Empleado

Valida numero de empleado, consejería y sueldo en moneda extranjera

Si los datos son correctos

Graba Registro de empleado

fin datos correctos

## **2.2 Baja de Empleados**

Lee datos de Baja de empleado

Valida numero de empleado y consejería

Si Existe Registro

Pregunta para confirmar

Borra registro

fin existe registro

### **2.3 Cambios de empleados**

Lee datos de modificación de empleado

Valida clave de empleado y consejería

Si existe registro

    Lee nuevos datos

    Graba Registro de empleado modificado

fin existe registro

### **3.1 Alta de Consejerías**

Lee datos de Consejería

Valida datos de alta de consejería

Si los datos son correctos

    Graba Nueva Consejería

fin datos correctos

### **3.2 Baja de Consejerías**

Lee datos de Baja de consejería

Valida numero de consejería

Si Existe Registro

Pregunta para confirmar

Borra registro

fin existe registro

### **3.3 Cambios en Consejerías**

Lee datos de consejería a modificar

Valida consejería

Si existe registro

Lee nuevos datos

Graba consejería modificada

fin existe registro

#### **4.1 Lee tipos de cambio**

Verifica fecha actual

si fecha es día Hábil

lee tipos de cambio

Si existen todos los tipos de cambio

lee tipos de cambio

fin existen

#### **4.2 Efectúa triangulaciones**

hacer para todas las monedas

Tipo de cambio pesos x dólar entre tipo de cambio moneda extranjera por dólar

igual a triangulación ó pesos x moneda extranjera

fin hacer

#### **4.3 Registra triangulaciones**

Recibe triangulación

Graba en archivo de triangulaciones

#### **5.1.1 Identifica y lee consejería a procesar**

Lee consejería

si no esta dada de baja

pasa número

fin baja

#### **5.1.2 Obtiene moneda de pago**

Analiza registro de monedas

identifica moneda de pago

### **5.2.1 Obtiene movimientos x consejería - empleado**

Con número de consejería hacer

lee catálogo de empleados y obtiene empleado correspondiente a esa consejería

con numero de empleado

lee movimientos en moneda nacional

fin empleado

fin consejería

### **5.2.2 Calcula ajuste x costo de vida**

con movimientos en moneda nacional

sueldo en consejería en MN - Sueldo en México = Diferencia de sueldos

sueldo en México x Impuestos descontados /percepciones que generan impuesto=

parte proporcional de impuesto

Ajuste por costo de vida = Diferencia de Sueldos + parte proporcional de imp.

fin mov.

### **5.2.3 Convierte monto de movimientos a moneda de pago**

importe en mn / triangulación

graba en recibo de pago

### **5.2.4 Acumula montos por empleado**

monto=monto +- importe de siguiente concepto.

graba en catálogo de empleados neto de ese empleado en MN y mon de pago

## **6.1 Escribe telex**

Lee consejería

con número de consejería

lee catálogo de empleados de consejerías

escribe en reporte de telex nombre de empleado y neto a pagar en moneda de pago

fin consejería

## 6.2 Suma montos

Con neto en moneda de pago de cada empleado

neto de consejería en moneda de pago=neto + neto siguiente empleado

fin empleado

## 7.1 Obtiene moneda de concertación

Lee catálogo de monedas de consejerías comerciales

## 7.2 Convierte mon de concertación

busca consejerías con esa moneda de concertación

con número de consejería

lee neto de consejería en Mon. Nal.

convierte a moneda de concertación

fin

### **7.3 Acumula concert**

Si es misma moneda

acumula moneda de concertación

### **7.4 Genera Concerta divisas**

Toma folio de concertación

y graba concertación

### **8.1 Obtiene moneda de Envío**

Lee catálogo de monedas de consejerías comerciales

Obtiene moneda de concertación

## **8.2 Convierte**

Lee catálogo de consejerías

Obtiene moneda de concertación

Convierte monto a moneda de envío

## **8.3 Concerta**

lee folio de bancos

Genera concertación en bancos

## **9.1 Obtiene moneda de envío y procesa datos de consejería.**

Lee catálogo de monedas de consejerías comerciales

Obtiene moneda de envío

## 9.2 Convierte mon

neto en moneda nacional x triangulación

## 9.3 Genera Órdenes de pago para Swift

Obtiene datos de banco

Obtiene datos del responsable

Genera Órdenes de pago

## 9.4 Formatea

lee datos de banco

lee reglas de validación de formato MT100

si los datos están OK

Genera archivo

Si no

Indica error

Fin datos

### **10.1.1 Procesa consejería x moneda**

Lee catálogo de consejerías x moneda hasta el fin

### **10.1.2 Obtiene neto x consejería**

Obtiene el neto en moneda nacional de cada consejería

### **10.1.3 Obtiene Ajuste x costo de vida x consejería**

lee archivo de movimientos de el periodo

obtiene ajuste x costo de vida de esa consejería

### **10.2.1 Obtiene Números de referencia**

con numero de consejería

obtiene referencia de bancos para esa consejería

### **10.2.2 Procesa datos y registra**

graba netos en moneda nacional

graba ajuste x costo de vida de esa consejería

graba neto en moneda de concertación

### **10.2.3 Registra compra venta de divisas**

graba monto de divisas compradas

graba equivalente en pesos

### **10.2.4 Totaliza y calcula monto de diferencia de mercado**

obtiene total cargos

obtiene total de abonos

obtiene diferencia de mercado

### 2.4.3 Diccionario de datos

El diccionario de datos es una agrupación de las definiciones de:

- Orígenes y destinos de la información
- Flujos de datos
- Archivos o Almacenes
- Procesos

Que contiene el sistema.

La técnica de análisis estructurado proporciona una serie de operadores que se aplican a los componentes del diccionario de datos, mismos que se presentan a continuación.

<b>Símbolo</b>	<b>Significado</b>
=	<i>Equivalente a</i>
+	<i>Y</i>
[ ]	<i>Uno u Otro, O</i>
{ }	<i>Iteraciones de</i>
( )	<i>Opcional</i>

En virtud de lo anterior a continuación se definen los elementos del sistema

### **Procesos**

- 1 Registra envíos y recuperaciones
  - 1.1 Registra recuperaciones
    - 1.1.1 Alta de recuperaciones
    - 1.1.2 Baja de recuperaciones
    - 1.1.3 Cambios en recuperaciones
  - 1.2 Registro de envíos
    - 1.2.1 Alta de envíos
    - 1.2.2 Baja de envíos
    - 1.2.3 Cambios en envíos
  
- 2 Mantenimiento al catálogo de empleados de consejerías
  - 2.1 Alta de empleados
  - 2.2 Baja de empleados
  - 2.3 Cambios de empleados
  
- 3 Mantenimiento al catálogo de consejerías
  - 3.1 Alta de consejerías
  - 3.2 Baja de consejerías
  - 3.3 Cambios en consejerías
  
- 4 Obtiene tipos de cambio
  - 4.1 Lee tipos de cambio
  - 4.2 Efectúa triangulaciones
  - 4.3 Registra triangulaciones
  
- 5 Calcula nómina
  - 5.1 Procesa consejería
    - 5.1.1 Identifica y lee consejería a procesar
    - 5.1.2 Obtiene moneda de pago
  - 5.2 Procesa empleado
    - 5.2.1 Obtiene mov x consejería-empleado
    - 5.2.2 Calcula ajuste x costo de vida
    - 5.2.3 Convierte monto de movimientos a moneda de pago
    - 5.2.4 Acumula montos x empleado

- 6      Genera telex
- 6.1    Escribe telex
- 6.2    Suma montos
  
- 7      Concerta divisas
- 7.1    Obtiene mon de concertación
- 7.2    Convierte mon de concertación
- 7.3    Acumula concert
- 7.4    Genera concerta divisas
  
- 8      Concerta bancos
- 8.1    Obtiene moneda de envío
- 8.2    Convierte
- 8.3    Concerta
  
- 9      Genera órdenes de pago
- 9.1    Obtiene mon de envío y procesa datos de consejería
- 9.2    convierte moneda
- 9.3    Genera orden de pago para SWIFT
  
- 10     Genera póliza
- 10.1   Obtiene montos
- 10.1.1 Procesa consejería por moneda
- 10.1.2 Obtiene neto x consejería
- 10.1.3 Obtiene ajuste x costo de vida x consejería
- 10.2   Emite póliza
- 10.2.1 Obtiene números de referencia
- 10.2.2 Procesa datos y registra
- 10.2.3 Registra compraventa de divisas
- 10,2,4 Totaliza y calcula monto de diferencia de mercado

**Entidades externas**

Tesorería  
Oficina de nómina  
Consejerías comerciales  
Contabilidad  
SWIFT y TELEX  
Nómina en moneda nacional

**Archivos o almacenes**

Envíos y recup  
 Triangulaciones  
 Catálogo de empleados  
 Total x empleado  
 Totales MN y mon ext  
 Catálogo de consejerías

**Flujos**

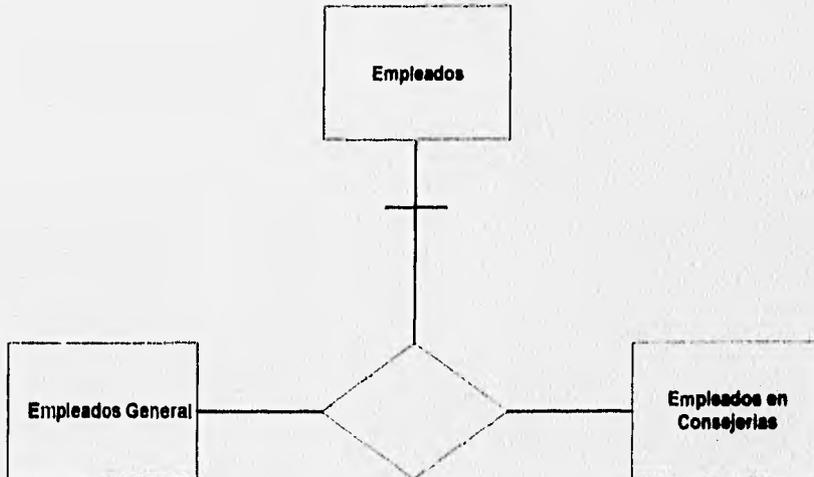
Referencias bancos	=	No de ref concerta divisas + No de ref concerta bancos
No de ref concerta divisas concertación	=	No. de folio que se obtiene al realizar una de divisas.
No. de ref concerta bancos concertación	=	No. de folio que se obtiene al realizar una en bancos.
Concertación de divisas	=	Conjunto de datos que indican la cantidad en moneda extranjera que se necesita comprar para su envío a las consejerías comerciales.
Concertación en bancos	=	Conjunto de datos que indican la cantidad en moneda extranjera a enviar a una consejería a través de un banco, y sirve para actualizar el saldo de la cuenta en ese banco.
Tipo de cambio	=	Es el equivalente cambiario con respecto al dólar de cualquier otra divisa
Formato de transferencia vía SWIFT	=	Es un formato con campos definidos para el envío de fondos vía SWIFT
No. de póliza	=	Es el folio que se asigna a la póliza
Póliza contable	=	Conjunto de registros con los que se realiza el asiento contable.
Conceptos Generados en MN	=	Movimientos que fueron calculados previamente por el proceso de nómina en moneda nacional
Recibos de nómina	=	Es el comprobante de pago de los empleados de consejerías comerciales y presenta el detalle de pago así como el neto a pagar
Relaciones de pago empleado	=	Documento que indica el monto enviado a la consejería y el neto correspondiente a cada empleado
Recuperaciones	=	Son aquellos movimientos que sirven para descontar la cantidad que se indique, a el monto global de la consejería.

- Envíos = Son aquellos movimientos que sirven para incrementar la cantidad que se indique, a el monto global de la consejería.
- Movimientos de consejerías = Datos de una consejería a dar de alta, de baja o modificar
- Movimientos de empleados = Datos de un empleado de consejería a dar de alta, de baja o modificar

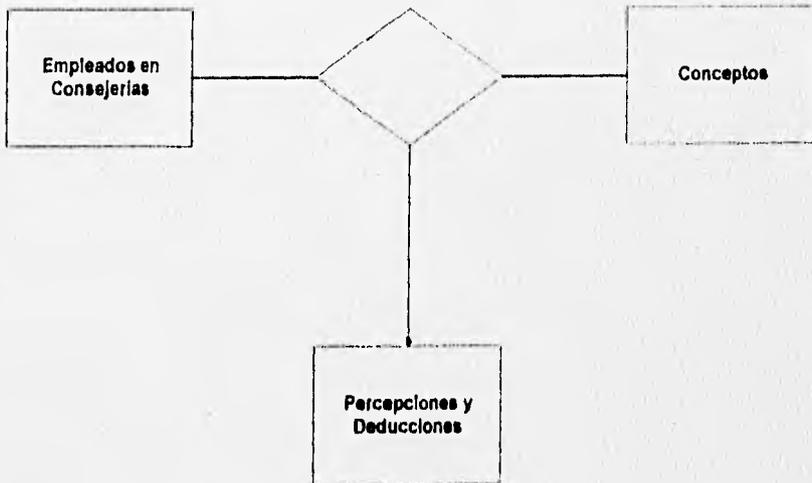
#### 2.4.4 Diagramas de entidad relación

El propósito de los diagramas entidad relación es observar mediante un diagrama la especificación de las reglas de comportamiento de las relaciones.

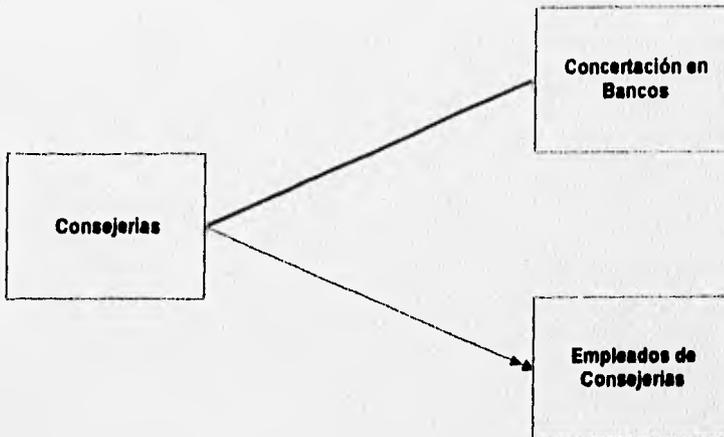
#### Diagramas Entidad Relación



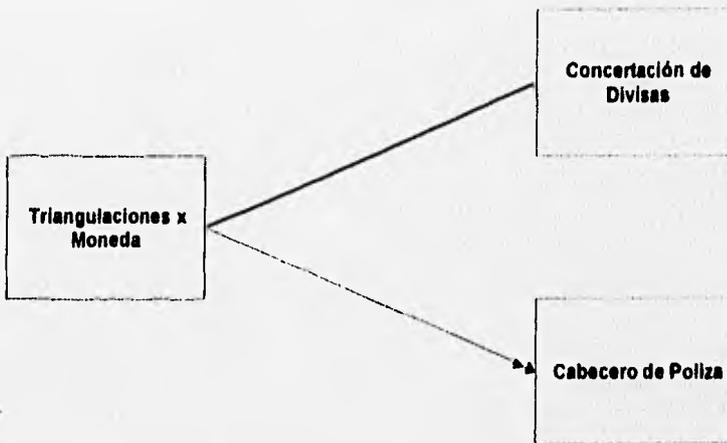
Diagramas Entidad Relación



Diagramas Entidad Relación



### Diagramas Entidad Relación



## CAPÍTULO 3 DESARROLLO DEL SISTEMA

### 3.1 Diseño

El diseño de sistemas puede ser efectuado en una forma metodológica por medio del análisis del flujo de datos asociado con el problema. El análisis de flujo de datos para la nómina de consejerías se llevó a cabo en el capítulo 2 de este trabajo, por lo que dicha información es el suministro para elaborar el diseño, mismo que está soportado por las herramientas que proporciona la técnica de diseño estructurado.

#### 3.1.1 Especificaciones de diseño

##### 3.1.1.1 Diseño estructurado

Esta técnica consiste en diseñar los componentes de un sistema y las interrelaciones entre éstos. Dentro del diseño estructurado se decide cuáles componentes interconectados de alguna forma, resuelven problemas específicos.

La herramienta principal del diseño estructurado es la carta de estructura, la cual muestra la partición del sistema en módulos y la relación jerárquica entre éstos, además muestra los flujos de datos y control entre los módulos.

De la técnica de diseño estructurado, se desprenden dos estrategias de diseño, las cuales son:

Diseño centrado en la transformación

Diseño centrado en la transacción

### ***3.1.1.2 Diseño centrado en la transformación***

Esta estrategia esta basada en el análisis de los flujos de datos.

El diseño no se reduce a una serie de pasos a aplicar, sino más bien, requiere del sentido común por parte del diseñador, para que modele el sistema al rededor de su transformación central.

La estrategia se basa en que el sistema resultante consiste en una estructura jerárquica.

### ***3.1.1.3 Diseño centrado en la transacción***

Esta también esta basada en el análisis de flujo de datos.

Requiere de la definición de transacciones en el diseño. Entendiéndose como transacción cualquier elemento de datos, control, signo, evento o cambio de estado que causa o inicializa una acción o secuencia de acciones, es decir especifica un módulo para cada tipo de transacción.

### ***3.1.2 Diagramas de estructura***

Después de analizar las dos estrategias de diseño estructurado, se utilizará la estrategia de diseño concentrado en la transformación, para lo cual se realizará lo siguiente:

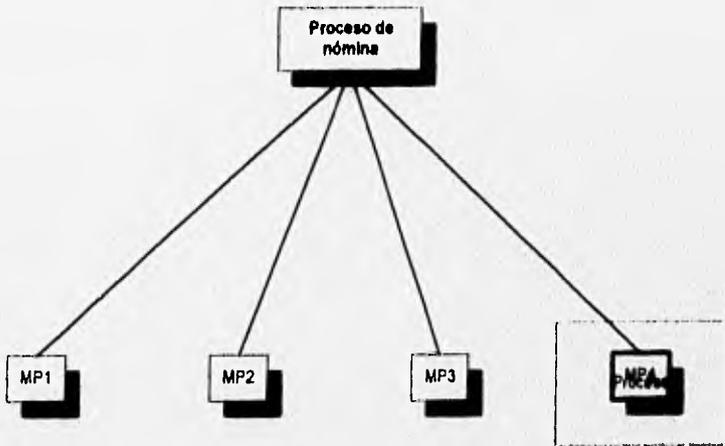
Retomar el problema a partir de los diagramas de flujo de datos.

Identificar los datos que entran y los datos que salen.

Establecer el primer nivel de factorización

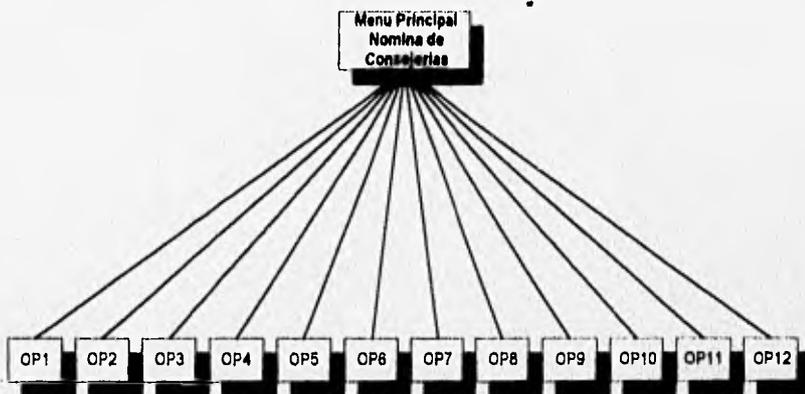
El siguiente diagrama de estructura nos muestra en donde se encuentra ubicado el proceso de nómina de consejerías comerciales dentro de la nómina general del banco.

### Diagrama de Estructura



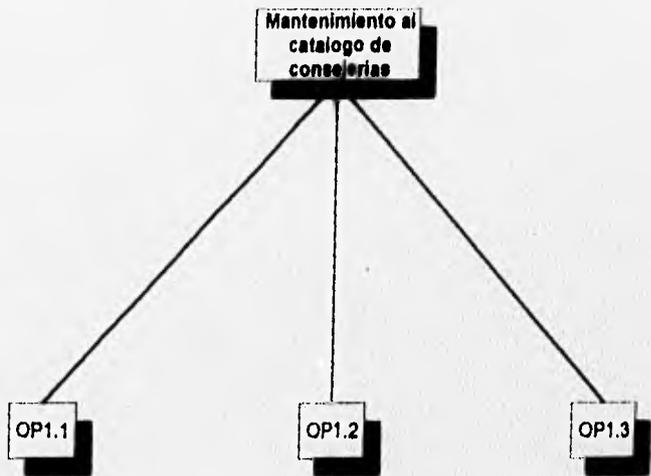
- MP1** Mantenimiento a parámetros de la nómina
- MP2** Control de procesos por excepción
- MP3** Consulta nómina del periodo
- MP4** NOMINA DE CONSEJERIAS COMERCIALES

## Diagrama de Estructura



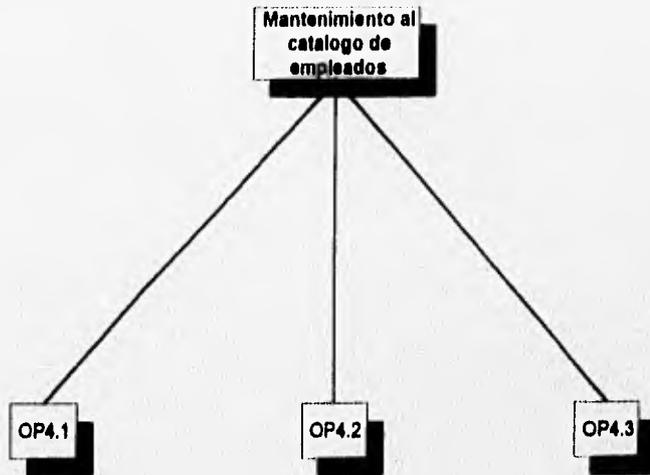
- OP1** Mantenimiento al catalogo de consejerías
- OP2** Actualización de tipos de cambio
- OP3** Concertación de la nómina
- OP4** Mantenimiento al catalogo de empleados
- OP5** Recuperaciones y envíos
- OP6** Paso de concertaciones
- OP7** Validación de la póliza y Paso
- OP8** Generación de telex
- OP9** Consultas
- OP10** REportes
- OP11** Nómina Automática

## Diagrama de Estructura OP 1



- OP1.1 Alta de consejerías comerciales**
- OP1.2 Modificación de consejerías comerciales**
- OP1.3 Baja de consejerías comerciales**

## Diagrama de Estructura OP 4

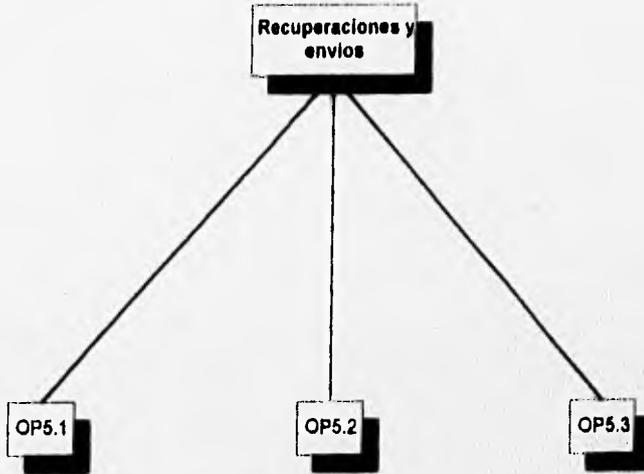


**OP4.1 Alta de empleados de consejerías comerciales**

**OP4.2 Modificación de empleados de consejerías comerciales**

**OP4.3 Baja de empleados de consejerías comerciales**

## Diagrama de Estructura OP 5



- OP5.1** Alta de recuperaciones y envíos
- OP5.2** Modificación de recuperaciones y envíos
- OP5.3** Baja de recuperaciones y envíos

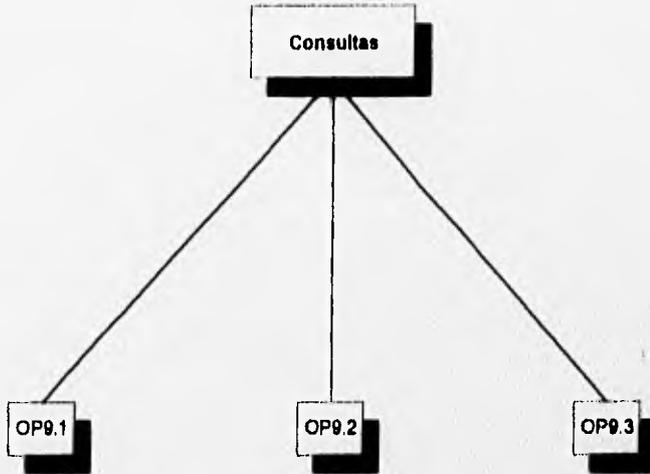
## Diagrama de Estructura OP 7



**OP7.1 Validación de la poliza de Consejerías Comerciales**

**OP7.2 Traspaso de movimientos contables al a contabilidad**

## Diagrama de Estructura OP 9

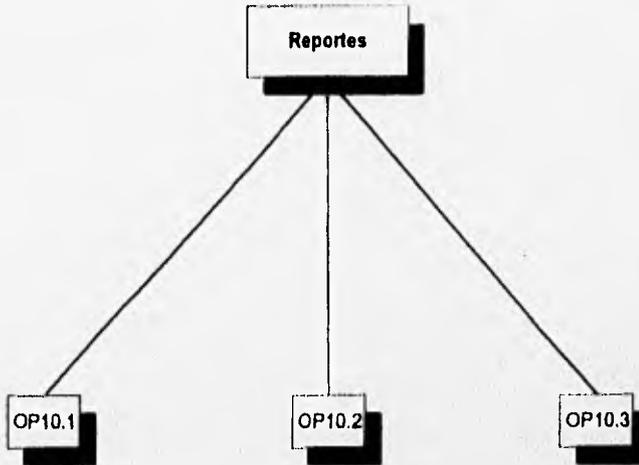


***OP9.1 Consulta a la nómina del periodo***

***OP9.2 Consulta movimientos de PEA***

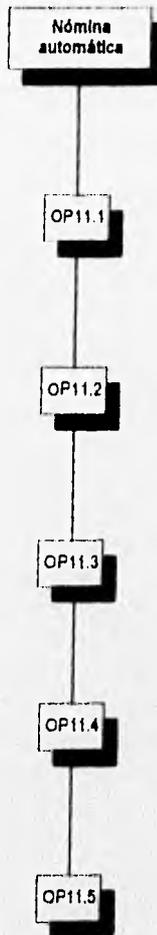
***OP9.3 Consulta Netos por consejería***

## Diagrama de Estructura OP 10



- OP10.1** *Reporte de cifras control*
- OP10.2** *Catálogo de empleados de Consejerías*
- OP10.3** *Catálogo de Consejerías*

## Diagrama de Estructura OP 11



**OP11.1** *Cálculo de nómina de consejerías comerciales*

**OP11.2** *Generación de concertaciones*

**OP11.3** *Generación de TELEX*

**OP11.4** *Generación de poliza contable*

**OP11.5** *Generación de ordenes de pago para SWIFT*











Ahora se presenta el catálogo de bancos

	Nombre de banco	Número de cuenta	Dirección

Tabla para almacenar el histórico de ajuste x costo de vida

Número de empleado	Periodo	Monto

Tabla de seguridad en ella se almacena el nombre del programa y el tipo de autoridad que tiene un usuario a través de su perfil a ese programa.

Perfil	Programa	Tipo de autoridad

Catálogo de monedas

Moneda	Descripción	Tipo

### 3.1.4 Diseño e implementación de los diversos módulos de programación

En seguida se presenta el pseudocódigo de los programas que realizan el cálculo de la nómina automática.

#### **Programa de cálculo de la nómina para consejerías comerciales**

Ubicado en la opción 11.1 de los menús de nómina de consejerías comerciales.

Archivos	Tipo
Catálogo de empleados de consejerías	Entrada
Catálogo de consejerías	Entrada
Total x empleado	Salida
Percepciones y deducciones	Entrada y Salida
Tipos de cambio	Entrada
Catálogo de conceptos	Entrada
Catálogo de procesos	Entrada
Recibo de Nómina	Salida

#### **Inicio**

Inicializa

Procesa consejería hasta que sea fin de archivo

**fin**

**Inicializa**

lee numero de periodo a procesar

inicializa banderas

fin-inicializa

**Procesa consejería**

lee catálogo de consejerías

si es fin de archivo

regresa

si no

obtiene número de consejería

obtiene moneda de pago

lee archivo de tipos de cambio

obtiene el equivalente en pesos de la moneda de pago

procesa empleados pertenecientes a la consejería

fin

fin proceso consejerías

**Procesa empleado de consejerías**

lee empleado de consejerías

si es fin de archivo o numero de consejería mayor a consejería actual

regresa

si no

obtiene sueldo mensual en moneda de pago

importe en pesos de sueldo = (sueldo en moneda local / 2) \* (equivalente en pesos de la moneda local)

imprime encabezado de reporte

obtiene numero de empleado

procesa percepciones y deducciones

Impuesto = ptu gravable + ptu exento

parte proporcional de impuesto = (sueldo en México \* ISR)/(total de  
percepciones que causan impuesto)

Ajuste x costo de vida = sueldo bruto - sueldo en México + parte  
proporcional de ISR

Graba ajuste en el archivo de percepciones y deducciones

ajuste en moneda de pago = importe de ajuste en pesos/equivalente en pesos  
de moneda de pago

escribe ajuste en ambas monedas

Escribe totales en ambas monedas

fin de catálogo de empleados de consejerías

fin procesa empleados de consejerías

### Procesa percepciones y deducciones

lee percepciones y deducciones

si es fin de archivo o numero de empleado es diferente al empleado procesado

regresa

si no

importe del concepto en moneda de pago = importe del concepto moneda /  
equivalente en pesos de la  
moneda de pago

obtiene número de concepto

lee catálogo de conceptos

obtiene nombre del concepto

identifica tipo de concepto

si es percepción

    escribe percepción en moneda nacional y moneda de pago

    acumula percepción en ambas monedas

sino

    escribe deducción en moneda nacional y moneda de pago

    acumula deducción en ambas monedas

fin de percepción

fin de archivo de percepciones y deducciones

fin procesa percepciones y deducciones

**Escribe percepción**

    mueve importe del concepto en pesos

    mueve importe del concepto en moneda de pago

    mueve nombre del concepto

    mueve número del concepto

    escribe detalle de reporte de percepción

fin escribe percepción

**Escribe deducción**

    mueve importe del concepto en pesos

    mueve importe del concepto en moneda de pago

    mueve nombre del concepto

    mueve número del concepto

    escribe detalle de reporte deducción

fin escribe deducción

**Programa que obtiene tipos de cambio.**

Ubicado en la opción 02 del menú de nómina de consejerías comerciales.

Archivos	Tipo
Catálogo de tipos de cambio	Entrada
Archivo de triangulaciones	Entrada-Salida

Inicio

Obtiene fecha de proceso

Procesa moneda

fin

Obtiene fecha de proceso

lee día de proceso del sistema

lee mes de proceso del sistema

lee año de proceso del sistema

fin obtiene fecha de proceso

Procesa moneda

lee archivo de triangulaciones

obtiene número de moneda

Obtiene tipo de cambio

Obtiene triangulación

graba datos

fin procesa moneda

Obtiene tipo de cambio

lee archivos de tipo de cambio

obtiene tipo de cambio

**Programa de cálculo de la póliza de la nómina para consejerías comerciales**

Ubicado en la opción 11.4 de los menús de nómina de consejerías comerciales.

Archivos	Tipo
Catálogo de empleados de consejerías	Entrada
Catálogo de consejerías	Entrada
Percepciones y deducciones	Entrada
Tipos de cambio	Entrada
Recuperaciones y envíos	Entrada
Catálogo de procesos	Entrada
Archivo de póliza	Salida

**Inicio**

procesa moneda hasta que sea fin de archivo

**Fin**

**Procesa moneda**

obtiene moneda

obtiene tipo de cambio

procesa consejerías

fin procesa moneda

procesa consejerías

lee consejería

obtiene moneda de concertación de la consejería

si moneda de concertación de la consejería es igual a moneda procesada

obtiene total en pesos en moneda de concertación

procesa empleados

escribe detalle

fin

fin procesa consejerías

procesa empleados

lee empleado de consejería procesada

acumula neto en moneda nacional sin ajuste

acumula total de ajuste por costo de vida

fin procesa empleados

Escribe detalle

si es primera vez o fin de pagina

escribe encabezado

fin

escribe cargo de acumulado netos de moneda nacional a cuenta de activos

escribe cargo de acumulado ajuste a cuenta de activos

Fin escribe detalle

**Programa que genera los telex para enviar a la consejerías comerciales**

Ubicado en la opción 08 del menú de nómina de consejerías comerciales

Archivos	Tipo
Catálogo de empleados de consejerías	Entrada
Catálogo de consejerías	Entrada
Catálogo de procesos	Entrada
Reporte de telex	Salida

**Inicio**

procesa consejerías

**Fin**

procesa consejerías

lee consejerías

escribe encabezado

escribe neto a enviar (moneda de cobro)

procesa empleados de consejerías

fin procesa consejerías

procesa empleados de consejerías

lee empleado

si es de la consejería en proceso

escribe datos de empleado

escribe neto en moneda de cobro

fin

fin procesa empleados de consejería

**Programa que genera órdenes de pago para envío de fondos vía SWIFT**

Ubicado en la opción 11.5 de los menús de nómina de consejerías comerciales.

Archivos	Tipo
Catálogo de consejerías	Entrada
Interface SWIFT	Salida
Reporte de órdenes de pago	Salida

**Inicio**

procesa consejería

**Fin**

procesa consejería

lee consejería

escribe datos de consejería

escribe datos de banco destino

escribe datos de banco local

escribe monto a enviar en moneda de envío

fin procesa consejería

## ***Conclusiones***

Se puede concluir que este sistema ha cumplido con los requerimientos necesarios para el objetivo que se planeó, ya que se logró automatizar satisfactoriamente el proceso de Nómina de Consejerías Comerciales del Banco Nacional de Comercio Exterior.

Además se logró mostrar las ventajas que da el desarrollo de sistemas utilizando las técnicas de análisis y diseño estructurado, dando como resultado sistemas flexibles de fácil mantenimiento y sobre todo modulares.

Con este sistema se logró reducir el tiempo del proceso que era de dos a tres días a solo cinco minutos.

Es importante mencionar que el sistema incluye interfaces con los siguientes sistemas:

Con el sistema de control de divisas, para obtener los tipos de cambio del día de cada una de las monedas con respecto al dólar y realizar la concertación de divisas.

Con el sistema de bancos para realizar la concertación en bancos de los montos a enviar a las consejerías.

Con el sistema de contabilidad para realizar la afectación contable de el proceso de Nómina de Consejerías Comerciales.

Se encuentra actualmente en uso por 29 Consejerías Comerciales alrededor de todo el mundo.

## ***BIBLIOGRAFÍA***

Yourdon, Edward, y Larry L. Constantine. Structured Design.,  
Prentice-Hall, Inc., 1991

Marco De, Tom. Structured Analisis and System Specification.,  
Prentice-Hall, Inc. , 1979

Gillenson, Mark L. ;Introducción a las bases de datos;  
McGraw-Hill; 1a. Edición, 1988

Fairley, Richard E. Ingeniería de Software.,  
McGraw-Hill; México, 1990

Kendall y Kendall. Análisis y Diseño de Sistemas.,  
Prentice Hall, México 1990

Pressman, Roger S. Ingeniería del Software.,  
McGraw-Hill, 1988