

01132  
64



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**“IMPLEMENTACIÓN DE UN SISTEMA  
BANCARIO DE CONTROL DE PAGOS”**

## **TESIS**

QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERO EN COMPUTACIÓN**

PRESENTA:  
**ISRAEL MENDOZA PACHECO**

DIRECTOR DE TESIS:  
**ING. LUCILA PATRICIA ARELLANO MENDOZA**

**TESIS CON  
FALLA DE ORIGEN**



MÉXICO, D.F.

2003



Universidad Nacional  
Autónoma de México



**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.

# **PAGINACIÓN DISCONTINUA**

## PALABRAS DE AGRADECIMIENTO

### **A Dios**

Por la vida y por permitirme concluir una más de mis metas.  
Gracias.

### **A mis padres Mario y Rosa María**

Por el cariño y los valores que me inculcaron desde niño. A mi Papá por su Fuerza de Voluntad y Ganas de Superación que logró trasmitirme. A ti Mamá por el amor y cuidados que siempre tuviste hacia mí.  
Gracias.

### **A mi esposa Bety**

A esa gran mujer por el inmenso amor que a diario me da, sin pedir nada a cambio, por tu confianza y por tus consejos que siempre me motivan para seguir adelante.  
Gracias.

### **A mis hijos Arturo y Daniel**

Por su Amor, Travesuras, Sonrisas y Juegos que siempre iluminan mi camino y me dan fortaleza para seguirme superando como persona.  
Gracias.

### **A mis hermanos Araceli, Mario Alberto, Wendy y Julieta**

Por todos los momentos agradables que compartimos de niños, reciban mi cariño y admiración, especialmente para July, nunca te des por vencida, siempre lucha por tus ideales y concluye todas las metas que te fijes.  
Gracias.

### **A mis tíos**

A Francisco, Felipa, Enrique, Cecilio y Carlos por el cariño y amor que llenaron de ilusiones mi infancia.  
Gracias.

### **A mis abuelos**

A Juanita y Francisco por ese inmenso amor que recibí de ustedes, a Jovita por su temple y carácter que me enseñaron a valorar las cosas.  
Gracias.

### **A la Sra. Gloria Gómez Saucedo**

Por su apoyo y afecto recibido en esta última etapa de mi formación académica.  
Gracias.

### **A la Ing. Lucila Patricia Arellano Mendoza**

Por su experiencia, conocimientos y orientación que sirvieron de guía para poder ver cristalizado este proyecto.  
Gracias.

### **A la UNAM y la Facultad de Ingeniería**

Por abrirme sus puertas y albergarme durante todos estos años.  
Gracias.

### **A mis profesores**

A todos los Profesores de la Facultad de Ingeniería, por sus conocimientos y tiempo que dedicaron para mi formación académica.  
Gracias.

**ISRAEL MENDOZA PACHECO**

# ÍNDICE

Prólogo.....	viii
1 Introducción.....	01
1.1 Antecedentes.....	02
1.2 Perspectivas del sistema.....	03
1.3 Teoría de bases de datos.....	04
1.3.1 Arquitectura de base de datos.....	05
1.3.2 Modelo de bases de datos.....	06
1.3.3 Sistema manejador de bases de datos (DBMS).....	08
1.3.4 Esquema conceptual.....	09
1.3.5 Esquema de la base de datos.....	09
1.3.6 Diccionario de recursos de información.....	10
1.4 Metodología.....	11
1.4.1 Modelos de datos.....	12
1.4.1.1 Modelos lógicos basados en objetos.....	13
1.4.1.1.1 Modelo entidad-relación.....	13
1.4.1.1.2 Modelo orientado a objetos.....	16
1.4.1.2 Modelos lógicos basados en registros.....	19
1.4.1.2.1 Modelo Relacional.....	19
1.4.1.2.2 Modelo de Red.....	27
1.4.1.2.3 Modelo Jerárquico.....	29
1.4.2 Modelo de procesos.....	30
1.4.2.1 Método estructurado de datos.....	31
1.4.2.1.1 Diagrama de Flujo de Datos.....	31
1.4.2.1.2 Diagrama de estructura.....	34
1.4.2.2 Método orientado a objetos.....	37
1.4.2.2.1 Modelo de objeto.....	37
1.4.2.2.2 Modelo dinámico.....	40
1.4.2.2.3 Modelo funcional.....	41
1.4.3 Modelo tecnológico.....	42

2	Análisis.....	44
	2.1 Objetivo del sistema.....	45
	2.2 Definición del problema.....	45
	2.3 Análisis de la situación actual.....	48
	2.3.1 Análisis del Software.....	48
	2.3.2 Análisis del Hardware.....	50
	2.4 Estrategia de solución.....	51
	2.5 Justificación del sistema.....	58
	2.6 Modelo del sistema.....	59
	2.6.1 Modelo de datos.....	59
	2.6.1.1 Diagrama entidad-relación Sistema para el Trámite de Pagos.....	60
	2.6.1.2 Diagrama entidad-relación Sistema Administrativo-Contable de Pagos.....	64
	2.6.2 Modelo de procesos.....	74
	2.6.2.1 Diagrama de Flujo Sistema para el Trámite de Pagos.....	75
	2.6.2.2 Diagrama de Flujo Sistema Administrativo-Contable de Pagos.....	84
	2.6.3 Modelo tecnológico.....	92
	2.6.3.1 Back-end.....	92
	2.6.3.2 Front-end.....	97
	2.6.3.3 Características del sistema.....	99
3	Diseño.....	101
	3.1 Diseño de datos.....	102
	3.1.1 Sistema para el Trámite de Pagos.....	102
	3.1.2 Sistema Administrativo-Contable de Pagos.....	109
	3.2 Módulos del sistema.....	132
	3.2.1 Sistema para el Trámite de Pagos.....	132
	3.2.2 Sistema Administrativo-Contable de pagos.....	140

4	Construcción.....	154
4.1	Sistema para el Trámite de Pagos.....	155
4.1.1	Relaciones.....	155
4.1.2	Formularios.....	160
4.1.3	Informes.....	168
4.1.4	Menús.....	174
4.1.5	Código.....	179
4.2	Sistema Administrativo-Contable de Pagos.....	194
4.2.1	Relaciones.....	194
4.2.2	Formularios.....	201
4.2.3	Informes.....	203
4.2.4	Menús.....	206
4.2.5	Código.....	207
5	Implantación.....	223
5.1	Pruebas y Ajustes.....	224
5.1.1	Tipos de Pruebas.....	225
5.2	Capacitación del Usuario.....	227
5.3	Documentación del Sistema.....	229
5.3.1	Manual del usuario.....	230
5.3.1.1	Sistema para el Trámite de Pagos.....	231
5.3.1.2	Sistema Administrativo-Contable de Pagos.....	250
5.3.2	Manual Técnico.....	276
5.3.2.1	Sistema para el Trámite de Pagos.....	277
5.3.2.2	Sistema Administrativo-Contable de Pagos.....	282
5.4	Mantenimiento al Sistema.....	290
5.4.1	Perfectivo.....	291
5.4.2	Adaptativo.....	292
	Conclusiones.....	294
	Bibliografía.....	299



## PRÓLOGO

El presente trabajo de tesis tiene como objetivo el desarrollo e implementación de un sistema de base de datos para controlar y dar seguimiento a los gastos operativos que tienen las Instituciones Bancarias.

El trabajo está organizado de la siguiente manera:

En el primer capítulo se muestra un panorama general de la problemática que presentan algunas Instituciones Bancarias por la falta de información estadística, lo que les impide tener un control adecuado sobre los gastos originados por la operación de sus servicios internos, se establece la propuesta de solución, indicando los beneficios que se obtendrán.

Para sustentar la solución propuesta, se realiza una breve introducción sobre los fundamentos de bases de datos, metodologías y modelos de información utilizados para la implementación de sistemas.

El segundo capítulo contiene el análisis del sistema, en él se expone el objetivo, el problema a resolver y sus posibles alternativas de implementación, aplicando para tal efecto, la teoría vista en el primer capítulo, en lo que se refiere al modelo de datos se mostrarán los diagramas de entidad relación del sistema, para el modelo de procesos se mostrarán los diagramas de flujo de datos y finalmente en lo referente al modelo tecnológico se mostrarán las principales características de la herramienta seleccionada para el desarrollo del sistema.

En el tercer capítulo se realiza el diseño de las relaciones establecidas en los diagramas entidad-relación, además, de acuerdo a los diagramas de flujo de datos establecidos en el capítulo anterior se llevará a cabo el diseño de los módulos del sistema.

El cuarto capítulo contiene la construcción de las relaciones, formularios, informes, menús y código diverso, realizando una breve explicación de la forma en que se llevan a cabo estas acciones utilizando la herramienta seleccionada para el desarrollo del sistema.

En el quinto capítulo se muestran los procedimientos para realizar las pruebas y ajustes del sistema antes de su implementación, se hace referencia a la capacitación del usuario, así como a la documentación del sistema, incluyendo en este apartado el manual del usuario y el manual técnico.

Finalmente se incluyen las conclusiones de este trabajo.



## 1.1 ANTECEDENTES

El paso del tiempo ha demostrado que la información proveniente de un proceso de cómputo se ha convertido en parte medular para la toma de decisiones, por ende, para tener un control eficiente de los servicios internos de una Institución Bancaria, se deberá contar con aplicaciones específicas, que la hagan competitiva en su entorno, dichas aplicaciones deberán procesar datos y generar información determinada tanto en línea, como en informes históricos y estadísticos.

Ahora bien, para que una Institución Bancaria sea rentable, ésta deberá contar con una operación óptima, con gastos operativos regidos bajo un esquema de política y procedimientos que le permitan tener costos bajos e indispensablemente necesarios, si nos enfocamos al control de gastos, los factores que se deben considerar para la consecución de éstos objetivos son: contar con proveedores que ofrezcan una pronta atención a cualquier problema o eventualidad a un costo aceptable; contar con el personal necesario y eficiente en las actividades que se le encomienden y por último contar con sistemas que faciliten el trabajo y hagan más productivos a los empleados.

Por lo anterior, las Instituciones Bancarias tienen la necesidad de conocer en forma veraz y eficiente todos los gastos originados por la operación de sus servicios internos, lo cual les permitirá realizar ajustes o recortes en las áreas que generen gastos excesivos, así como incentivar a los proveedores que ofrecen un servicio de excelencia, retribuyéndolos con un pago oportuno y de forma automática a sus cuentas bancarias.

Actualmente, algunas Instituciones Bancarias llevan un control manual a través de hojas de cálculo en Excel de todos sus gastos originados por su operación diaria. Sus áreas administrativas que originan gasto cuenta con personal que realiza funciones propias del área contable, lo que origina duplicidad de funciones y recaptura de información, provocando con esto errores al momento de realizar la afectación contable del gasto y por otra parte el tiempo para el pago a los prestadores de servicios se eleva considerablemente.

El pago al prestador de servicios se realiza vía emisión de un cheque, con lo que se desaprovecha la banca electrónica de los principales bancos, haciendo uso de este recurso se disminuirán notablemente las comisiones que se cobran por la emisión y dotación de cheques.

## 1.2. PERSPECTIVAS DEL SISTEMA

Dotar a las Instituciones Bancarias de un Sistema Integral de Pagos, que permita automatizar los trámites solicitados al Área de Central de Pagos y que a su vez permita llevar el control y seguimiento contable de los pagos realizados.

Esto se logrará a través de la implementación de:

- Un Sistema para el Trámite de Pagos: Este brindará apoyo a todas las áreas administrativas que soliciten un pago.
- Un Sistema Administrativo-Contable de Pagos: El cual permitirá al Área de Central de Pagos controlar y dar seguimiento a todos los pagos efectuados.

Dentro de los beneficios que se obtendrán con la implantación de estos sistemas podemos mencionar los siguientes:

- Automatización del procedimiento que se sigue para el trámite de cualquier pago solicitado al Área de Central de Pagos, tales como: anticipos y comprobaciones de viaje y viáticos de empleados, gastos de servicios administrativos( teléfono, luz, rentas de inmuebles, etc.)
- Unificación y normalización de criterios para la solicitud de pagos.
- Dotar de un sistema único, ágil y sencillo a todas las áreas solicitantes de pagos.
- Sistematizar la emisión de pólizas contables para evitar errores en el llenado de éstas.
- Optimizar el registro de información en la emisión de cheques.

- Automatizar el pago de facturas.
- Automatizar la emisión de reportes y formatos.
- Automatizar los cálculos que se realizan manualmente.
- Generar consultas de información por diferentes criterios.
- Garantizar la seguridad en la emisión de cheques y pago de facturas.

Será responsabilidad del Área de Central de Pagos emitir el pago correspondiente a los empleados y funcionarios de la institución, proveedores y prestadores de servicios a través de depósitos en cuenta de cheques o de débito de cualquier banco, quedando abierta la posibilidad de que excepcionalmente se puedan emitir cheques.

Será responsabilidad del área usuaria presentar en la Ventanilla de Central de Pagos los comprobantes que amparen el pago de los gastos debidamente autorizados por el funcionario facultado.

### **1.3 TEORÍA DE BASES DE DATOS**

Hasta hace relativamente poco tiempo, las bases de datos eran el resultado de una compleja programación y de complicados mecanismos de almacenamiento. Con el auge de la microinformática, la aparición de aplicaciones específicas también trajo con ella la disponibilidad de herramientas de gestión de datos, que acabaron desembocando en los denominados sistemas de gestión de bases de datos, identificados por sus siglas DBMS (DataBase Management Systems). De esta manera, la gestión de base de datos pudo liberarse de los grandes ordenadores centrales, pudiendo distribuirse según los intereses de los usuarios, y dotando de autonomía en la gestión de información a muchas entidades. Los DBMS permitieron a todo tipo de usuarios crear y mantener sus bases de datos, dotándolos de una herramienta capaz de transformar el nivel lógico que éstos diseñaban en un conjunto de datos, representaciones y relaciones, traduciéndolo al nivel físico correspondiente.

### 1.3.1 ARQUITECTURA DE BASE DE DATOS

Las bases de datos respetan la arquitectura de tres niveles definida para cualquier tipo de base de datos, por el grupo ANSI/SPARC (Standard Planning and Requirements Committee of the American National Standards Institute on Computers and Information Processing). En esta arquitectura la base de datos se divide en tres niveles: (fig. 1.1)

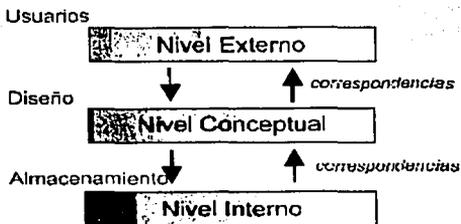


Figura 1.1 Arquitectura de Bases de Datos

- **Nivel interno:** Es el nivel más bajo de abstracción y define cómo se almacenan los datos en el soporte físico, así como los métodos de acceso.
- **Nivel conceptual:** Es el nivel medio de abstracción. Se trata de la representación de los datos realizada por la organización, que recoge las vistas parciales de los requerimientos de los diferentes usuarios y las aplicaciones posibles. Se configura como visión organizativa total, e incluye la definición de datos y las relaciones entre ellos.

En ocasiones puede encontrarse el nivel conceptual dividido en dos niveles, conceptual y lógico. El primero de ellos corresponde a la visión del sistema global desde un punto de vista no informático. El segundo corresponde a la visión de la base de datos expresada en términos del sistema que se va a implantar con medios informáticos.

- **Nivel externo:** Es el nivel de mayor abstracción. A este nivel corresponden las diferentes vistas parciales que tienen de la base de datos los diferentes usuarios. En cierto modo, es la parte del modelo conceptual a la que tienen acceso.

El modelo de arquitectura propuesto permite establecer el principio de independencia de los datos. Esta independencia puede ser lógica y física. Por independencia lógica se entiende que los cambios en el esquema lógico no deben afectar a los esquemas externos que no utilicen los datos modificados. Por independencia física se entiende que el esquema lógico no se vea afectado por cambios realizados en el esquema interno, correspondientes a modos de acceso, etc.

La capacidad de compartir es uno de los beneficios más importantes de los sistemas de bases de datos, las aplicaciones existentes pueden hacer referencia a los mismos datos y las nuevas aplicaciones pueden hacer alusión a los datos existentes.

Algunas ventajas de los sistemas de bases de datos son:

- Se elimina la redundancia de datos.
- Se garantiza la integridad de los datos.
- Los datos pueden ser compartidos.
- Se elimina la inconsistencia de información.
- Se pueden aplicar restricciones de seguridad.

### **1.3.2 MODELO DE BASES DE DATOS**

En el proceso de abstracción que conduce a la creación de una base de datos desempeña una función prioritaria el modelo de datos. El modelo de datos, como abstracción del universo de discurso, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos.

Los objetivos del modelo de datos son dos:

1. Formalización: Definir formalmente las estructuras permitidas y las restricciones a fin de representar los datos de un sistema.
2. Diseño: El modelo resultante es un elemento básico para el desarrollo de la metodología de diseño de la base de datos.

Los diferentes modelos de datos comparten, aunque con diferentes nombres y notaciones, unos elementos comunes, componentes básicos de la representación de la realidad que abstraen. Estos componentes se identifican gracias a la clasificación y pueden definirse como conceptos estáticos y conceptos dinámicos.

Los conceptos estáticos corresponden a:

- Objeto: Cualquier entidad con existencia independiente sobre el que almacenan datos. Puede ser simples o compuestos.
- Relación: Asociación entre objetos.
- Restricción estática: Propiedad estática del mundo real que no puede expresarse como los anteriores ya que sólo se da en la base de datos; suele corresponder a valores u ocurrencias, sobre atributos, entidades y relaciones.
- Objeto compuesto: Definidos como nuevos objetos dentro de la base de datos, tomando como punto de partida otros existentes, mediante mecanismos de agregación y asociación.
- Generalización: Se trata de relaciones de subclase entre objetos, es decir, parte de las características de diferentes entidades pueden resultar comunes entre ellas.

Por su parte, los conceptos dinámicos responden a:

- Operación: Acción básica sobre objetos o relaciones (crear, modificar, eliminar...).

- Transacción: Conjunto de operaciones que deben ejecutarse en su conjunto obligatoriamente.
- Restricción dinámica: Propiedades del mundo real que restringen la evolución en el tiempo de la base de datos.

### 1.3.3 SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

Para plasmar los tres niveles en el enfoque o modelo de datos seleccionado, es necesaria una aplicación que actúe de interfaz entre el usuario, los modelos y el sistema físico. Ésta es la función que desempeñan los DBMS y que pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario.

Las principales funciones que debe cumplir un DBMS se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

El DBMS incorpora como herramienta fundamental para la definición y la manipulación de los datos dos lenguajes:

- El lenguaje de definición de datos (DDL, Data Definition Language): Provee de los medios necesarios para definir los datos con precisión, especificando las distintas estructuras. Acorde con el modelo de arquitectura de tres niveles, habrá un lenguaje de definición de la estructura lógica global, otro para la definición de la estructura interna y un tercero para la definición de las estructuras externas.
- El lenguaje de manipulación de datos (DML, Data Manipulation/ Management Language): Es el encargado de facilitar a los usuarios el acceso y manipulación de los

datos. Pueden diferenciarse en procedimentales (aquellos que requieren qué datos se necesitan y cómo obtenerlos) y no procedimentales (que datos se necesitan, sin especificar cómo obtenerlos), se encargan de la recuperación de los datos almacenados, de la inserción y supresión de datos en la base de datos y de la modificación de los existentes.

#### 1.3.4 ESQUEMA CONCEPTUAL

El esquema conceptual puede definirse como una descripción abstracta y general de la parte o sector del universo real que el contenido de la base de datos va a representar. En este nivel de análisis se está tratando con una descripción de la realidad, no con datos, y suele contener listas de tipos de entidades, de las relaciones existentes entre esas entidades y de las restricciones de integridad que se aplican sobre ellas.

El esquema conceptual de la base de datos puede utilizarse para integrar los intereses de los diferentes usuarios, como herramienta de representación y de formación, así como para prever futuras modificaciones del sistema.

#### 1.3.5 ESQUEMA DE LA BASE DE DATOS

Una vez construido el esquema conceptual, el diseño de bases de datos obliga a realizar varias tareas previas a la construcción del esquema lógico global del sistema, también llamado esquema de bases de datos.

El esquema de la base de datos representa la descripción de los datos, mientras que el esquema conceptual representaba a la realidad, para establecer la definición del esquema de la base de datos, básicamente se tienen que realizar tres pasos:

- Identificación de los datos requeridos: Para obtener como resultado las partes del área de aplicación que deben representarse mediante datos, y en que forma deben presentarse éstos a los usuarios.

- Análisis de datos: Consistente en la definición y clasificación de esos datos, su descripción suele presentarse en forma de diccionario de datos, como una "metabase de datos".
- Especificación de los paquetes de entrada y de salida: Correspondientes con los datos que deben introducir y obtener como respuesta los usuarios, según sus necesidades.

Tomando como punto de partida estos tres elementos, se construye la especificación del esquema de la base de datos, que responderá al contenido total de la base de datos y las características de las vías de acceso requeridas a través de estos datos.

### 1.3.6 DICCIONARIO DE RECURSOS DE INFORMACIÓN

La gestión efectiva de los datos involucrados en la base de datos implica necesariamente disponer de alguna herramienta que controle las características y funciones de aquéllos. Esta función es cubierta mediante el diccionario de recursos de información (DRI), que asegura la integración de toda la información contenida en el sistema.

Se habla entonces de metadatos, como datos que definen y describen los datos existentes en el sistema. En un primer momento, este tipo de cuestiones eran resueltas a través de los diccionarios de datos, que reunían información sobre los datos almacenados, sus descripciones, significados, restricciones, usos, etc., y los directorios de datos, subsistemas del sistema de gestión, encargados de describir dónde y cómo se almacenaban los datos. Actualmente se aplica el concepto de diccionario de recursos de información, que engloban todo lo señalado anteriormente, dando lugar a lo que ha pasado a llamarse "metabases".

Algunos de los beneficios de usar el diccionario de recursos son:

- Describen el significado de los datos en los flujos y almacenamientos.

- Especifican los valores relevantes y unidades de los tipos de datos.
- Describen los detalles de las relaciones entre los datos almacenados.
- Eliminan redundancia e inconsistencia.
- Contienen las restricciones de integridad de cada relación.

## 1.4 METODOLOGÍA

El desarrollo de sistemas debe ser tratado bajo una disciplina de Ingeniería, para que a través de sus métodos y herramientas se planifique el proyecto, se analicen los requisitos del sistema, se diseñen las estructuras de datos y se realicen las pruebas necesarias que garanticen la correcta operación del sistema.

La Metodología para la Integración de Sistemas proporciona un conjunto integrado de procedimientos, estándares y prácticas que garantizan la realización exitosa de proyectos de sistemas a desarrollar.

Fases de la Metodología:

- **Análisis:** En esta fase se identifican los requerimientos y limitaciones del sistema. Se especifica el problema a resolver y sus posibles soluciones. En un concepto general significa, descomponer un problema en partes, resolver lo complejo de lo simple, teniendo siempre como objetivo llegar a conocer los principios o elementos del problema.
- **Diseño:** Es la fase del desarrollo de sistemas que define la organización, estructuras y métodos de acceso óptimos a los datos, que impliquen el manejo de dichos sistemas.
- **Construcción:** En esta etapa se desarrolla la codificación de los programas que previamente se identificaron y especificaron. Estos se elaboran con el uso de algún lenguaje de programación previamente definido; se efectúan tanto las

pruebas individuales por programa, como las pruebas de integración de los módulos que conforman el nuevo sistema.

- **Implantación:** Se pone en operación el nuevo sistema, realizando las pruebas de aceptación para su instalación con el usuario final.

## **INGENIERÍA DE INFORMACIÓN**

Es un conjunto de técnicas formales, mediante las cuales se establecen modelos de datos y modelos de procesos, cuyo propósito es crear y mantener sistemas de procesamiento de datos.

Para el desarrollo de un Sistema de Información existen tres modelos comúnmente utilizados:

- **MODELO DE DATOS**
- **MODELO DE PROCESOS**
- **MODELO TECNOLÓGICO**

### **1.4.1 MODELOS DE DATOS**

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: los datos, las relaciones entre los datos y las restricciones que deben cumplirse sobre los datos. Adicionalmente contiene el conjunto de operaciones básicas para la realización de consultas (lecturas) y actualizaciones de datos.

Los modelos de datos pueden clasificarse en dos grandes grupos:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.

#### 1.4.1.1 MODELOS LÓGICOS BASADOS EN OBJETOS

Como su nombre lo indica son modelos basados en objetos: los dos más extendidos son el **modelo entidad-relación** y el **orientado a objetos**.

El modelo entidad-relación (E-R): Se basa en una percepción del mundo compuesta por objetos, llamados entidades y relaciones entre ellos. Las entidades se diferencian unas de otras a través de atributos.

El orientado a objetos: Al igual que el modelo E-R se basa en objetos, los cuales contienen valores y métodos, entendidos como órdenes que actúan sobre los valores. Los objetos se agrupan en clases, relacionándose mediante el envío de mensajes.

##### 1.4.1.1.1 MODELO ENTIDAD RELACIÓN

El modelo de Entidad-Relación fue propuesto por Peter Chen en 1976, en su escrito "The Entity-Relationship Model-Toward a Unified View of Data". Este modelo toma como punto de partida considerar la existencia de entidades, que representan objetos, personas, etc., sobre las que se quiere almacenar información relevante.

A las características necesarias para describir completamente a cada tipo de **entidad** se les denominará **atributo**. Posteriormente, las entidades y sus atributos se representan físicamente a través de **tablas** en las que los datos se almacenan en dos dimensiones. Las filas de la tabla contienen los atributos de cada una de las entidades, y las columnas el conjunto de atributos del mismo tipo de cada entidad. El grado de la tabla corresponderá al número de columnas de la tabla.

El principio fundamental en este modelo, que no puede obviarse de ninguna forma, es que hechos distintos deben almacenarse en objetos distintos.

Uno de los puntos fuertes de este modelo es que prevé que las entidades puedan mantener relaciones entre ellas. En primer lugar, es necesario definir la **clave de la entidad**. Las claves serán el atributo o conjunto de atributos, perteneciente al mismo tipo de entidad que hacen único el acceso a esa entidad u ocurrencia de la tabla, determinando de esta forma a una única entidad. La presencia de varios atributos que pueden funcionar como clave da lugar a la existencia de claves candidatas, y por otra parte se puede hablar de **claves simples** (formadas por un único atributo) y **claves múltiples**, compuestas o concatenadas (formadas por un conjunto de atributos). No hay que obviar tampoco el concepto de **clave ajena**, aquel atributo de una tabla que puede funcionar como clave en otra. La ocurrencia de entidad será, en este contexto, cada uno de los posibles valores reales que puede tomar la clave de una entidad.

Las relaciones entre tablas basadas en la conexión de éstas a través de las claves, pueden ofrecer diferentes **cardinalidades**, entendiéndose como tal el número de ocurrencias de una entidad que se relacionan con ocurrencias de la otra entidad. Pueden identificarse tres tipos:

- (1,1), donde una ocurrencia se relaciona con otra.
- (1,m), donde una ocurrencia puede relacionarse con varias; y
- (m,n), donde varias ocurrencias de una entidad pueden relacionarse con varias ocurrencias de la otra entidad.

El modelo de Chen es n-ario, lo cual quiere decir que las relaciones pueden establecerse entre una, dos o más entidades. Las entidades pueden ser de dos tipos:

- **Entidad regular:** aquella sobre la que se puede definir la clave primaria dentro de sus propios atributos.
- **Entidad débil:** aquellas que no puede utilizar sus propios atributos como clave, al estar asociada a otra entidad.

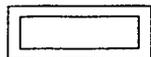
La definición del modelo conceptual con la técnica propuesta por Chen propone una secuencia de fases para la obtención del modelo:

1. Identificar las entidades dentro del sistema: Para esto, debe conocerse el funcionamiento del sistema en estudio, a través de estudios de usuarios, de necesidades de información, de tipos de información, etc. Como guía puede utilizarse para la definición de las entidades objetos reales, personas, actividades del sistema, objetos abstractos, etc.
2. Determinar las claves o identificadores de entidades: Señalar aquellos atributos que identifiquen inequívocamente cada ocurrencia de la entidad, y que no puedan ofrecer valores nulos.
3. Establecer las relaciones entre las entidades, describiendo el grado de las mismas: Estudiar las asociaciones entre las entidades, para definir su importancia dentro del contexto del sistema y obtener su cardinalidad.
4. Dibujar el modelo de datos: Representar gráficamente el modelo obtenido.
5. Identificar y describir los atributos de cada entidad: Señalar aquellas propiedades de la entidad de interés para el sistema.
6. Verificaciones: Eliminación de las relaciones redundantes y que puedan ser obtenidas a través de combinar otras asociaciones.

El modelo obtenido se representa mediante una notación gráfica especializada, a través de diagramas, cuyos principales símbolos son :

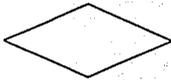


Entidad Regular.



Entidad Débil.





Relación.



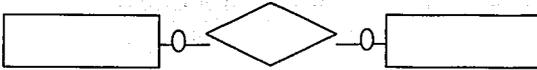
Relación 1:1



Relación 1:m



Relación m:n



Relación Opcional

TESIS CON  
FALLA DE ORIGEN

#### 1.4.1.1.2 MODELO ORIENTADO A OBJETOS

La orientación a objetos surge en Noruega en 1967 con un lenguaje llamado Simula 67, desarrollado por Krinsten Nygaard y Ole-Johan Dahl, en el Centro de Cálculo Noruego. Simula 67 introdujo por primera vez los conceptos de clases, corrutinas y subclases, pero este pasó inadvertido para gran parte de los programadores.

Uno de los problemas de inicio de los años setentas era que pocos sistemas lograban terminarse, pocos se terminaban con los requisitos iniciales y no todos los que se terminaban cumplían con los requerimientos según lo planificado. El problema consistía en

cómo adaptar el software a nuevos requerimientos imposibles de haber sido planificados inicialmente.

En los 70's científicos del centro de investigación en Palo Alto Xerox inventaron el lenguaje Small talk que dio respuesta al problema anterior (investigar no planificar). Small talk fue el primer lenguaje orientado a objetos puro es decir, únicamente utiliza clases y objetos.

En el inicio de los 90's se consolida la orientación a objetos como una de las mejores maneras para resolver problemas. Aumenta la necesidad de generar prototipos más rápidamente (concepto RAD Rapid Application Developments). Sin esperar a que los requerimientos iniciales estén totalmente precisos.

La orientación a objetos es una manera de pensar, es otra forma de resolver un problema, es lo más reciente en metodologías de desarrollo de software.

La orientación a objetos es un paradigma, es decir, es un modelo para aclarar algo o para explicarlo, este modelo pretende mejorar el diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo del desarrollo del software.

El paradigma orientación a objetos agrupa procesos e información como una unidad. La información en estas unidades es típicamente oculta, siendo revelada por una interface o grupo de comportamientos.

La orientación a objetos, toma como eje al objeto que tiene ciertos comportamientos y atributos, los mismos que determinan cómo actúan y funcionan.

La orientación a objetos está basada en tres métodos de organización: Entre un objeto y sus atributos ; Entre un objeto y sus componentes; Entre un objeto y su relación con otros objetos.

Aunque no existe una definición que englobe adecuadamente a este paradigma, mucha gente plantea que éste engloba al menos cuatro conceptos generales:

**Abstracción**  
**Encapsulamiento**  
**Herencia**  
**Polimorfismo**

Abstracción: Permite a los usuarios ignorar los detalles de la implementación y concentrarse en un nivel alto de vista de una entidad, es decir es la capacidad de representar, denotar y manejar información en un nivel alto que aquel que es inherente al lenguaje base del computador.

Encapsulamiento: Es la habilidad de agrupar partes relacionadas de información y procesos dentro de una unidad contenedora. El trabajo interno de un objeto está escondido u oculto, permitiendo al usuario concentrarse solamente en las características del objeto. Por ejemplo, los mecanismos internos de un teléfono están ocultos para el usuario por el cobertor, de tal manera que los alambres, switches y demás, están encapsulados y el usuario solamente se concentrará en usar el teléfono y no en las operaciones internas que éste ejecuta.

Herencia: Permite compartir funciones y datos entre clases, subclases y objetos. Es decir la herencia es la habilidad de una subclase para tomar las características de la clase o subclase en la cual esta es basada.

La clase general es referida como la base o clase padre. Una clase más específica es referida como la derivada o clase hija. La clase derivada obtiene los atributos y comportamientos de la clase base.

Polimorfismo: Habilita a objetos relacionados a responder en diferente forma, pero apropiadamente al mismo mensaje. Es la habilidad de aplicar métodos definidos para más de una clase. Los métodos pueden ser invocados sin conocer la clase actual.

### 1.4.1.2 MODELOS LÓGICOS BASADOS EN REGISTROS

Son modelos basados en registros, se usan para especificar la estructura lógica global de la base de datos, el más extendido es el **relacional**, mientras que los otros dos existentes, **jerárquico** y **de red**, se encuentran en retroceso.

El modelo relacional: Representa los datos y sus relaciones mediante tablas bidimensionales, que contienen datos tomados de los dominios correspondientes.

El modelo de red: Está formado por colecciones de registros, relacionados mediante punteros o ligas en grafos arbitrarios.

El modelo jerárquico: Es similar al de red, pero los registros se organizan como colecciones de árboles.

#### 1.4.1.2.1 MODELO RELACIONAL

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco) para relacionar registros de distintos ficheros. Codd demostró que estas bases de datos limitaban en gran medida los tipos de operaciones que los usuarios podían realizar sobre los datos. Además, estas bases de datos eran muy vulnerables a cambios en el entorno físico, si se añadían los controladores de un nuevo disco al sistema y los datos se movían de una localización física a otra, se requería una conversión de los ficheros de datos. Estos sistemas se basaban en el modelo de red y el modelo jerárquico, los dos modelos lógicos que constituyeron la primera generación de los DBMS.

El modelo relacional representa la segunda generación de los DBMS. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Pero detrás de esa simple

estructura hay un fundamento teórico importante del que carecen los DBMS de la primera generación, lo que constituye otro punto a su favor.

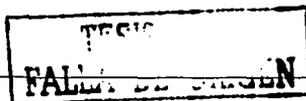
El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Al mismo tiempo, estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

La estructura fundamental del modelo relacional es precisamente esa "relación", es decir, una tabla bidimensional constituida por líneas (**tupla**) y columnas (**atributos**). Las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad. Por ejemplo, si en la base de datos se tienen que representar personas, se podrá definir una relación llamada "Personas", cuyos atributos describen las características de las personas, ver figura 1.2. Cada tupla de la relación "Personas" representará una persona concreta.

PERSONAS				
NOMBRE	APELLIDO	FEC NAC.	SEXO	EDO CIVIL
Beatriz	González	02/05/1969	F	Casada
Arturo	Mendoza	12/10/1992	M	Soltero
Daniel	Mendoza	16/04/1995	M	Soltero
Rosa María	Pacheco	17/04/1950	F	Casada
Mario	Mendoza	15/08/1945	M	Casado

Figura 1.2 Relación "Personas"

En realidad, siendo rigurosos, una relación es sólo la definición de la estructura de la tabla, es decir su nombre y la lista de los atributos que la componen. Cuando se prueba con las tuplas, se habla de "instancia de relación". Por eso, la figura 1.2 representa una instancia de la relación persona. Una representación de la definición de esa relación podría ser la siguiente:



*Personas (nombre, apellido, fec\_nac, sexo, edo\_civil)*

Las tuplas en una relación son un conjunto en el sentido matemático del término, es decir, una colección no ordenada de elementos diferentes. Para distinguir una tupla de otra, se recurre al concepto de "llave primaria", o sea a un conjunto de atributos que permiten identificar unívocamente una tupla en una relación. Naturalmente, en una relación puede haber más combinaciones de atributos que permitan identificar unívocamente una tupla ("llaves candidatas"), pero entre éstas se elegirá una sola para utilizarla como llave primaria.

Los atributos de la llave primaria no pueden asumir el valor nulo. Esta propiedad de las relaciones y de sus llaves primarias está bajo el nombre de **integridad de las entidades** (entity integrity).

Cada atributo de una relación se caracteriza por un **nombre** y por un **dominio**. El dominio indica qué valores pueden ser asumidos por una columna de la relación. A menudo un dominio se define a través de la declaración de un tipo para el atributo, por ejemplo, diciendo que es una cadena de diez caracteres, pero también es posible definir dominios más complejos y precisos. Por ejemplo, para el atributo "sexo" de nuestra relación "Personas" se puede definir un dominio en el que los únicos valores válidos sean 'M' y 'F'; o bien, para el atributo "fecha\_nacimiento" se puede definir un dominio en el que se consideren válidas sólo las fechas de nacimiento después del uno de enero de 1940. El DBMS se ocupará de controlar que en los atributos de las relaciones se incluyan sólo los valores permitidos por sus dominios.

Una característica fundamental de los dominios de una base de datos relacional es que sean "atómicos", es decir que los valores contenidos en las columnas no se puedan separar en valores de dominios más simples. Por ejemplo, si una característica de las personas en nuestra base de datos fuese la de tener uno o más hijos, no sería posible escribir la relación Personas de la siguiente manera:

*Personas (nombre, apellido, fec\_nac, sexo, edo\_civil, hijos)*

En efecto, el atributo hijos es un atributo no-atómico, bien porque una persona puede tener más de un hijo o porque cada hijo tendrá diferentes características que lo describen. Para representar estas entidades en una base de datos relacional hay que definir dos relaciones:

*Personas (\*número\_persona, nombre, apellido, fec\_nac, sexo, edo\_civil)*

*Hijos(\*número\_persona, \*nombre\_apellido, edad, sexo)*

En las relaciones precedentes, los asteriscos (\*) indican los atributos que componen sus llaves primarias. Como puede observarse, se agrego en la relación "Personas" el atributo número\_persona, a través del cual se asignará a cada persona un identificador numérico unívoco que se usará como llave primaria. Estas relaciones contienen sólo atributos atómicos. Si una persona tiene más de un hijo, éstos se representarán en tuplas diferentes de la relación "Hijos". Las diferentes características de los hijos las representan los atributos de la relación Hijos.

La unión entre las dos relaciones está constituida por el atributo número\_persona que aparece en ambas relaciones y que permite asignar a cada tupla de la relación "Hijos" una tupla concreta de la relación "Personas". En otras palabras, se dice que el atributo número\_persona de la relación "Hijos" es una **llave externa** (foreign key) hacia la relación "Personas".

Una llave externa es una combinación de atributos de una relación, la cual es a su vez, una llave primaria para otra relación. Una característica fundamental de los valores presentes en una llave externa radica en que tienen que corresponder a valores existentes en la llave primaria de la relación a la que se refieren. Siguiendo con nuestro ejemplo, esto significa que no puede existir en la relación "Hijos" una tupla con un valor del atributo número\_persona sin que también en la relación Personas exista una tupla con el mismo valor para su llave primaria, a esta propiedad se le conoce con el nombre de **integridad referencial** (referential integrity).

Una de las grandes ventajas con las que cuenta el modelo relacional, es el uso de una "**álgebra relacional**", a través de la cual se realizan todas las manipulaciones posibles sobre las relaciones, utilizando para esto tan solo cinco operadores: **RESTRICT**, **PROJECT**, **TIMES**, **UNION** y **MINUS**. Por comodidad, se han definido también tres operadores adicionales, los cuales se pueden obtener aplicando los cinco fundamentales, siendo estos: **JOIN**, **INTERSECT** y **DIVIDE**.

Los operadores relacionales reciben como argumento una relación o un conjunto de relaciones y restituyen una única relación como resultado, veamos brevemente estos ocho operadores:

- RESTRICT:** Restituye una relación que contiene un subconjunto de las tuplas de la relación a la que se aplica. Los atributos se quedan como estaban.
  
- PROJECT:** Restituye una relación con un subconjunto de los atributos de la relación a la que viene aplicado. Las tuplas de la relación resultado se componen de las tuplas de la relación original, de manera que siguen siendo un conjunto en sentido matemático.
  
- TIMES:** Se aplica a dos relaciones y efectúa el producto cartesiano de las tuplas. Cada tupla de la primera relación está concatenada con cada tupla de la segunda.
  
- UNION:** Aplicando este operador a dos relaciones compatibles, se obtiene una que contiene las tuplas de ambas relaciones. Dos relaciones son compatibles si tienen el mismo número de atributos y los atributos correspondientes en las dos relaciones tienen el mismo dominio.
  
- MINUS:** Aplicado a dos relaciones compatibles restituye una tercera que contiene las tuplas que se encuentran sólo en la primera relación.
  
- JOIN:** Se concatenan las tuplas de dos relaciones de acuerdo con el valor de un conjunto de sus atributos.

**INTERSECT:** Aplicado a dos relaciones compatibles restituye una relación que contiene las tuplas que existen en ambas.

**DIVIDE:** Aplicado a dos relaciones que tengan atributos comunes, restituye una tercera que contiene todas las tuplas de la primera relación que se puede hacer que correspondan con todos los valores de la segunda relación.

Considerando las tablas ilustradas en las figuras 1.3 y 1.4, a título de ejemplo, se presentan los resultados de la aplicación de algunos operadores relacionales a las relaciones "Personas" e "Hijos".

PERSONAS					
NÚMERO_PERSONA	NOMBRE	APELLIDO	FEC_NAC.	SEXO	EDO_CIVIL
1	Beatriz	González	02/05/1969	F	Casada
2	Arturo	Mendoza	12/10/1992	M	Soltero
3	Daniel	Mendoza	16/04/1995	M	Soltero
4	Rosa María	Pacheco	17/04/1950	F	Casada
5	Mario	Mendoza	15/08/1945	M	Casado

Figura 1.3 Relación "Personas"

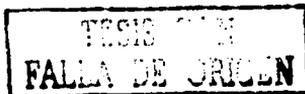
HIJOS			
NÚMERO_PERSONA	NOMBRE_APELLIDO	EDAD	SEXO
5	Araceli Mendoza	34	F
5	Wendy Mendoza	32	F
5	M.Alberto Mendoza	30	M
5	Julieta Mendoza	20	F

Figura 1.4 Relación "Hijos"

La figura 1.5 muestra el resultado que se obtiene al aplicar el operador Restrict sobre la tabla de "Personas".

**RESTRICT (Personas)**

Sexo = 'M'



NÚMERO_PERSONA	NOMBRE	APELLIDO	FEC_NAC.	SEXO	EDO_CIVIL
2	Arturo	Mendoza	12/10/1992	M	Soltero
3	Daniel	Mendoza	16/04/1995	M	Soltero
5	Mario	Mendoza	15/08/1945	M	Casado

Figura 1.5 Resultado operador Restrict

La figura 1.6 muestra el resultado que se obtiene al aplicar el operador Project sobre la tabla de "Personas".

**PROJECT** sexo (Personas)

SEXO
M
F

Figura 1.6 Resultado operador Project

La figura 1.7 muestra el resultado que se obtiene al aplicar el operador Time sobre las tablas de "Personas" e "Hijos".

**TIMES** (Personas) , (Hijos)

NÚMERO_PERSONA	NOMBRE	APELLIDO	FEC_NAC.	SEXO	EDO_CIVIL	NOMBRE	EDAD	SEXO
5	Mario	Mendoza	15/08/1945	M	Casado	Araceli Mendoza	34	F
5	Mario	Mendoza	15/08/1945	M	Casado	Wendy Mendoza	32	F
5	Mario	Mendoza	15/08/1945	M	Casado	M Alberto Mendoza	30	M
5	Mario	Mendoza	15/08/1945	M	Casado	Julieta Mendoza	20	F

Figura 1.7 Resultado operador Times

Las bases de datos relacionales efectúan todas las operaciones en las tablas usando el álgebra relacional, aunque normalmente no le permiten al usuario usarla. El usuario interactúa con la base de datos a través de una interfaz diferente, por ejemplo el lenguaje SQL, las instrucciones SQL vienen descompuestas por el DBMS en una serie de operaciones relacionales.

El modelo conceptual de datos obtenido mediante la técnica de modelado relacional, será refinado y convertido en un modelo lógico relacional, utilizando la normalización, lo que ofrecerá como resultado el conjunto de tablas a implementar en la base de datos.

## **NORMALIZACIÓN**

Su finalidad es reducir las inconsistencias y redundancias de los datos, facilitar el mantenimiento y evitar las anomalías en las manipulaciones de datos. El objetivo será obtener un modelo lógico normalizado que represente las entidades normalizadas y las interrelaciones existentes entre ellas. Para lo cual, se toma como punto teórico de partida el concepto de dependencia funcional, que dice:

"Un atributo B depende funcionalmente de otro atributo A, de la misma entidad si a cada valor de A le corresponde sólo un valor de B."

El procedimiento de normalización consiste en someter a las tablas que representan entidades a un análisis formal para ver si cumplen, o no, las restricciones necesarias que aseguren evitar los problemas citados con anterioridad. A mayor nivel de normalización, mayor calidad en la organización de los datos y menor peligro para la integridad de éstos.

Las formas normales se clasifican en:

- **Primera Forma Normal (1FN)** : Se dice que una relación esta en 1FN si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos.
- **Segunda Forma Normal (2FN)**: Una relación pertenece a la 2FN si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal.

- Tercera Forma Normal (3FN): Una relación es 3FN si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave.
- Cuarta Forma Normal (4FN): En esta forma se prohíbe la independencia de relaciones uno a muchos entre columnas clave primaria y columna no-clave.

#### 1.4.1.2.2 MODELO DE RED

El modelo de red como su nombre lo indica, está formado por una colección de registros, los cuales están conectados entre sí por medio de enlaces. El registro es similar a una entidad como las empleadas en el modelo entidad-relación.

Un **registro** es una colección de campos (atributos), cada uno de los cuales contiene solamente almacenado un solo valor, el **enlace** es la asociación entre dos registros exclusivamente, así que podemos verla como una relación estrictamente binaria.

Este modelo se basa en representaciones entre registros por medio de ligas, existen relaciones en las que participan solo dos entidades (binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación.

La representación gráfica del modelo consta de dos componentes básicos:

**Celdas:** representan a los campos del registro.

**Líneas:** representan a los enlaces entre los registros.

El modelo representativo del enfoque de red, fue desarrollado en 1971 por un grupo conocido como CODASYL (Conference on Data System Languages), Data Base Task Group (DBTG), de ahí el nombre; este grupo es el que desarrolló los estándares para COBOL, el modelo CODASYL ha evolucionado durante los últimos años y existen diversos productos DBMS orientados a transacciones, sin embargo hoy día, estos productos están de salida ya que este modelo es complejo y no cohesivo.

Una estructura de datos en red, es muy similar a una estructura jerárquica, de hecho no es más que un superconjunto de ésta. Al igual que en la estructura jerárquica, cada nodo puede tener varios hijos pero, a diferencia de ésta, también puede tener varios padres, la figura 1.8 ilustra dicha situación:

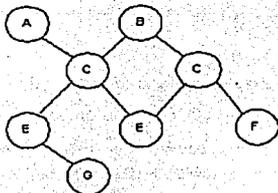


Figura 1.8 Estructura modelo de red

El concepto básico en el enfoque de red es el conjunto ('set'), definido por el comité CODASYL. Un conjunto está constituido por dos tipos de registros que mantienen una relación de muchos a muchos. Para conseguir representar este tipo de relación es necesario que los dos tipos de registros estén interconectados por medio de un registro conector llamado conjunto conector. Los conjuntos poseen las siguientes características:

- El registro padre se denomina propietario del conjunto, mientras que el registro hijo se denomina miembro.
- Un conjunto está formado en un solo registro propietario y uno o más registros miembros.
- Una ocurrencia de conjuntos es una colección de registros, uno de ellos es el propietario y los otros los miembros.
- Todos los registros propietarios de ocurrencias del mismo tipo de conjunto deben ser del mismo tipo de registro.
- El tipo de registro propietario de un tipo de conjunto debe ser distinto de los tipos de los registros miembro.

TESIS CON  
FALLA DE ORIGEN

- Sólo se permite que un registro miembro aparezca una vez en las ocurrencias de conjuntos del mismo tipo.
- Un registro miembro puede asociarse con más de un propietario, es decir, puede pertenecer al mismo tiempo a dos o más tipos de conjuntos distintos.
- Se pueden definir niveles múltiples de jerarquías donde un tipo de registro puede ser miembro en un conjunto y al mismo tiempo propietario en otro conjunto diferente.

### 1.4.1.2.3 MODELO JERÁRQUICO

Un modelo jerárquico consiste en una colección de registros que se conectan entre sí por medio de **enlaces**. Los registros son similares a los expuestos en el modelo de red. Cada registro es una colección de campos (atributos), que contienen un solo valor cada uno de ellos. Un enlace es una asociación o unión entre dos registros exclusivamente.

Por tanto, este concepto es similar al de enlace para modelos de red. La principal diferencia es que en el modelo de red los registros se organizan en forma de un grafo arbitrario, mientras que en modelo de estructura de árbol los registros se organizan en forma de un árbol con raíz.

Un modelo jerárquico utiliza jerarquías o árboles para la representación lógica de los datos. Los archivos son organizados en **jerarquías**, y normalmente cada uno de ellos se corresponde con una de las entidades de la base de datos. Los árboles jerárquicos se representan de forma invertida, con la raíz hacia arriba y las hojas hacia abajo, la figura 1.9 muestra esta situación:

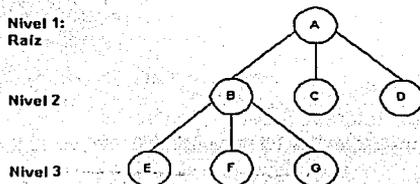


Figura 1.9 Estructura del modelo jerárquico

Un modelo jerárquico recorre los distintos nodos de un árbol en un preorden que requiere tres pasos:

1. Visitar la raíz.
2. Visitar el hijo más a la izquierda, si lo hubiera, que no haya sido visitado.
3. Si todos los descendientes del segmento considerado se han visitado, volver a su padre e ir al punto 1.

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que los describen, las relaciones entre entidades están representadas por las ramas.

Las principales características del modelo jerárquico, son:

1. Los segmentos de un archivo jerárquico están dispuestos en forma de árbol.
2. Los segmentos están enlazados mediante relaciones uno a muchos.
3. Cada nodo consta de uno o más campos.
4. Cada ocurrencia de un registro padre puede tener distinto número de ocurrencias de registros hijos.
5. Cuando se elimina un registro padre se deben eliminar todos los registros hijos (integridad de los datos).
6. Todo registro hijo debe tener un único registro padre excepto la raíz.

#### 1.4.2 MODELO DE PROCESOS

El objetivo del modelo de procesos es describir la estructura o composición de los procesos, de una manera jerárquica y de esta manera comprender su funcionamiento y relación con su medio ambiente, para realizar el modelo de procesos podemos apoyarnos en el **Método Estructurado de Datos** o en el **Método Orientado a Objetos**.

### **1.4.2.1 MÉTODO ESTRUCTURADO DE DATOS**

Es una actividad de construcción de modelos, que refleja el flujo y contenido de la información (datos y control), en donde el sistema se parte funcionalmente y, según los distintos comportamientos se establece la esencia de lo que se debe construir.

El método estructurado se puede resumir en las siguientes actividades:

- Definición del Diagrama de Flujo de Datos.
- Definición del Diagrama de Estructura.

#### **1.4.2.1.1 DIAGRAMA DE FLUJO DE DATOS**

Es una herramienta de modelado que permite representar el sistema como una red de procesos funcionales conectados unos con otros.

El Diagrama de Flujo de Datos (DFD) es una de las herramientas de modelado más utilizada, particularmente para sistemas en los cuales las funciones son de principal importancia y más complejas que los datos que manipula el sistema.

Esta es una herramienta gráfica que describe el flujo de los datos a través del sistema. El diagrama muestra la transformación de datos de entrada en datos de salida por medio de los procesos.

Los DFD muestran:

- De dónde vienen y hacia dónde van los datos cuando salen del sistema (Entidades Externas)
- Cuáles son los datos que llegan y salen (Flujo de Datos)

- Dónde se almacenan los datos (Depósito de Datos)
- Qué procesos transforman los datos
- Las interacciones entre depósitos de datos y procesos.

#### Recomendaciones de Yourdon para la representación de Flujos de Datos.

- Todos los componentes del gráfico participan en la actividad que se está modelando, por lo tanto, en el DFD, cada uno de sus componentes debe estar conectado por lo menos con otro componente.
- Cada proceso produce la información que el sistema necesita, por lo tanto, en el DFD cada proceso debe usar, al menos, un flujo de datos de entrada y producir, al menos, un flujo de salida.
- El modelo del proceso identifica la fuente de todos los datos utilizados en el sistema, por lo tanto, cada elemento de datos en un flujo de salida debe corresponder o derivarse de un elemento de datos de uno de los flujos de entrada del proceso.
- Cada flujo de datos soporta, al menos, una actividad del sistema, por lo tanto, en el DFD cada flujo de datos, desde o hacia un almacenamiento de datos o una entidad externa, debe conectarse a un proceso.

#### Convenciones para la denominación de componentes de un DFD:

- **A un proceso:** Con una frase compuesta de verbo + complemento, que identifique la actividad que desarrolla.
- **A un flujo de datos:** Con un sustantivo en singular, que identifique la información comunicada.

- **A un almacenamiento de datos:** Con un sustantivo en plural que identifique la información recogida.
- **A una entidad externa:** Con un sustantivo en singular, que describa quién o qué interactúa con el sistema.

El DFD esta compuesto de los siguientes símbolos:



Representa una función dentro del programa que muestra una transformación de datos.



Flujo de datos (Siempre de izquierda a derecha).



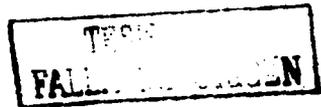
Unidad física o lógica que interviene en el sistema. (Utilidad un archivo).



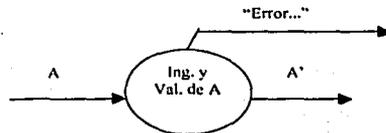
Dispositivo físico que interviene en el sistema. (Distingue cuando un dato sale por impresora).

Al definir un diagrama de flujo de datos es necesario documentarlo, es decir, explicar algunos conceptos como son:

1. Nombre de las transformaciones.
2. Entrada a la transformación.
3. Salida de la transformación.
4. Descripción de la transformación.



*Ejemplo:*



Nombre de las transformaciones: Ingreso y Validación de A  
 Entrada a la transformación: A  
 Salida de la transformación: A validada (A'), mensaje de error: "Error...".  
 Descripción de la transformación: Permite el ingreso de la variable numérica A y valida que su rango sea (0, 100)

#### 1.4.2.1.2 DIAGRAMA DE ESTRUCTURA

Un Diagrama de Estructura (D.E.) permite definir la organización de un programa en términos jerárquicos, definiendo las relaciones entre los módulos e identificando las interfaces y los métodos de control.

Un diagrama de estructura es un grafo que tiene los siguientes elementos.



Representa un módulo del programa (en el se ejecutan los procesos determinados).



Representa la dependencia jerárquica entre los módulos (flecha que va entre módulos o módulos y distintos periféricos).



Representa datos que fluyen a través de los módulos de diseño estructurado.



Representa datos de control (en general se omiten)



Representa una ejecución alternativa



Representa una ejecución de repetición en los módulos.



Dispositivo físico que interviene en el sistema (Distingue cuando un dato sale por impresora, entra de teclado, etc.)



Unidad física o lógica que intervienen en el sistema. (Utilizar un archivo)

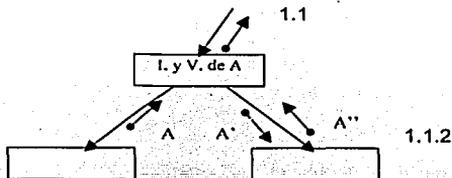
Cada módulo del diagrama de estructura representa un conjunto de pasos o instrucciones que se ejecutan en ese lugar del programa dando origen obviamente al algoritmo del módulo.

Para documentar un D.E. se debe describir cada uno de los módulos que pertenecen al citado D.E., considerando los siguientes elementos:

1. Nombre del módulo.
2. Objetivo del módulo.
3. Entradas del módulo.
4. Salidas del módulo.
5. Algoritmo del módulo.
6. Descripción de variables del módulo.

TESIS CON  
FALLA DE ORIGEN

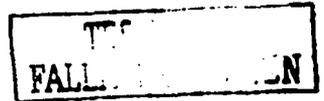
Ejemplo:



**Nombre del módulo:** Ingreso, validación y verificación de A  
**Objetivo del módulo:** Controla la ejecución de los módulos 1.1 y 1.1.2  
**Entradas del módulo:** A  
**Salidas del módulo:** A validada (A') y verificada (A'')  
**Algoritmo del módulo:** Begin

```

While (Cond_1) do
  Begin
    Llamar Mod_1.1.1
    If (Cond_2) Then
      Llamar Mod_1.1.2
    End
  End
End
  
```



**Descripción de variables del módulo (en forma tabular)**

Nombre De Variable	Descripción	Longitud	Tipo	Rango de Validación

**Nombre de variable:** Se escribe el nombre que ha sido dado para la variable en el algoritmo.

**Descripción:** Se define brevemente la variable indicando el objetivo o el uso que se le da.

**Longitud:** Se define el largo total de la variable.

**Tipo:** Se define el tipo de dato de la variable.

**R. Validación:** Describe los posibles valores que pueden tomar las variables en descripción.

### **1.4.2.2 MÉTODO ORIENTADO A OBJETOS**

Los primeros intentos de describir un método orientado a los objetos surge hasta la década de los ochenta. Los precursores de este método fueron Abbott y Booch, los cuales establecen que el diseño orientado a objetos debe comenzar con una descripción en lenguaje natural de la estrategia de solución, a través del cual se implementará el software para un problema real.

La metodología orientada a objetos consiste en:

- Saber el espacio del problema.
- Realizar una abstracción.
- Crear los objetos.
- Instanciarlos.
- Dejarlos vivir.

Para lograr lo anterior, la metodología orientada a objetos se apoya básicamente en 3 clases de modelos para describir un problema.

- Modelo de objeto:
- Modelo dinámico.
- Modelo Funcional.

#### **1.4.2.2.1 MODELO DE OBJETO**

El propósito del modelado de objeto es describir los objetos, un objeto es una abstracción con límites bien definidos y con significado a efectos del problema que se tenga entre manos, se representa a través de un gráfico cuyos nodos son clases de objetos y cuyos arcos son relaciones entre clases, el cual representa los aspectos estáticos y estructurales de datos del sistema.

Los conceptos más importantes de este modelo son:

**Clase:** Una clase de objeto describe un grupo de objetos con atributos similares, con relaciones comunes con otros objetos y con una semántica común.

**Atributos:** Es un valor de un dato que está almacenado en los objetos de una clase, los atributos deberán ser valores puros de datos y no objetos, el nombre de cada atributo puede ir seguido por detalles como el tipo y el valor por omisión.

**Operaciones:** Es una función o transformación que puede ser aplicada por los objetos de una clase, todos los objetos de una clase comparten las mismas operaciones y una misma operación puede aplicarse a clases distintas, cada operación es polimórfica, es decir, una misma operación adopta distintas formas en distintas clases.

**Método:** Es la implementación de una operación para una clase, por ejemplo, la clase archivo puede tener la operación *imprimir* y se puede implementar de distintos métodos para hacerlo (ASCII, binario, etc.), todos los métodos realizan la misma tarea lógica.

**Enlaces y asociaciones:** Son los medios para establecer relaciones entre objetos y clases, un enlace es una conexión física o conceptual entre instancias de objetos, matemáticamente se define como un ente ordenado de instancias de objetos, un enlace es una instancia de una asociación, ésta describe un grupo de enlaces con estructura y semántica comunes, las asociaciones describen un conjunto de enlaces potenciales, del mismo modo que las clases describen un conjunto de objetos potenciales, y suelen implementarse en los lenguajes de programación como punteros que van de un objeto a otro. La notación para las asociaciones es una línea entre clases, y los enlaces es una línea entre objetos, el nombre de la asociación se pone en  *cursiva*, la figura 1.10 ilustra lo descrito anteriormente:

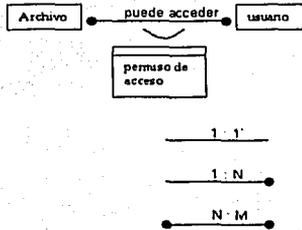


Figura 1.10 Enlace y Asociaciones

**Multiplicidad:** Limita el número de objetos relacionados, los diagramas de objetos lo indican mediante símbolos al final de la línea de asociación, subestimar la multiplicidad puede evitar la flexibilidad de una aplicación.

**Rol:** Es un nombre que identifica en forma única un extremo de la asociación, el uso de nombres de rol proporciona una forma de recorrer las asociaciones desde un objeto de un extremo sin mencionar explícitamente la asociación, los roles pueden aparecer como sustantivos en la descripción del problema, la figura 1.11 muestra esta situación:

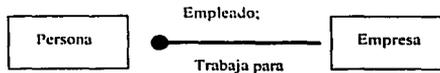


Figura 1.11 Empleo del Rol entre Asociaciones

**Clasificación:** Normalmente los objetos del lado *muchos* en una asociación no tienen un orden explícito y se pueden considerar como un conjunto, la clasificación es una parte inherente de la asociación, si es un conjunto ordenado de objetos se escribe *ordenado* o *clasificado* al lado del símbolo de multiplicidad.

**Herencia:** Ha llegado a ser un sinónimo de reutilización del código dentro de la programación orientada a objeto, frecuentemente hay un código que está disponible de trabajos anteriores (biblioteca), la aplicación más importante es la simplificación conceptual que proviene de reducir el número de características independientes dentro del sistema.

**Módulo:** Es una construcción lógica para agrupar clases, el nombre del módulo debe especificarse en la parte superior de la hoja, los módulos nos permiten descomponer al modelo en segmentos manejables.

**Hojas:** Una hoja es el mecanismo para descomponer un modelo de objetos grande, en un conjunto de páginas; una hoja es una notación cómoda, no una estructura lógica.

**Clases abstractas:** Es una clase que no tiene instancias directas, en cambio una clase concreta puede unir instancias discretas, una clase concreta puede formar subclases abstractas.

**Claves candidatas:** Es un conjunto mínimo de atributos que definen de manera unívoca un objeto. Una clase puede tener más de una clave candidata, cada una de las cuales tendrá distintas combinaciones y número de atributos.

#### **1.4.2.2.2 MODELO DINÁMICO**

Los aspectos del sistema que están relacionados con los tiempos y con los cambios constituyen lo modelos dinámicos, los conceptos más importantes del modelado dinámico son:

**Estados:** Son los valores de los atributos y de los enlaces mantenidos por un objeto, un diagrama de estado es una red de estados y recursos, el modelo dinámico consta de múltiples diagramas de estado, cada uno de ellos para cada clase que contenga un comportamiento dinámico importante y muestre la actividad del sistema.

**Suceso:** Es algo que transcurre durante un período de tiempo, un suceso es una transmisión de información de dirección única entre un objeto y otro. Todo suceso se agrupa en clases a los que se les da un nombre para una comodidad de estructura (jerárquica) y de comportamiento, todo suceso aporta información de un objeto a otro, los valores de los datos aportados son sus atributos.

**Escenario:** Es una secuencia de sucesos que se producen durante una ejecución completa de un sistema, el ambiente puede incluir a todos los sucesos o solo a aquellos que afecten algunos objetos del sistema, el paso siguiente a la escritura de un escenario consiste en identificar a los objetos emisores y receptores de cada suceso. Los objetos que intercambian sucesos se pueden mostrar en un ambiente mejorado llamado *diagrama de segmentos de trazos de sucesos*.

**Actividad:** Es una operación cuya realización requiere un cierto tiempo, toda actividad esta asociada a un estado; entre las actividades se encuentran las operaciones continuas, un estado puede controlar una actividad continua, la cual persiste hasta que se produce un suceso que le da fin, produciendo una transición que sale de ese estado.

**Acción:** Es una operación instantánea que va asociada a un suceso, las acciones también pueden representar operaciones internas de control, tales como dar un valor a un atributo o generar otros sucesos, estas acciones son mecanismos para estructurar el control dentro de una implementación.

#### **1.4.2.2.3 MODELO FUNCIONAL**

Muestra la forma en la que se derivan los valores producidos en un cálculo a partir de los valores introducidos, sin tener en cuenta el orden en que se calculan, consta de múltiples Diagramas de Flujo de Datos (DFD) que muestran el flujo de valores desde las entradas externas a través de las operaciones y almacenes hasta las salidas externas; los conceptos más importantes del modelo funcional son:

**DFD:** Muestra las relaciones funcionales entre los valores calculados por un sistema incluyendo los valores introducidos, los obtenidos y los almacenes internos de datos. Un DFD contiene procesos que transforman datos, flujos de datos, objetos actores que producen y consumen datos y almacenes de datos que los almacenan en forma pasiva.

**Procesos:** Transforma valores de datos, los de bajo nivel son funciones puras sin efectos laterales, un gráfico completo de un flujo de datos es un proceso de alto nivel y pueden tener efectos laterales como almacenes de datos de objetos externos.

**Flujo de datos:** Conecta la salida de un objeto o proceso con la entrada de otro objeto o proceso, se dibuja como flechas entre el productor y el consumidor. Las flechas se rotulan con una descripción de los datos.

**Actores:** Es un objeto activo que controla el gráfico de flujo de datos produciendo o consumiendo valores, están asociados a las entradas y a las salidas del gráfico de flujo de datos, se denominan también terminador.

**Almacén de datos:** Es un objeto pasivo dentro de un DFD, a diferencia de los actores no genera ninguna operación por sí mismo, sino que almacena y accede a datos.

**Flujo de control:** Un diagrama de flujo de control muestra todas las posibles vías de computación para los valores, no muestra cuales son las vías que se ejecutan ni en que orden, se muestran con una línea discontinua que va de un proceso que produce un valor hasta el que se está controlando.

### 1.4.3 MODELO TECNOLÓGICO

El modelo tecnológico es utilizado para la instrumentación del proyecto a desarrollar, normalmente se inicia en paralelo durante la realización de los modelos de datos y procesos, es aquí donde deberán indicarse los recursos informáticos necesarios para la implementación del proyecto tomándose básicamente dos aspectos:

- **SopORTE lógico:** Enumeración de las necesidades de software.
- **SopORTE físico:** Enumeración del equipamiento que se utilizará.

Dentro del **soporte lógico**, se deberá considerar la funcionalidad del sistema que se implementará, definiendo el objetivo y los requisitos indispensables para su realización, alcance del mismo, lenguaje a utilizar en su desarrollo, soporte lógico de datos, sistema operativo sobre el cual se implementará, estimación de volúmenes de información. Definir los subsistemas principales y las líneas de flujo de información. Establecer los límites de información entre los que se implementará el sistema y el entorno en el que éste va a funcionar. Definir todos los productores de información internos y externos utilizados por el sistema, todos los consumidores de información y las entidades que se comunican con el sistema.

Dentro del **soporte físico**, se deberán incluir los productos de hardware a adquirir, así como la capacitación para el uso de los mismos, si la misma fuera necesaria. Deberá indicarse la distribución física del equipamiento para las distintas áreas, los responsables de la administración de los servidores a lo largo del ciclo de vida del proyecto. Se indicará, también si existe la necesidad de ampliar el cableado o realizar algún cambio de topología de red.



## **2.1 OBJETIVO DEL SISTEMA**

Diseño, desarrollo e implementación de un sistema de control y seguimiento de pagos, para facilitar el procesamiento y la manipulación de datos provenientes de las pólizas contables que son entregadas al Área de Central de Pagos de las Instituciones Bancarias.

A través de la integración de los datos en el Área de Central de Pagos, se podrá dar seguimiento a la situación que guarda cualquier solicitud de pago, ya sea de un proveedor externo o de un empleado, con lo que se notará una considerable disminución de tiempo para la entrega de sus pagos. Así mismo, se contará con el detalle contable de todos los pagos efectuados en un sistema propio del departamento de pagos, con lo que se evitará la saturación de información en el Sistema Central de las Instituciones Bancarias.

## **2.2 DEFINICIÓN DEL PROBLEMA**

El Área de Central de Pagos de una Institución Bancaria, es la encargada de concentrar todos los requerimientos de pagos provenientes de los distintos gastos que son originados por la operación diaria de sus instalaciones, tales como: rentas de inmuebles, luz, agua, teléfono, estacionamiento, mantenimiento de elevadores, pagos de impuestos federales, mantenimiento de inmuebles adjudicados, compra y mantenimiento de equipos de cómputo, pago por el uso de licencias de software, etc.; por otra parte, concentra los requerimientos de pagos provenientes de los distintos gastos originados por sus empleados, tales como: gastos por el uso de servicio médico, pasajes, anticipos de viaje y viáticos, etc.

Para lograr el objetivo que es pagar el servicio otorgado por un proveedor externo o pagar los gastos originados por sus empleados, el Área de Central de Pagos tiene que interactuar con los departamentos que originan gasto, con el Área de Ventanilla, con el Área de Emisión de Pagos y con el Sistema Central de Contabilidad.

En esta interacción existen varios problemas que afectan la operación diaria del Área de Central de Pagos, lo que se traduce en una mala calidad del servicio, en una entrega extemporánea de pagos, en una mala administración y en un casi nulo seguimiento de los pagos otorgados. Dentro de los problemas detectados, podemos mencionar:

#### Departamentos:

- La elaboración y el llenado de las pólizas contables se realizan en Excel.
- No todos los departamentos cuentan con los formatos de pólizas contables, lo que provoca que algunos usuarios elaboren sus propios formatos, los cuales difieren a los establecidos por el Área de Central de Pagos.
- En algunos casos los usuarios no conocen perfectamente cual es el procedimiento y que cuentas contables deben afectar para el correcto llenado de las pólizas.
- Dentro de la estructura de personal de los departamentos, existen varias personas cuya única función es la elaboración de pólizas, provocando con esto la existencia de pequeñas áreas de contabilidad.
- No existen conceptos de pagos, el usuario debe saber a la perfección el significado de cada cuenta contable para poder realizar su correcta afectación.
- Los usuarios deben realizar manualmente el cálculo de impuestos tales como IVA, ISR (Impuesto Sobre la Renta), Retención de IVA y DESCUENTOS, provocando en algunos casos errores en los cálculos realizados.
- Existen departamentos que generan bastantes pólizas, traduciéndose esto en saturación de la Ventanilla cuando llevan sus pólizas a pagos.

#### Ventanilla:

- En la recepción de pólizas, nuevamente se vuelve a capturar en Excel los datos del proveedor, el concepto de pago y el importe a pagar.
- El llenado de contra-recibos que se entrega como contraseña de recepción de pólizas se realiza a mano.

- La fecha de pago de pólizas se asigna manualmente y se registra en Excel.
- Al momento de recibir las pólizas de departamentos con gran volumen, se ocasiona un cuello de botella en la Ventanilla, originando demasiado tiempo de espera para el resto de usuarios.
- Después del cierre de la Ventanilla, se clasifican manualmente las pólizas que serán entregadas a cada responsable para su revisión, esta labor se realiza a través de las cuentas contables afectadas por cada póliza, lo que ocasiona pérdida de tiempo.
- Si un proveedor quiere saber cuándo y qué pólizas le serán pagadas, el personal de Ventanilla pregunta telefónicamente a los responsables de revisión la situación de pago de esas pólizas, ocasionando con esto pérdida de tiempo en la recepción de Ventanilla.
- Cuando el Área de Emisión de Pagos entrega los cheques elaborados, el personal de Ventanilla transcribe en una libreta: el número de cheque, el nombre del beneficiario, el importe y la fecha de pago, para que al momento de ser entregados el proveedor firme de recibido.

#### Emisión de pagos:

- No existe un registro magnético de los contra-recibos, provocando con esto la revisión manual de pólizas y la verificación de importes vía calculadora.
- Se lleva un control a través de Excel, para saber que pólizas se están autorizando para su pago.
- Debido a la pérdida de tiempo producto de la duplicidad de captura y control a través de Excel, en algunas ocasiones, la solicitud de fondos al área de Tesorería no se hace a tiempo, retrasando con esto los pagos previstos para una fecha dada.
- No se cuenta con reportes de control que indique los cheques emitidos.
- No se tienen reportes para indicar qué pólizas se han pagado y cuáles están pendientes.
- No se cuenta con un histórico de pagos por proveedor.

- La generación de cheques se realiza a través de un aplicativo que solo permite capturar el nombre del beneficiario y el importe para la impresión de cheques.
- Para realizar el arqueo de cheques se tiene que recurrir a la libreta de control de cheques emitidos, para saber cuáles fueron entregados y cuáles no han sido entregados.
- A pesar de que se cuenta con un aplicativo de banca electrónica, solamente se recurre a éste para los pagos urgentes solicitados directamente al Director del Área de Central de Pagos.

#### Sistema Central de Contabilidad:

- Se tiene que recurrir a la libreta de registro de cheques, para saber cuáles fueron entregados durante el día y así poder separar las pólizas para su registro en el Sistema Central de Contabilidad.
- El registro de información al sistema de contabilidad se realiza manualmente y a nivel de cuenta contable.
- La duplicidad de captura de información contenida en las pólizas, provoca errores en la afectación contable.
- Debido a la gran cantidad de pólizas y a que la afectación contable se debe realizar el mismo día en que es entregado un cheque, se cuenta con un centro de captura exclusivo para el registro de información en el Sistema Central de Contabilidad.

## 2.3 ANÁLISIS DE LA SITUACIÓN ACTUAL

### 2.3.1 ANÁLISIS DEL SOFTWARE

Actualmente las Instituciones Bancarias dependen más de la tecnología para asegurar a sus clientes un servicio 100% confiable y eficiente. Además, internamente en sus servicios administrativos deben proporcionar a sus empleados herramientas que garanticen tiempos

de respuesta óptimos para la transferencia de información administrativa-contable, que repercute en un mejor control de los gastos operativos.

Para garantizar lo anterior, las Instituciones Bancarias utilizan software que les permiten tener uso simultáneo a la base de datos por múltiples usuarios y herramientas para el control y manipulación de información en línea.

El software con el que normalmente cuentan algunas Instituciones Bancarias, se enumera a continuación:

Dentro de los sistemas operativos podemos mencionar: WINDOWS NT , UNIX y AS400

Dentro del manejador de base de datos DBMS, tenemos: ORACLE, DB2 e INFORMIX.

Dentro de los lenguajes de programación, se encuentran: COBOL, RPG y C.

El software anteriormente descrito, básicamente se ocupa para la administración de la base de datos central, a través de la cual se lleva el control contable de gastos, el control de la cartera vigente y vencida, movimientos de clientes, pagos de los diferentes créditos, cuentas de cheques, etc.

Ahora bien, el software para el desarrollo de aplicaciones departamentales, varía de acuerdo al presupuesto, a los estándares y a las políticas de cada Institución Bancaria. Partiendo de lo anterior son necesarias aplicaciones que no representen costos elevados para su desarrollo, elaboradas con lenguajes de programación que no representen costos extra por licencias de uso, de fácil instalación y que utilicen una interfaz gráfica amigable para el usuario final.

Para la explotación y manipulación de información proveniente del equipo central, así como la información que será registrada en éste, el usuario final (departamental) hace uso de las aplicaciones proporcionadas por Microsoft: Excel, Word, Power Point y Access.

### 2.3.2 ANÁLISIS DEL HARDWARE

En lo que se refiere a la plataforma tecnológica de equipos y comunicaciones, las Instituciones Bancarias tienen especial interés en contar con una infraestructura que les dé la tranquilidad de poder ofrecer un servicio eficiente, primordialmente en la atención a los clientes en su red de sucursales y que a su vez pueda ser utilizada por toda su infraestructura administrativa.

Para poder satisfacer estas necesidades cuentan con comunicaciones vía satelital, garantizando así que las transacciones bancarias se realicen en línea y sean 100% seguras para sus clientes. Además, cuentan con líneas telefónicas privadas de ancho de banda de mayor capacidad a las convencionales, que permiten tener tiempos de respuesta óptimos entre sus servidores.

Asimismo, se cuenta con servidores mainframe HP 9000, equipos A16 de UNIX, equipos AS400 de IBM y servidores Windows NT, en los que residen principalmente las aplicaciones de la red de sucursales.

Por lo regular, las redes instaladas son del tipo estrella o árbol, conectadas a través de cable UTP o fibra óptica, para garantizar la conectividad entre los servidores.

Toda esta infraestructura se mantiene dentro de una Institución Bancaria para poder soportar y albergar los sistemas de gran impacto, siendo algunos de estos: el sistema de la red de sucursales, a través del cual se realizan todas transacciones diarias de la sucursal, el sistema de cheques, el Sistema Central de Contabilidad, el sistema de control de cartera, entre otros.

Para poder establecer un sistema departamental, se deberá tomar en cuenta, que solamente se podrá hacer uso de una pequeña parte de esta infraestructura, para no entorpecer las actividades esenciales que tienen relación directa con el cliente, pero sin perder de vista que las áreas departamentales también son importantes dentro de la actividad diaria de una Institución Bancaria.

Normalmente, las Instituciones Bancarias asignan un servidor por cada área que es vital y tiene que ver con el buen funcionamiento del banco, tal es el caso del Área de Central de Pagos.

Por último, el equipo de cómputo con el que cuentan algunas Instituciones Bancarias y que tienen distribuidos en sus áreas departamentales, son equipos: con procesadores 486 a 66 Mhz, con memoria RAM de 16 MB y discos duros de 600 MB; y en otros casos, equipos con procesadores PENTIUM I y II a 100 o 166 Mhz, con memoria RAM de 16 a 32 MB y discos duros de 3 a 20 GB.

## 2.4 ESTRATEGIA DE SOLUCIÓN

Debido a la problemática existente, es necesario desarrollar un sistema departamental que automatice la forma de operación para la obtención de un pago, que sea capaz de generar información estadística de una manera adecuada y eficiente, y que cubra todas las áreas involucradas.

El desarrollo del sistema se implementará considerando dos causas, la primera consistirá en dotar a todas las áreas administrativas que originan gasto de un **Sistema para el Trámite de Pagos**; la segunda consistirá en implementar un **Sistema Administrativo-Contable de Pagos** que le permita al Área de Central de Pagos controlar las operaciones de sus Áreas de Ventanilla, Emisión de Pagos y la parte de control encargada del registro contable en el Sistema Central de Contabilidad.

Para poder establecer esta solución, el Área de Central de Pagos deberá mantener un esquema de 4 grandes bloques, en donde cada uno de ellos englobará funciones específicas que permitan tener una solución para cada uno de los problemas detectados, lo cual en su conjunto permitirá tener una operación óptima y de calidad para los usuarios que soliciten algún pago, la figura 2.1 ilustra esta situación:

TÉCNICOS  
**FALLA** N

**DIAGRAMA GENERAL DE LA ESTRATEGIA DE SOLUCIÓN:**

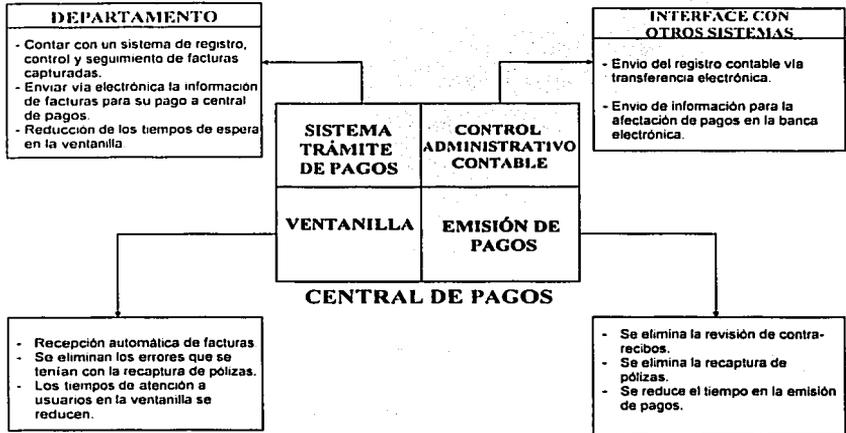


Figura 2.1 Estrategia de solución

El desarrollo del sistema departamental, implicará un cambio importante en el procedimiento a seguir para la solicitud de algún pago, éste consistirá en la eliminación de la generación de pólizas que actualmente realizan las áreas administrativas, para que estas sean concentradas y elaboradas en la propia Área de Central de Pagos, evitando así el llenado incorrecto y la recaptura de información que actualmente se tiene.

Lo anterior, permitirá poder establecer una guía contable única y conformada por conceptos de pago sencillos que serán de fácil comprensión para el personal de las áreas administrativas, el flujo de información bajo este concepto se muestra en la figura 2.2:

DIAGRAMA CONCEPTUAL DE LA ESTRATEGIA DE SOLUCIÓN:

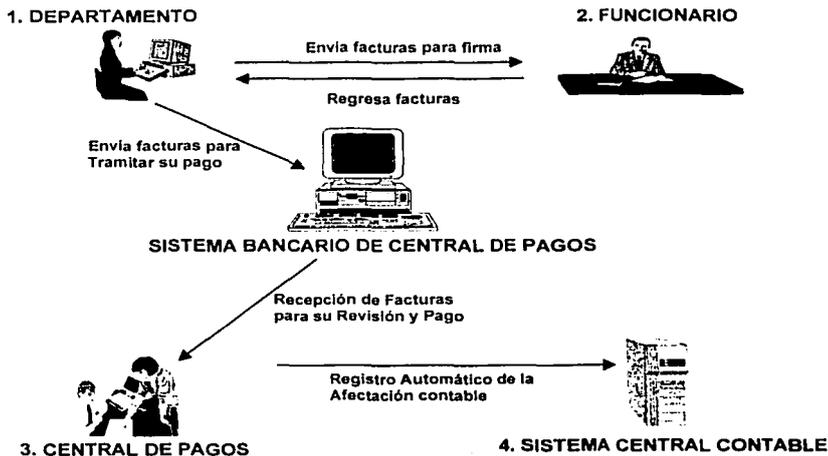


Figura 2.2 Esquema conceptual de solución

Las principales funciones con las que contará cada involucrado en esta solución, son:

1. DEPARTAMENTO

- 1.1. Captura Facturas.
- 1.2. Valida importes de Facturas.
- 1.3. Asigna referencia a Facturas.
- 1.4. Imprime relación de Facturas.
- 1.5. Envía Facturas para autorización del funcionario facultado.
- 1.6. Recibe Facturas autorizadas o rechazadas del funcionario facultado.
- 1.7. Envía relación de Facturas autorizadas al Área de Central de Pagos.

## 2. FUNCIONARIO

- 2.1. Recibe Facturas para su firma.
- 2.2. Autoriza/rechaza Facturas.
- 2.3. Regresa las Facturas.

## 3. ÁREA CENTRAL DE PAGOS

- 3.1. Recibe en forma automática las Facturas.
- 3.2. Genera y entrega los contra-recibos correspondientes.
- 3.3. Revisa las Facturas.
- 3.4. Autoriza/rechaza la emisión de pago.
- 3.5. Genera la emisión de cheques o abonos a cuenta de cheques.
- 3.6. Genera el registro contable de las Facturas.
- 3.7. Realiza la emisión de Pólizas.
- 3.8. Envía al Sistema Central de Contabilidad el registro contable.

## 4. SISTEMA CENTRAL CONTABLE

- 4.1. Recibe en forma automática el registro contable.
- 4.2. Valida el registro contable.
- 4.3. Realiza la aplicación contable.

Para poder implementar la solución propuesta, y considerando la infraestructura tecnológica, las posibles alternativas son:

- A. Desarrollar el sistema en un gestor de bases de datos.
- B. Desarrollar el sistema con herramientas Visual Basic y Access.
- C. Desarrollar el sistema con herramientas xBase utilizando FoxPro.

Analizando la opción A:

Ventajas:

- Los gestores de bases de datos DBMS permiten consultar, manipular y controlar los datos mediante SQL.

- Los gestores de bases de datos cuentan con su propio motor de base de datos relacional.
- La aplicación puede plasmar en línea cualquier cambio que se realice en la base de datos.
- Se aprovecha la infraestructura de red para la instalación del sistema.

#### Desventajas:

- En general los gestores de bases de datos, cuentan con demasiados módulos, por lo que se deberá contar con programadores que conozcan bastante bien el producto, traduciéndose esto en un mayor costo para la capacitación y el desarrollo del sistema.
- El desarrollo y diseño de sistemas en los gestores de bases de datos es largo y tedioso, aunado a que se debe contar con otros productos para su correcto funcionamiento.
- Se requiere configurar las terminales que tendrán acceso al sistema, por lo que deberá contar con licencias extras para la instalación de éste, lo cual elevaría el costo de instalación.
- Normalmente las aplicaciones diseñadas en gestores de bases de datos, no son 100% visuales y el diseño de pantallas son poco amigables para usuarios finales.
- Debido a la cantidad de usuarios nuevos que entrarían con la puesta en operación del sistema, se saturaría el servidor, lo cual pudiera provocar caídas del sistema central que repercutirá a otros sistemas y usuarios.

#### Analizando la opción B.

#### Ventajas:

- Access es la base de datos más usada que ofrece Microsoft.
- Capacidad para adaptarse a cambios tecnológicos, tanto de hardware como de software.
- Bajo costo.

- Visual Basic es un software para fronts que es 100% programado para hacer conexión, consultas y modificaciones a la base de datos que se encuentra detrás.
- Visual Basic permite crear aplicaciones 100% visuales, amigables y de fácil uso para los usuarios finales.

#### Desventajas:

- Para la implementación del sistema se tendrá que anclarse (casarse) a un back-end, en este caso Access, lo cual complica un poco el mantenimiento ya que se deberá contar con alguien más especializado que un usuario para la administración del sistema.
- Visual Basic no cuenta con una base de datos propia, por lo que debe conectarse a los datos a través de ODBC, lo cual complica un poco la configuración en las terminales.
- La mancuerna Visual Basic- Access no ofrece una gran robustez de almacenamiento de datos, por lo que al paso del tiempo se tendrá que pensar en migrar los datos a SQL.

#### Analizando la opción C.

#### Ventajas:

- FoxPro es una herramienta poderosa de desarrollo rápido de aplicaciones.
- Mejor distribución de la carga de procesamiento.
- Capacidad para adaptarse a cambios tecnológicos, tanto de hardware como de software.
- FoxPro tiene su propia base de datos relacional, lo cual permite un mejor control de los datos y una fácil portabilidad de una máquina a otra.
- Es una herramienta potente para sistemas administrativos muy económica.
- Tiene la capacidad de convertirse en cliente servidor en caso de crecimiento, sin tener que reprogramar mucho.

- Los archivos DBF (Data Base File) tienen una gran facilidad de exportación o importación para compartir, ver o modificar con otros programas que van desde Office hasta otros lenguajes.
- Al igual que Visual Basic, FoxPro tiene la posibilidad de crear aplicaciones 100% visuales, amigables y de fácil uso para los usuarios finales.
- El lenguaje xBase incluye apoyo para Índices, lo que le da acceso rápido a vastas cantidades de datos, un requisito indispensable para las aplicaciones de negocios.

**Desventajas:**

- La base de datos propia de FoxPro tiene una capacidad de almacenamiento de 2 Gigabytes por tabla como máximo.

**Selección de alternativa:**

Cabe mencionar que cualquiera de las tres opciones presentadas ofrecen tecnológicamente la posibilidad de poder implementar el sistema, pero la que mejor se adapta a las necesidades y limitantes tanto presupuestales, como de infraestructura tecnológica es la opción C.

Para solventar la desventaja planteada en la alternativa seleccionada, que hace referencia a la capacidad de almacenamiento, se implementarán las siguientes acciones:

- Eliminar redundancia en los datos que serán almacenados, cuando se diseñen las tablas del sistema.
- Permitir solamente un histórico de hasta un año y medio en las tablas del sistema.
- Establecer mecanismos de respaldo y restauración de información, y
- En lo referente a la información capturada por los departamentos, ésta residirá localmente en sus equipos.

## 2.5 JUSTIFICACIÓN DEL SISTEMA

Considerando que el sistema está orientado hacia aquellas Instituciones Bancarias que no cuentan con tecnología de vanguardia y en donde su infraestructura tecnológica corre bajo ambientes UNIX o AS400, con bases de datos Oracle, DB2 o INFORMIX, y con lenguajes de programación COBOL y RPG, aunado a que no cuentan con suficiente presupuesto para implementar un cambio de infraestructura en todos sus sistemas departamentales, la mejor opción resulta ser: desarrollar, diseñar e implementar el sistema a través de herramientas xbase utilizando FoxPro.

xBase es un lenguaje de alto nivel diseñado para producir aplicaciones de negocios, las opciones que típicamente requieren muchos comandos en otro lenguaje de programación, en xBase se ejecutan fácilmente, por ejemplo no se tiene que conocer la estructura física de sus datos, como si se tiene que hacer en un lenguaje de programación general. El punto es que el lenguaje xbase tiene apoyo incluido para todos los requisitos comunes en las aplicaciones de negocios, esto hace posible construir nuevos sistemas rápidamente a un costo bajo y de fácil mantenimiento.

Para poder implementar el sistema, solo se deberá contar con una licencia de uso de software de FoxPro, no son necesarias licencias adicionales para la instalación del sistema en las terminales, ya que los sistemas creados en FoxPro son 100% distribuibles sin ningún costo extra.

Para la instalación del sistema, solamente se requiere que las terminales cuenten con Windows, no será necesaria la configuración de alguna base de datos o driver ODBC, esto debido a que FoxPro cuenta con su propia base de datos, la cual se distribuye con una librería llamada FOXW2600.ESL que únicamente requiere 2.5 Megabytes de espacio en disco duro.

Con la realización de este sistema, se podrá eliminar todos los procesos de recaptura de información en Excel, se optimizará el almacenamiento de la información disminuyendo considerablemente el tiempo para obtener el pago de un servicio, y se podrá contar con información estadística para la toma de decisiones.

Se contará con mayor seguridad y veracidad de la información, además, de ser fácilmente actualizada.

Para poder actualizar los datos en el Sistema Central de Contabilidad se implementará el uso de una interface, que envíe mediante un archivo plano en formato ASCII la afectación contable correspondiente, el cual será cargado ejecutando un proceso batch diariamente, evitando con esto la saturación del gestor de bases de datos, sin importar cual sea ORACLE, DB2, INFORMIX u otro.

## 2.6 MODELO DEL SISTEMA

Para el desarrollo de la aplicación será necesario hacer uso de métodos y herramientas que permitan planificar y analizar el proyecto, los métodos a utilizar serán los descritos anteriormente en el apartado 1.4 del presente trabajo.

Inicialmente se hará uso del modelo de datos entidad-relación para definir la información que será almacenada y procesada, posteriormente se utilizará el modelo de procesos, con la ayuda del método estructurado de datos, con el fin de comprender y establecer el flujo de información, así como su funcionamiento y comportamiento. Finalmente se definirá el modelo tecnológico a utilizar, en donde se indicarán los recursos informáticos necesarios para la implementación de la aplicación.

Con el uso de estos modelos se pretende obtener una visión coherente de la información, que se apegue a los objetivos y requisitos indispensables para garantizar la correcta funcionalidad de la aplicación.

### 2.6.1 MODELO DE DATOS

Durante el modelado de datos se realizará la abstracción de la información que intervendrá en los **Sistemas para el Trámite de Pagos y Sistema Administrativo-Contable de Pagos**, esto con el fin de obtener todas las entidades que serán almacenadas y procesadas en ambos sistemas.

## 2.6.1.1 DIAGRAMA ENTIDAD-RELACIÓN SISTEMA PARA EL TRÁMITE DE PAGOS

Los departamentos son las áreas internas de las Instituciones Bancarias que originan gasto, a través de éstos, se solicita al Área de Central de Pagos se realicen los pagos correspondientes por el servicio otorgado por empresas o personas externas a la institución.

Para el análisis y definición de los elementos que intervienen en el proceso que da origen a un gasto, nos apoyaremos en el modelo entidad-relación.

Primeramente tenemos al BANCO, el cual esta dividido a nivel nacional por REGIONES y dentro de cada región en PLAZAS.

La división en regiones o plazas dentro de un banco obedecen a que gracias a esta agrupación se pueden distribuir de manera más eficiente la administración y operación diaria de las diversas áreas, gráficamente las entidades son: (figura 2.3)

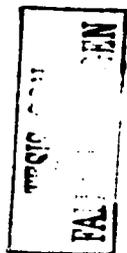


Figura 2.3 Relación de las entidades de banco, región y Plaza

Por otra parte tenemos a los DEPARTAMENTOS, los cuales son las entidades encargadas de solicitar un servicio, por ejemplo: papelería, servicio de fotocopiado, etc.

La satisfacción de esta necesidad es cubierta por un PROVEEDOR externo que ofrece su servicio no solo a un departamento sino a todo el banco previo convenio o contrato de prestación de servicios. Asimismo, podemos mencionar a los empleados de los bancos,

los cuales en algunas ocasiones tienen el comportamiento de un proveedor, dicha situación ocurre cuando realizan actividades fuera de su área habitual de trabajo, tales como los viajes y traslados de un edificio a otro, gráficamente las entidades son: (figura 2.4)

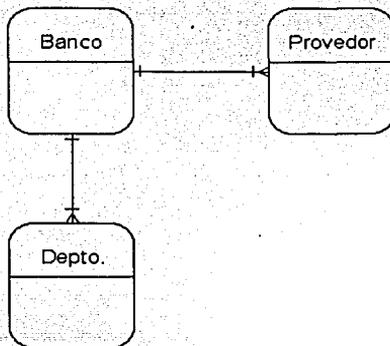


Figura 2.4 Relación de las entidades de banco, proveedor y departamento

El elemento que soporta la erogación de un gasto es el documento o FACTURA que es entregado por el PROVEEDOR a los DEPARTAMENTOS. Asimismo, dentro de la factura se encuentra el DETALLE y el CONCEPTO del servicio efectuado, cuando el PROVEEDOR es un empleado que origina gastos de viaje y VIÁTICOS, tenemos la ZONA y CIUDAD a la que viajó, por otra parte, para la comprobación de los gastos de viaje y viáticos, contablemente se asigna un número único ÍTEM, a través del cual se lleva el control de forma individual de estos gastos, gráficamente las entidades son: (figura 2.5)

TESIS CON  
FALLA DE ORIGEN

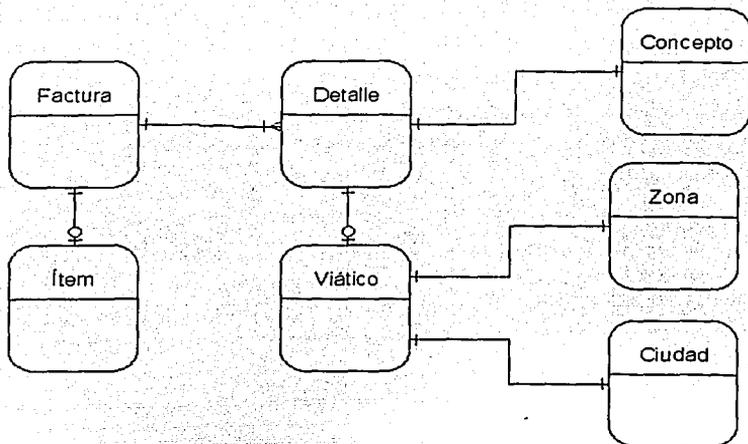


Figura 2.5 Entidades que intervienen con las facturas (documentos)

Como se vio en el análisis de la solución propuesta, se elimina la generación de pólizas contables dentro de los departamentos, las cuales serán sustituidas por LOTES de facturas que se enviarán al Área de Central de Pagos, gráficamente las entidades son: (figura 2.6)

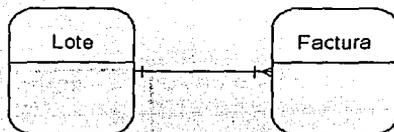


Figura 2.6 Relación de las entidades de lote y factura

Por último, agrupando todas las entidades involucradas en la generación de gastos por parte de los departamentos, tenemos el modelo de datos resultante mostrado en la figura 2.7.

E-R Trámite de Pagos

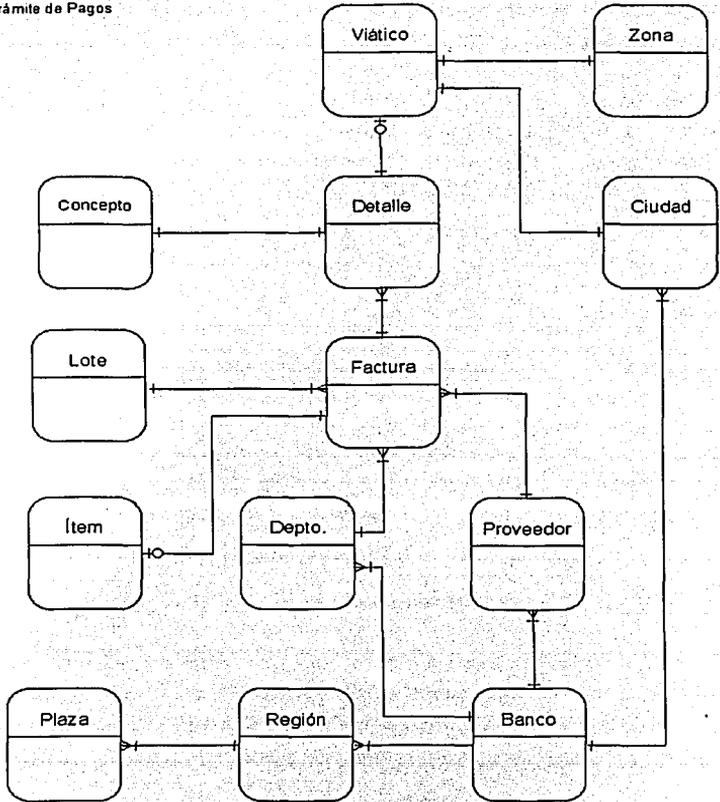


Figura 2.7 Modelo E-R para el trámite de pagos

FALLA EN EL PROCESO

## 2.6.1.2 DIAGRAMA ENTIDAD-RELACIÓN SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

Dentro de la estructura del Área de Central de Pagos, la primera área que tiene contacto con los departamentos que originan gasto es la **Ventanilla**, a través de la cual se realiza la recepción de documentos para la tramitación de un pago.

La recepción de documentos en la Ventanilla será acompañada por los archivos de transferencia enviados por los departamentos, en **LOTES** de **FACTURAS**, los cuales tendrán asignados un número de referencia para identificar el documento, y así poder asignar un número de **CONTRA-RECIBO**, gráficamente las entidades son: (figura 2.8)

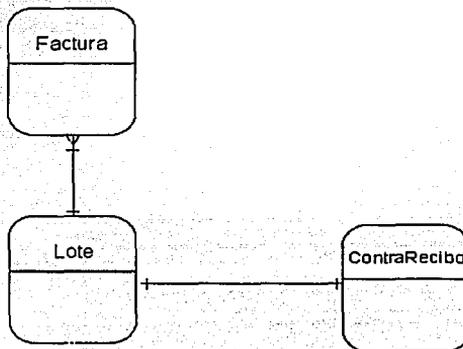


Figura 2.8 Entidades que intervienen con el contra-recibo

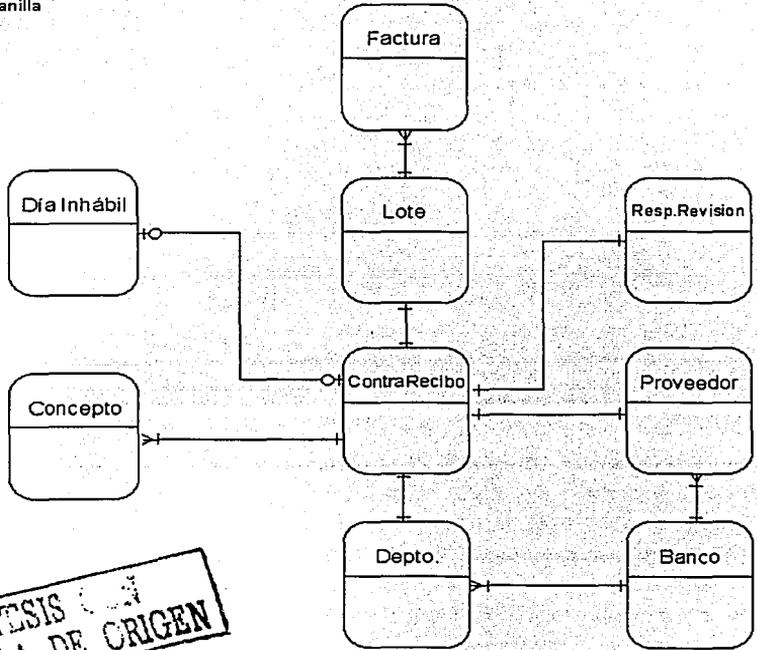
Ahora bien, el documento soporte que servirá para realizar el trámite de un pago dentro del Área de Central de Pagos será el **CONTRA-RECIBO**, el cual contará con las entidades **BANCO**, **DEPARTAMENTO**, **PROVEEDOR** y **CONCEPTO**, provenientes de las **FACTURAS**.

Otra de las entidades que se deberá considera son los **DÍAS INHÁBILES**, esto con el fin de no asignar fechas de pago en esos días, por otra parte para facilitar la entrega de los

CONTRA-RECIBOS se deberá asignar un RESPONSABLE DE REVISIÓN, el cual se asignará de acuerdo al CONCEPTO de pago.

Considerando todo lo anterior, el modelo de datos resultante para la VENTANILLA se muestra en la figura 2.9:

E-R Ventanilla



TESIS CON FALLA DE ORIGEN

Figura 2.9 Modelo E-R de la Ventanilla

Una vez que la Ventanilla ha recibido la documentación soporte para el otorgamiento de un pago y ha entregado los contra-recibos correspondientes, toca el turno del **Área de Emisión de Pagos**, la cual en primera instancia realiza la selección de **CONTRA-RECIBOS**, de acuerdo a los **RESPONSABLES DE REVISIÓN** previamente asignados por la Ventanilla, las entidades que intervienen se muestran en la figura 2.10:

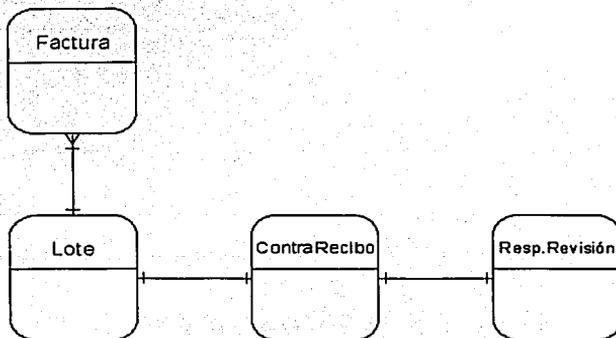
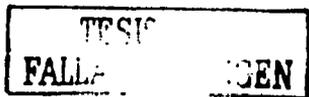


Figura 2.10 Entidades que intervienen con los responsables de revisión

Cada **RESPONSABLE DE REVISIÓN** selecciona sus **CONTRA-RECIBOS** y comienza a validar la información registrada contra la documentación soporte, para determinar si procede o se **RECHAZA** el pago, este control de movimientos se determina por el **ESTATUS** que se asigna al **CONTRA-RECIBO**, los principales rubros que se verifican son el **CONCEPTO** de pago, los datos del **PROVEEDOR**, el **DEPARTAMENTO** que origina el gasto, las cantidades registradas en los documentos y la fecha programada de pago, entre otros, las entidades que intervienen se muestran en la figura 2.11:



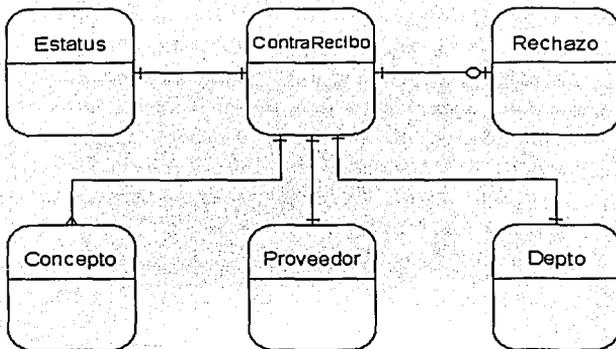


Figura 2.11 Entidades que se usan para la revisión de contra-recibos

Teniendo los CONTRA-RECIBOS validados, éstos se turnan para la emisión de CHEQUES o ABONOS a las cuentas de los PROVEEDORES, la figura 2.12 muestra estas entidades:



Figura 2.12 Entidades utilizadas para la emisión de cheques o abonos

TESIS CON  
FALLA DE ORIGEN

En lo que se refiere a la emisión de CHEQUES o ABONOS, cabe hacer mención que la elaboración de éstos se realiza días antes de la fecha programada de pago, con el fin de tenerlos preparados para la entrega a los PROVEEDORES correspondientes, pero existen excepciones o casos urgentes que requieren ser elaborados el mismo día, dentro de los cuales podemos mencionar los viáticos de empleados, anticipos de abogados para el pago de interdictos o demandas, pagos de agua, luz y predial que tienen fecha de vencimiento en ese mismo día, pagos de impuestos a Hacienda, etc., por lo que es necesario realizar varias emisiones de CHEQUES o ABONOS en la fecha previamente programada, mismas que se registrarán en un CONTROL DE EMISIONES, asimismo, al momento de la impresión de cheques puede haber CANCELACIONES producto de fallas en la impresora o errores en el nombre del proveedor, por lo tanto, las entidades que se tienen son: (figura 2.13)

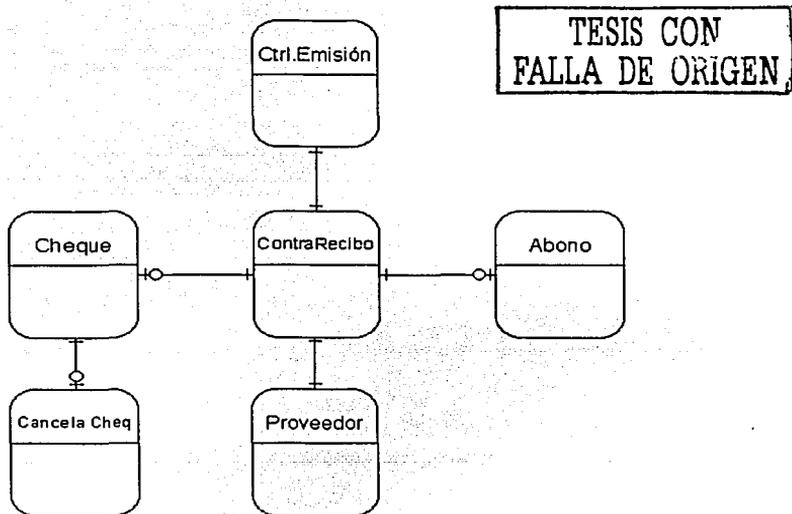


Figura 2.13 Entidades que intervienen durante la emisión de cheques o abonos

Considerando todo lo anterior, el modelo de datos resultante para la EMISIÓN DE PAGOS se muestra en la figura 2.14:

E-R Emisión Pagos

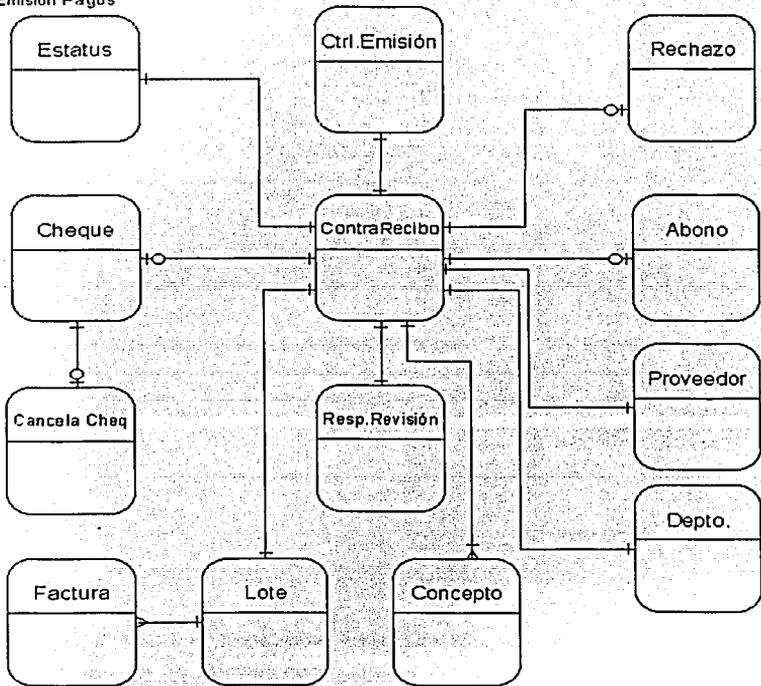


Figura 2.14 Modelo E-R de la emisión de pagos

TESIS CON FALLA DE ORIGEN

Durante los días de entrega de pagos y después del cierre de la Ventanilla, el Área Contable realiza la selección de las emisiones de pagos para determinar los CHEQUES o ABONOS que fueron entregados, adicionalmente, se seleccionan los CONTRA-RECIBOS para establecer el DEPARTAMENTO y CONCEPTO de pago con los que se generarán los movimientos contables correspondientes, las entidades que intervienen son: (figura 2.15)

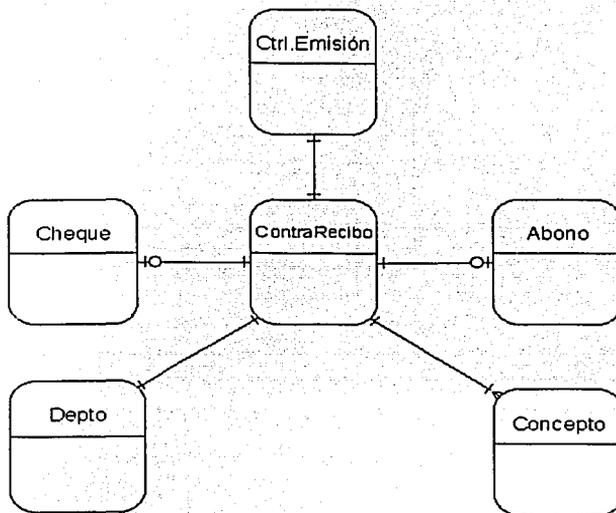


Figura 2.15 Entidades que intervienen para la selección de emisiones de pagos

Para realizar la contabilización de los pagos efectuados, el DEPARTAMENTO es la entidad a través de la cual se determina la aplicación contable de la retención del ISR, el IVA y la retención del IVA, por otra parte, para identificar el gasto se asignan CENTROS DE COSTO, en los cuales residirá contablemente el gasto que justifica la emisión de un pago.

Para saber que CENTROS DE COSTOS serán afectados, es necesario conocer a que BANCO, REGION y PLAZA pertenece el DEPARTAMENTO, por lo que las entidades que intervienen son: (figura 2.16)

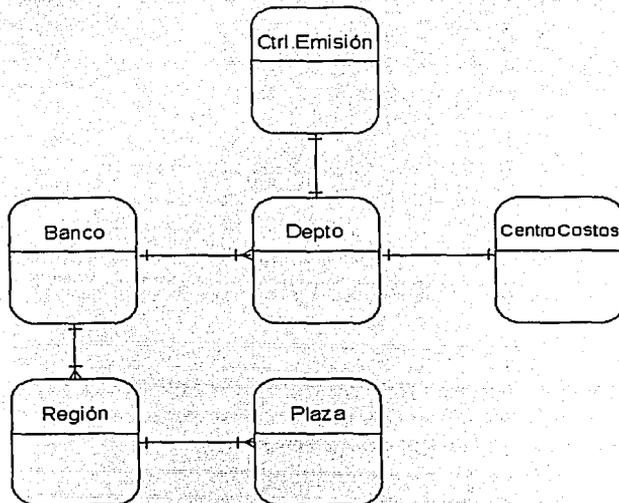


Figura 2.16 Entidades que intervienen para la asignación de centros de costos

La contabilización de movimientos se realiza mediante el uso de CUENTAS CONTABLES, la definición de éstas, dependen del CONCEPTO de pago indicado en el CONTRA-RECIBO, la aplicación de las cuentas contables se realiza de forma general, es decir, los gastos de honorarios se aplican a la cuenta 5108, los gastos de rentas a la 5109, sin importar el proveedor, pero existen movimientos que son específicos y requieren una cuenta única o ITEM ligada a una cuenta contable, por ejemplo, la comprobación de los gastos de viaje y viáticos de empleados son registrados en la cuenta 5112, pero por cada empleado se genera un ITEM único para tener un control de las comprobaciones de éste, las entidades que intervienen son: (figura 2.17)

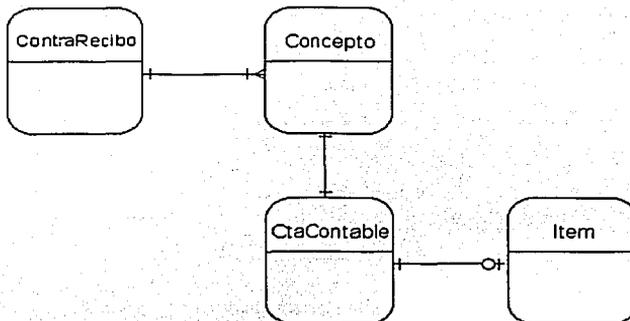


Figura 2.17 Entidades que intervienen en la asignación de cuentas contables

El último proceso realizado por el Área Contable es la generación de pólizas contables, quedando registradas en los movimientos de CARGOS y ABONOS producto de los pagos entregados en el día, pero puede darse el caso de errores en el proceso, por lo que se deberá llevar un control de ERRORES CONTABLES, las entidades que intervienen son: (figura 2.18)

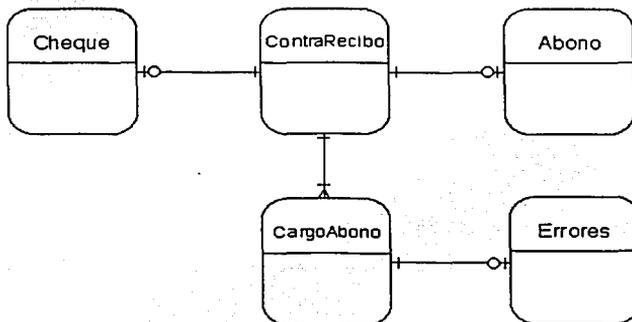


Figura 2.18 Entidades que intervienen en el control de errores contables

Considerando lo descrito anteriormente, la figura 2.19 muestra el modelo de datos resultante para la CONTABILIZACIÓN:

E-R Contabilización

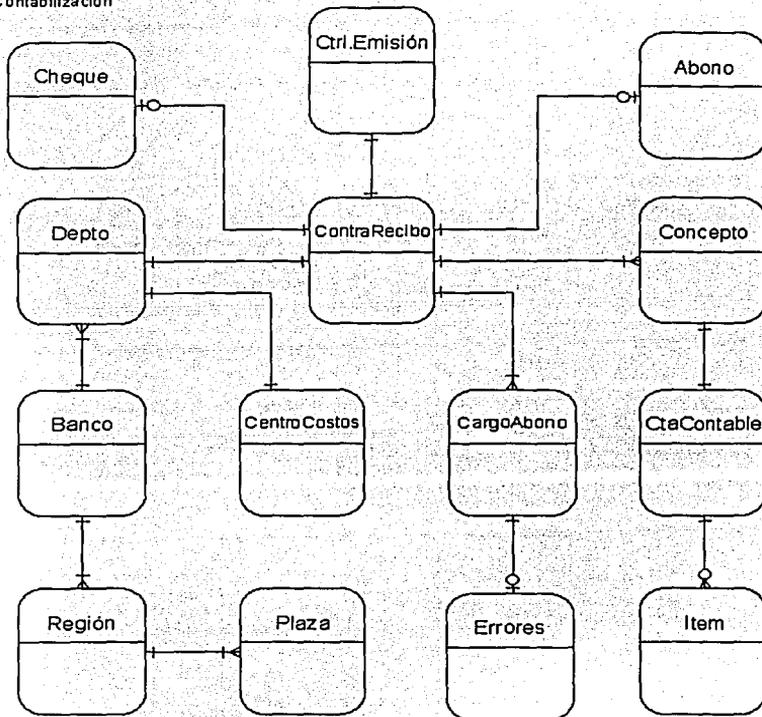


Figura 2.19 Modelo E-R para la contabilización

## 2.6.2 MODELO DE PROCESOS

Para describir la estructura de los procesos, así como el flujo de información que intervendrán durante el desarrollo del **Sistema para el Trámite de Pagos** y el **Sistema Administrativo-Contable de Pagos**, será necesario hacer uso de modelos que nos permitan representar el sistema como una red de procesos funcionales que interactuen entre sí, para tal tarea nos apoyaremos en los diagramas de flujo de datos.

La figura 2.20 presenta el diagrama de contexto del Sistema Bancario de Control de Pagos:

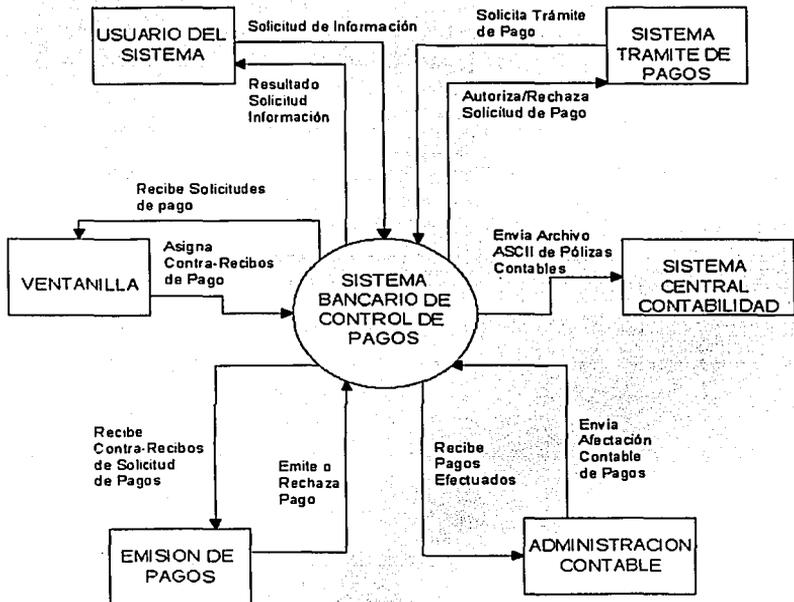


Figura 2.20 Diagrama de contexto del sistema bancario de control de pagos

## 2.6.2.1 DIAGRAMA DE FLUJO SISTEMA PARA EL TRÁMITE DE PAGOS

Para mostrar el flujo de los datos que intervienen en el **Sistema para el Trámite de Pagos**, se emplearán los diagramas de flujos a nivel 0 y nivel 1, quedando definidos de la siguiente forma:

### Diagrama de Flujo de Datos Nivel 0 (Sistema para el Trámite de Pago)

En la figura 2.21 se muestran los agentes externos que interactúan con el **Sistema para el Trámite de pagos**, conformados por: el proveedor, el cual solicita el pago por los servicios otorgados a la Institución; el empleado, que solicita el reembolso de las actividades extras, tales como comidas, pasajes, viáticos, etc., necesarias para la eficiente consecución de su funciones dentro de la Institución; el departamento, encargado de generar la relación de facturas recibidas; el funcionario, encargado de autorizar o rechazar las facturas recibidas; el usuario del sistema, que realizará la actualización y consulta de información; y finalmente, la Ventanilla, encargada de recibir la facturas autorizadas para su pago.

### Diagrama de Flujo de Datos Nivel 1 (Proveedor)

En la figura 2.22 se muestran los procesos que interactúan con el proveedor, comenzando con la solicitud del pago por sus servicios; el departamento revisa y envía la solicitud del pago a la Ventanilla, entregando para tal efecto el contra-recibo que ampara los documentos recibidos; y finalmente, después de la revisión de facturas, se realizar la entrega del pago correspondiente.

### Diagrama de Flujo de Datos Nivel 1 (Empleado)

En la figura 2.23 se muestran los procesos que interactúan con el empleado, comenzando con la solicitud del reembolso de servicios extras, la solicitud de anticipos de viaje, o en su defecto la comprobación de gastos de viaje; posteriormente el departamento revisa y envía la solicitud del pago a la Ventanilla, entregando para tal efecto el contra-recibo que

ampara los documentos entregados; y finalmente, después de la revisión de facturas, se realiza la entrega del pago correspondiente.

#### **Diagrama de Flujo de Datos Nivel 1 (Departamento)**

En la figura 2.24 se muestran los procesos que interactúan con el departamento, comenzando con la solicitud de pago por parte del proveedor o empleado, el cual se encargará de capturar, validar, autorizar y asignar un número de referencia a las facturas recibidas, para posteriormente generar la relación de facturas que serán enviadas a la Ventanilla del Área de Central de Pagos.

#### **Diagrama de Flujo de Datos Nivel 1 (Funcionario)**

En la figura 2.25 se muestran los procesos que interactúan con el funcionario, comenzando con la recepción de facturas para su autorización o rechazo, generando el lote de referencia para las facturas autorizadas, el cual se empleará para realizar el formato de solicitud de pago.

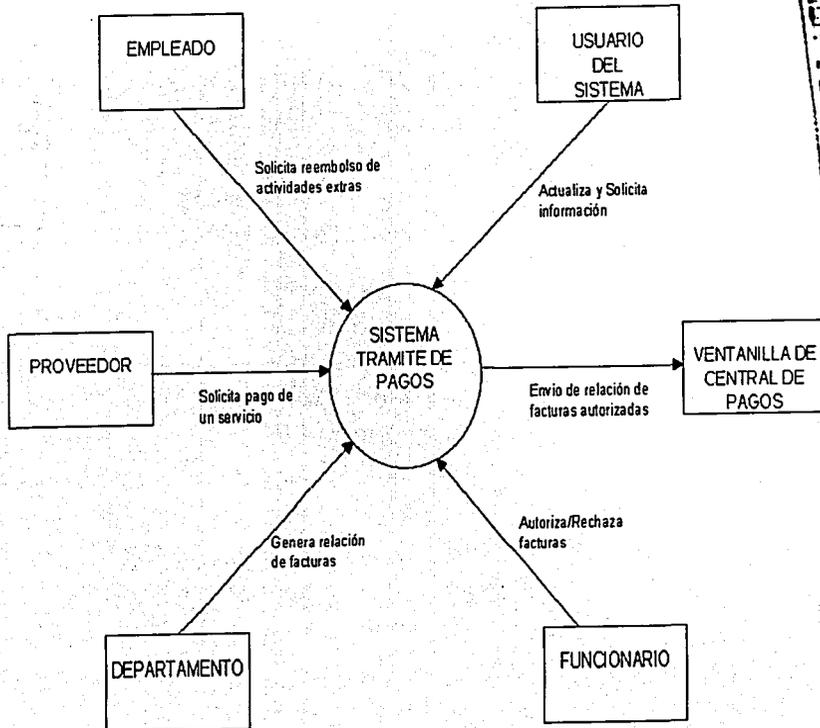
#### **Diagrama de Flujo de Datos Nivel 1 (Usuarios Solicitud de Información)**

En la figura 2.26 se muestran los procesos que interactúan con el usuario cuando solicita información al sistema, inicialmente el usuario seleccionará el banco y departamento para realizar la consulta de facturas capturadas, el detalle de viajes y viáticos de empleados, y la consulta de lotes enviados, así como su relación de facturas.

#### **Diagrama de Flujo de Datos Nivel 1 (Usuarios Actualización de Información)**

En la figura 2.27 se muestran los procesos que interactúan con el usuario cuando actualiza información del sistema, inicialmente el usuario seleccionará el banco y departamento para realizar la actualización de catálogos, así como el respaldo y restauración de información.

DIAGRAMA DE FLUJO DE DATOS NIVEL 0 (Sistema para el Trámite de Pagos)



TRABAJOS REALIZADOS  
FALLA DE REGISTRO

Figura 2.21

## DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Proveedor)

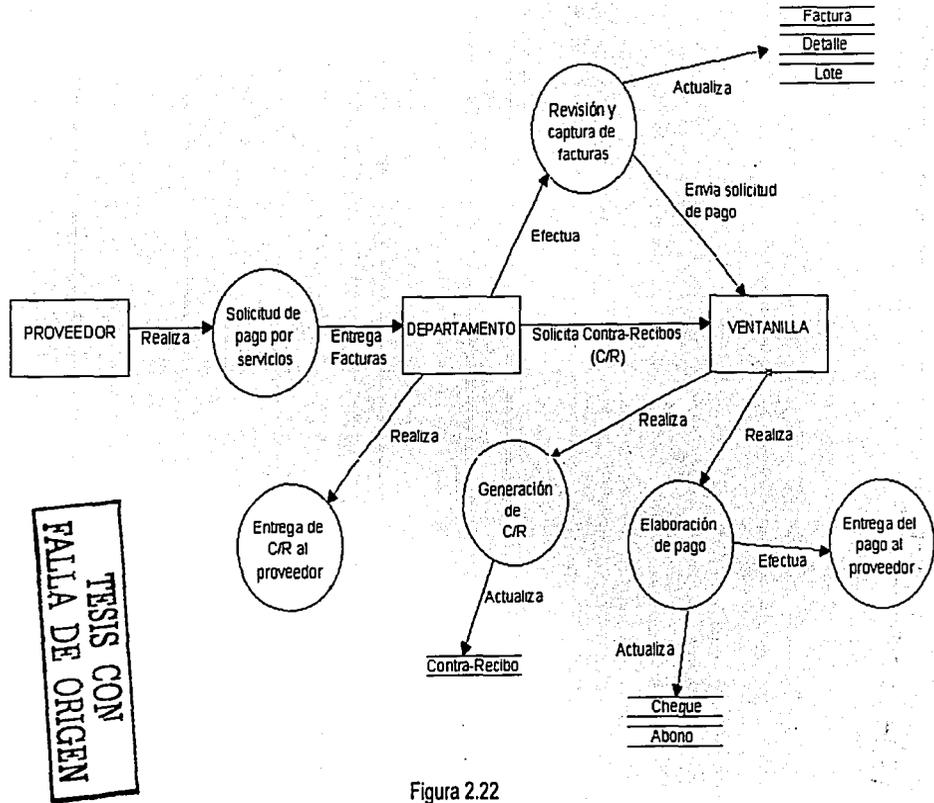


Figura 2.22

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Empleado)

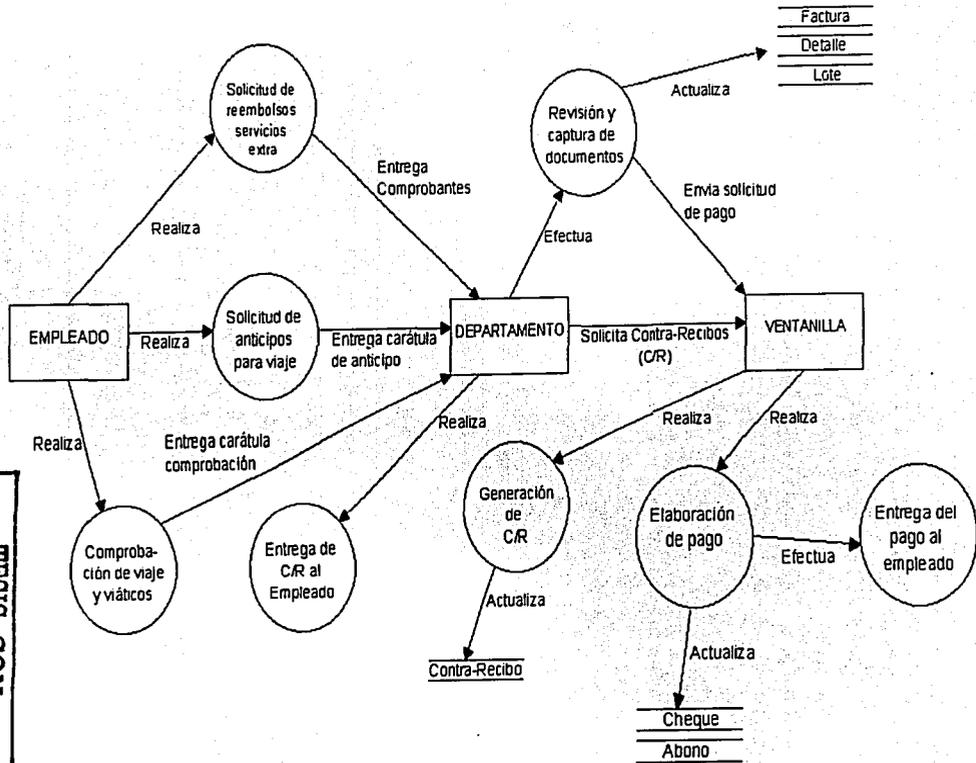


Figura 2.23

TESIS CON  
 FALLA DE ORIGEN

## DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Departamento)

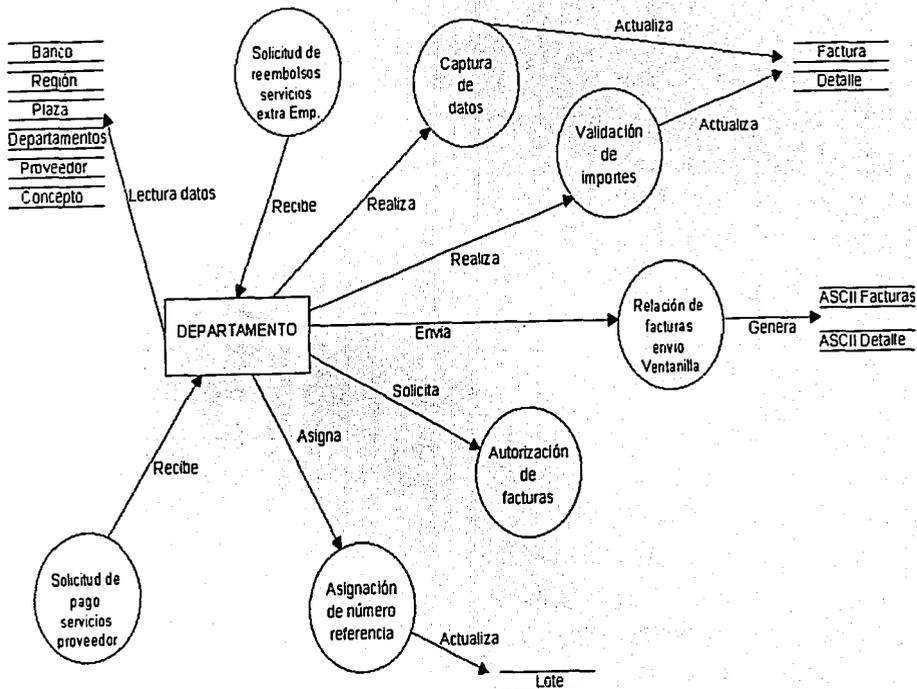
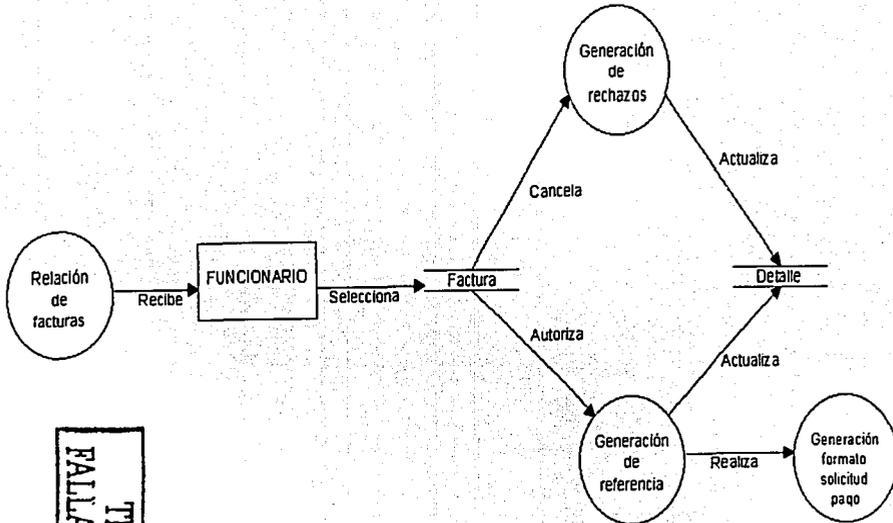


Figura 2.24

TESIS CON  
FALTA DE ORIGEN

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Funcionario)



TESIS CON  
FALLA DE ORIGEN

Figura 2.25

**TESIS CON  
FALLA DE ORIGEN**

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Usuarios Solicitud de Información)

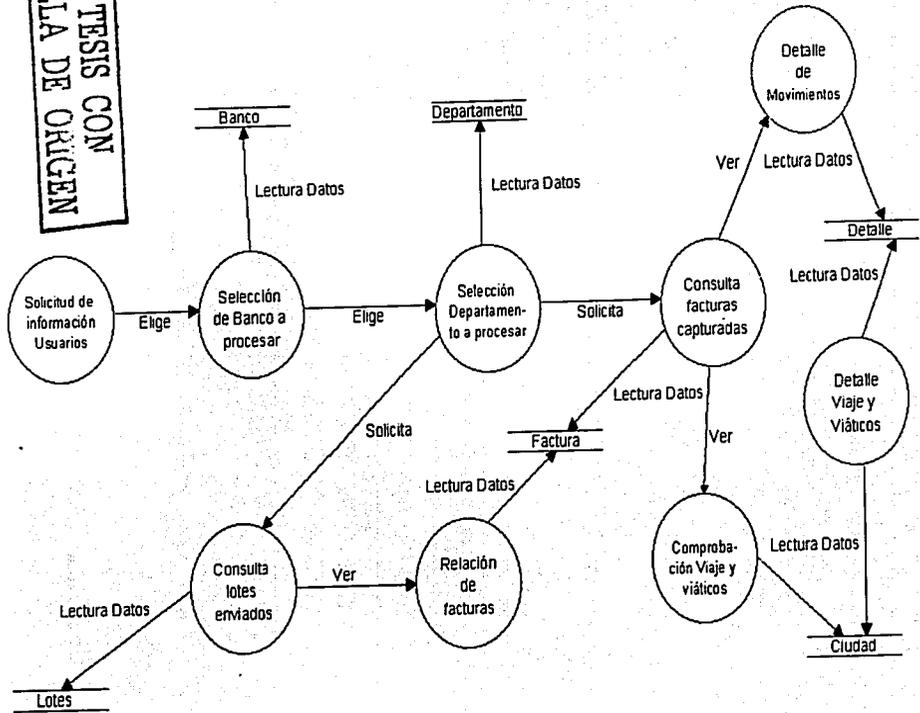


Figura 2.26

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Usuarios Actualización de Información)

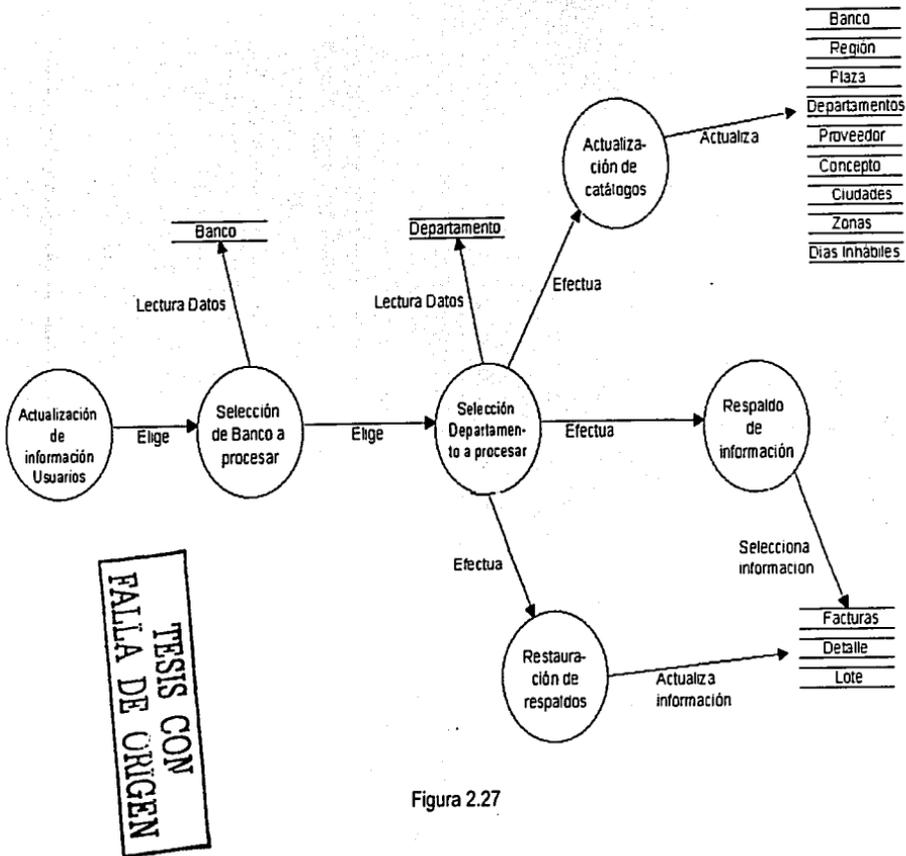


Figura 2.27

### 2.6.2.2 DIAGRAMA DE FLUJO SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

Para mostrar el flujo de los datos que intervienen en el **Sistema Administrativo-Contable de Pagos**, se emplearán los diagramas de flujos a nivel 0 y nivel 1, quedando definidos de la siguiente forma:

#### Diagrama de Flujo de Datos Nivel 0 (Sistema Administrativo-Contable de Pagos)

En la figura 2.28 se muestran los agentes externos que interactúan con el **Sistema Administrativo-Contable de Pagos**, conformados por: la Ventanilla, la cual genera los contra-recibos correspondientes de las facturas recibidas; la emisión de pagos, encargada de la elaboración de los pagos; la administración contable, encargada de generar la afectación contable de los pagos realizados; el usuario del sistema, que realizará la actualización y consulta de información; y finalmente, el Sistema Central de Contabilidad, encargado de realizar la afectación contable de las pólizas contables generadas por el Área de Central de Pagos.

#### Diagrama de Flujo de Datos Nivel 1 (Ventanilla)

En la figura 2.29 se muestran los procesos que interactúan con la Ventanilla, comenzando con la solicitud de pagos, la cual se realiza de forma automática; generando para tal efecto los contra-recibos que ampara los documentos recibidos, realiza la asignación de contra-recibos a los responsables de revisión para la verificación de documentos; cuando los pagos han sido autorizados realiza la entrega de cheques a los beneficiarios del pago y finalmente realiza la actualización de los catálogos utilizados por ésta.

#### Diagrama de Flujo de Datos Nivel 1 (Emisión de Pagos)

En la figura 2.30 se muestran los procesos que interactúan con la emisión de pagos, comenzando con recepción de contra-recibos autorizados para su pago, los cuales se seleccionarán para la generación de cheques o abonos, generando para tal efecto un

previo contacto, con el cual se puede determinar si procede la emisión de pagos o se realiza la cancelación de ésta.

#### **Diagrama de Flujo de Datos Nivel 1 (Administración Contable)**

En la figura 2.31 se muestran los procesos que interactúan con la administración contable, comenzando con la relación de emisiones de pagos efectuadas, a partir de las cuales se realizará el proceso de afectación contable, verificando que no existan errores contables, en caso de existir, se realizará la cancelación de los movimientos efectuados, caso contrario, se procederá a la elaboración de pólizas contables y generación del archivo de transferencia que se enviará al Sistema Central de Contabilidad.

#### **Diagrama de Flujo de Datos Nivel 1 (Usuarios Solicitud de Información)**

En la figura 2.32 se muestran los procesos que interactúan con el usuario cuando solicita información al sistema, inicialmente el usuario seleccionará el banco y departamento para realizar la consulta de contra-recibos emitidos, así como la consulta de pagos efectuados y pólizas enviadas al Sistema Central de Contabilidad.

#### **Diagrama de Flujo de Datos Nivel 1 (Usuarios Actualización de Información)**

En la figura 2.33 se muestran los procesos que interactúan con el usuario cuando actualiza información del sistema, inicialmente el usuario seleccionará el banco y departamento para realizar el mantenimiento de usuarios, realizar la carga automática de ítems contables, realizar la actualización de catálogos, realizar la transferencia de catálogos al Sistema para el Trámite de Pagos, así como el respaldo y restauración de información.

TESIS CON  
FALLA DE ORIGEN

## DIAGRAMA DE FLUJO DE DATOS NIVEL 0 (Sistema Administrativo-Contable)

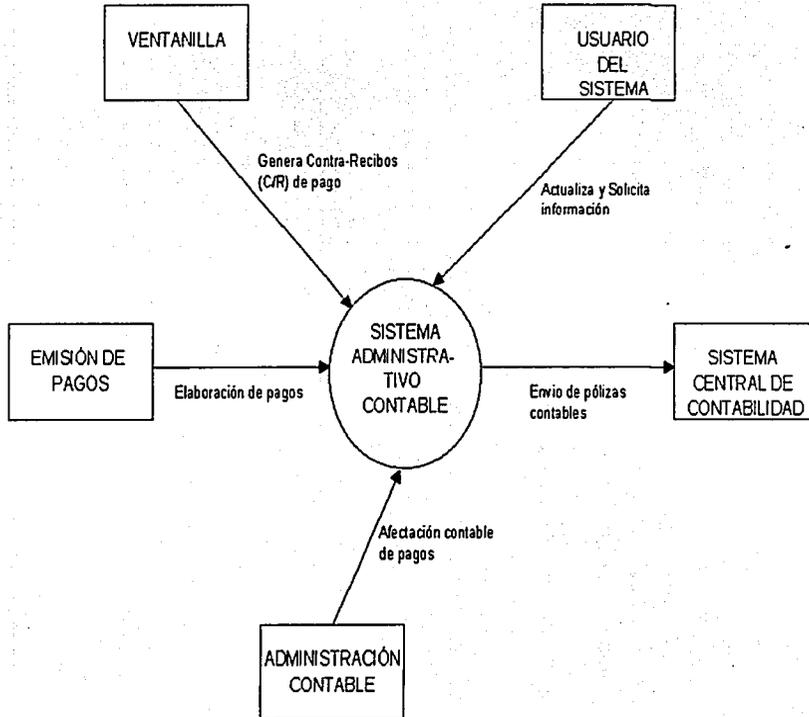
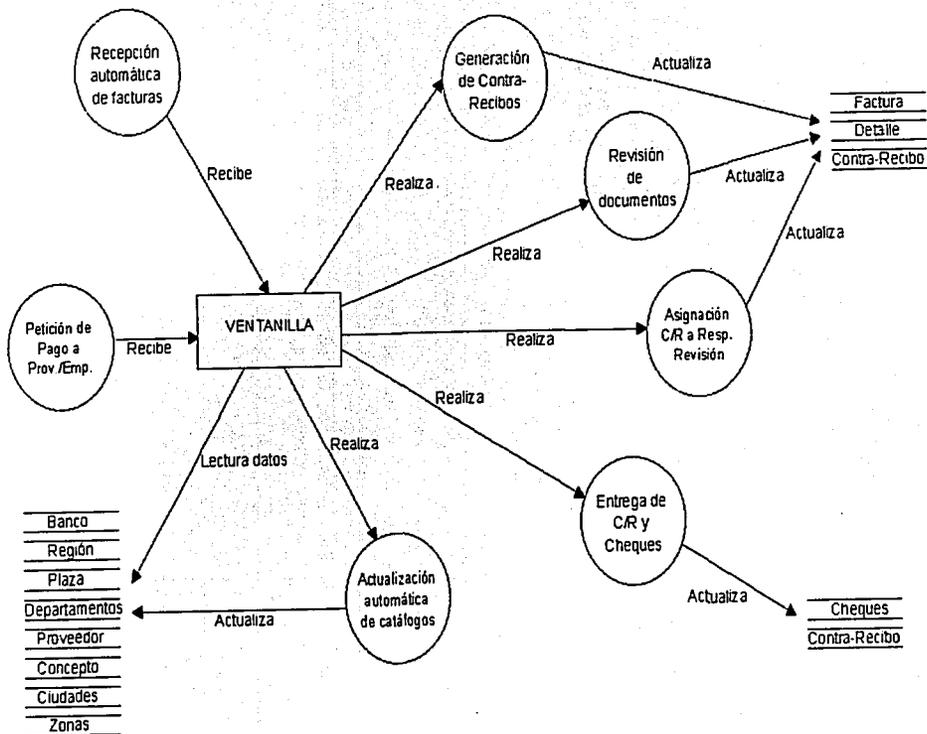


Figura 2.28

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Ventanilla)



TESIS CON  
 FALLA DE ORIGEN

Figura 2.29

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Emisión de Pagos)

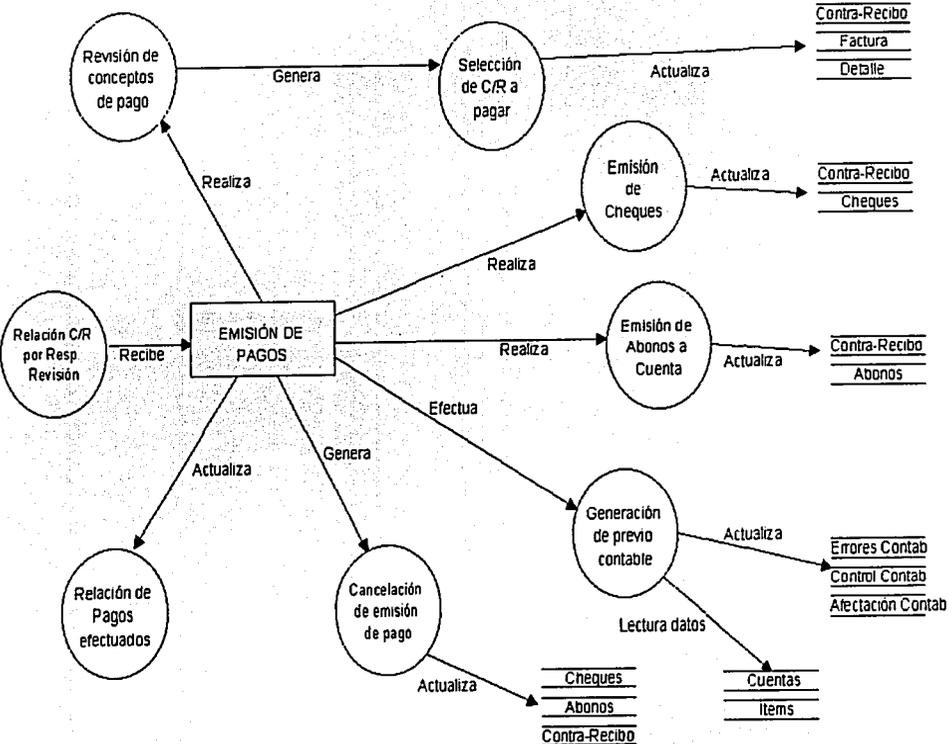


Figura 2.30

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Administración Contable)

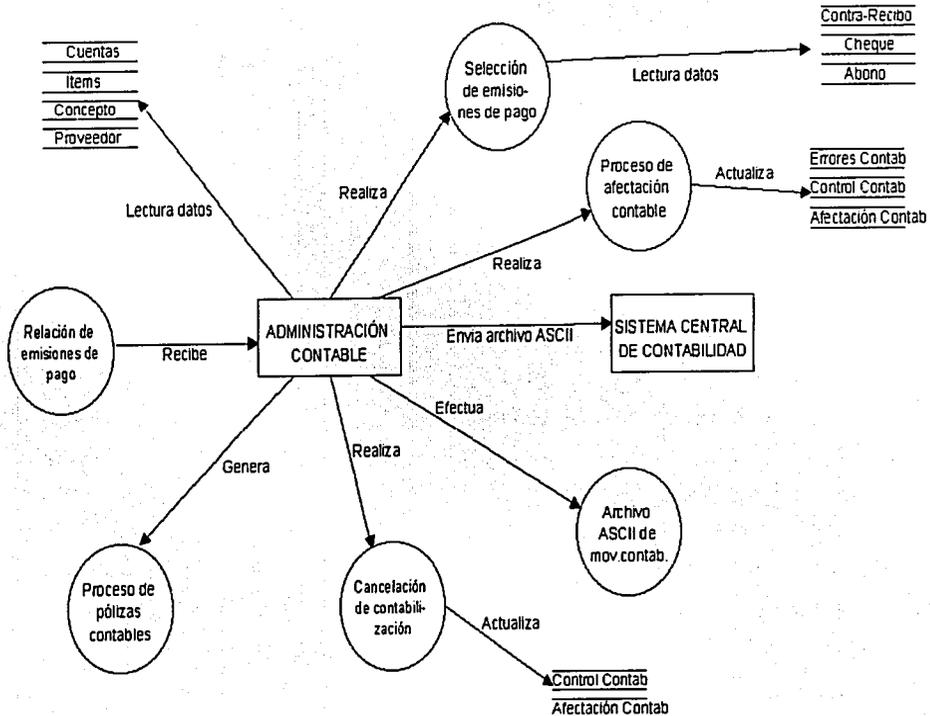


Figura 2.31

# DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Usuarios Actualización de Información)

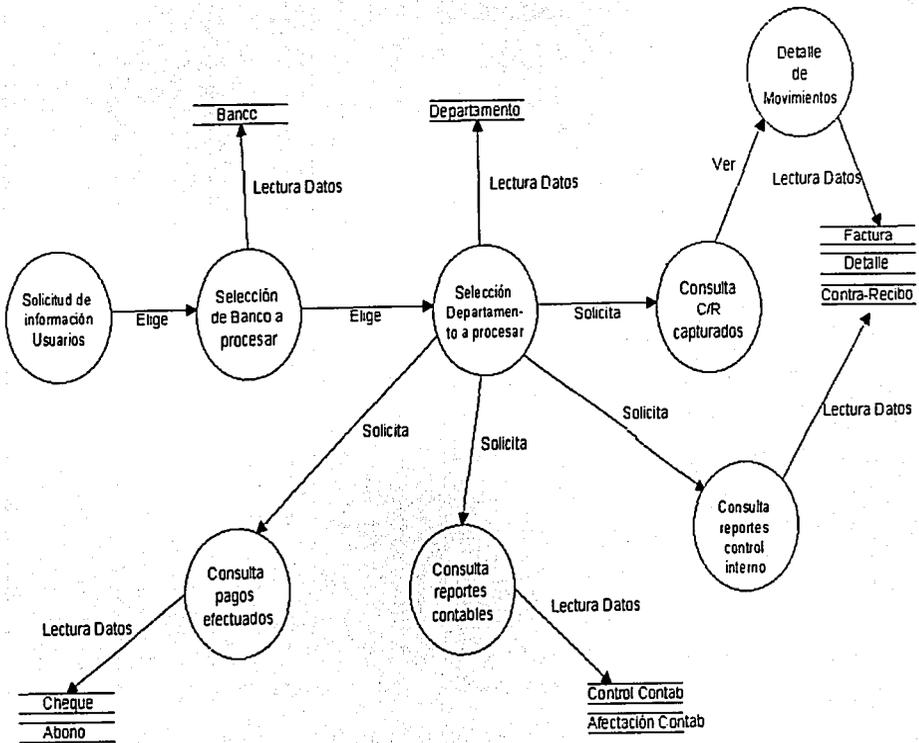


Figura 2.32

DIAGRAMA DE FLUJO DE DATOS NIVEL 1 (Usuarios Actualización de Información)

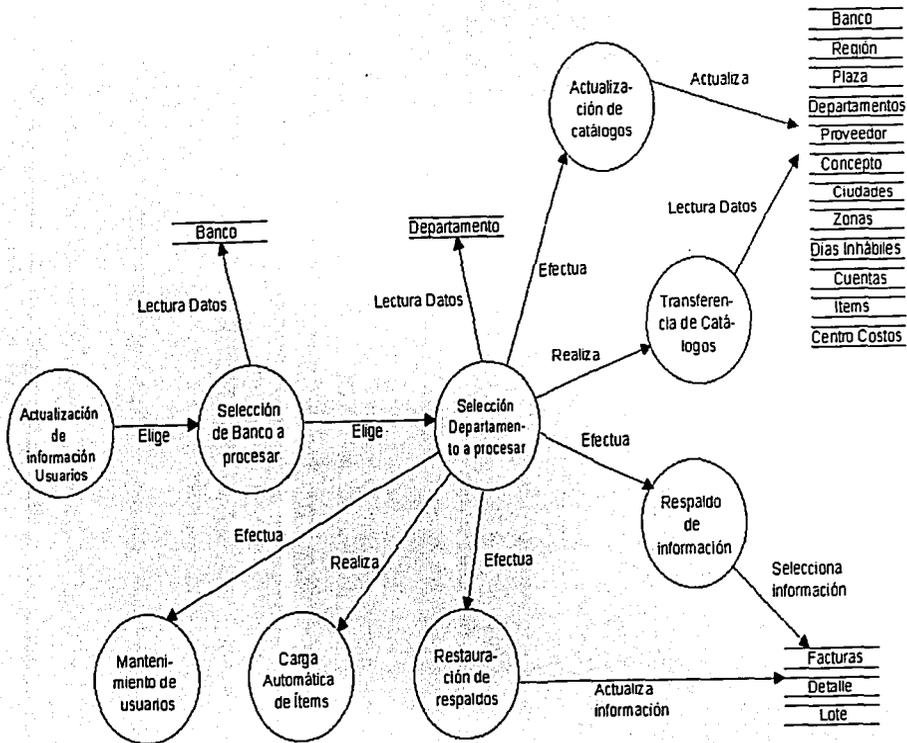


Figura 2.33

## 2.6.3 MODELO TECNOLÓGICO

Como ya se menciono anteriormente en el análisis y justificación del sistema, la plataforma tecnológica que se utilizará para la implementación y desarrollo del sistema consistirá en utilizar el lenguaje de programación FOXPRO, el cual emplea la tecnología xBase para la creación de tablas en las que se almacenará la información.

La arquitectura xBase se utilizará en la implantación del sistema como el back-end y el lenguaje FOXPRO se utilizará como nuestro front-end.

### 2.6.3.1 BACK-END

Ahora bien, profundizando un poco más en la arquitectura xBase, esté es el nombre que se le da genéricamente a todos aquellos lenguajes de programación cuyo conjunto de instrucciones y funciones son un derivado de los de dBASE III+, y que manejan archivos de datos que tienen la estructura conocida como DBF (Data Base File).

El formato nativo de archivos de datos de FOXPRO es la estructura conocida como DBF, que fue mostrada por primera vez en el dBASE III+, si bien muchos programadores "modernos" la consideran vieja, obsoleta y decadente, el archivo DBF tiene la sencillez de un clásico, porque fue creado para durar, visto desde el punto de vista romántico, un DBF es como las Pirámides de Egipto, un ejemplo vivo de una cultura antigua pero que ha durado por siglos, por mucho más tiempo, que obras maestras de la arquitectura moderna como Las Torres Gemelas de NY.

Un archivo DBF puede almacenar los siguientes tipos de datos: (figura 2.34)

Carácter	Cadenas de caracteres de longitud variable de hasta 64 kbytes de longitud
Numérico	Hasta 19 dígitos de longitud, contando el punto decimal en campos no enteros

Fecha	Longitud fija de 8 bytes
Lógica	Valores booleanos (.T. y .F.) de 1 byte de longitud
Memo	Cadenas de caracteres de longitud variable los cuales se almacenan en un archivo con la extensión .FTP. El valor mínimo para un campo memo es de 1 byte, aunque por omisión se maneje un valor de 15 bytes, estos memos ofrecen la ventaja de que su tamaño mínimo puede ser establecido por el usuario desde programa.

Figura 2.34 Tipos de datos de un archivo DBF

El archivo DBF está dividido en 2 partes: el *Área de Encabezado* (Header Area) y el *Área de Datos* (Data Area).

El *Área de Encabezado* almacena información sumamente importante, dividida en 2 partes: *Información del archivo* (file information) y *Descripción de campos* (Field description).

Todos los datos en el *Área de encabezado* están escritos en formato hexadecimal, el formato del encabezado en su sección *Información de Archivo* es como sigue: (figura 2.35)

Byte	Información almacenada
0	Este byte indica si se ha especificado la existencia de campos memo. Si no hay memos, el valor es 03h, si los campos memo existen para este archivo el valor es un 83h.
1	Almacena el año de la última actualización, sólo se almacenan los dos últimos dígitos del año, por ejemplo si el archivo fue modificado en 1999 contendrá un valor de 63h.
2	Almacena el mes de la última actualización, Enero (01h), Febrero (02h), Marzo (03h), Abril (04h), Mayo (05h), Junio (06h), Julio (07h), Agosto (08h), Septiembre (09h), Octubre (0Ah), Noviembre (0Bh) y Diciembre (0Ch)

3	Almacena el día de la última actualización, su valor mínimo es 1 (01h) y el máximo es 31 (1Fh).
4 al 7	Estos 4 bytes almacenan el número de registros dentro de la base de datos, por lo que el número de registros que puede tener un archivo DBF esta definido por el valor de estos 4 bytes, si pensamos que su valor máximo es de FFFFFFFFh, tenemos que en teoría, el número máximo de registros que puede almacenar un DBF es de <b>4,294,967,295</b> .
8 y 9	Declaran la longitud del encabezado.
10 y 11	Indican la longitud total del registro, incluyendo la bandera de borrado.  Mucho se ha hablado del número de campos que puede contener la estructura de un DBF, unos dicen que 100, unos que más, otros que menos, lo cierto es que el número de campos que puede contener un DBF, va en razón directa de su longitud, entre menor sea la longitud más campos podemos tener, por lo que el número de campos esta precisamente establecido por estos 2 bytes, cuyo máximo valor es FFFFh ( <b>65,535</b> ), mientras la longitud de todos los campos de la base de datos no exceda este valor, podremos tener tantos campos como necesitemos.
12 al 31	No se usan.

Figura 2.35 Sección de Información de un archivo DBF

*Descripción de campos:* Después de los primeros 31 bytes de la sección de *Información de archivo*, viene la sección de *Descripción de campos*, que comienza en el byte 32 (20h). Cada campo dentro de la estructura del DBF, sin importar su tipo ni su longitud genera 32 bytes dentro de esta sección, de tal forma que si nuestra base de datos tiene 10 campos, esta sección medirá 320 bytes, si tiene 100 medirá 3200 bytes y así sucesivamente, como es fácil adivinar esta sección no tiene una longitud fija, ya que esta en función del número de campos que tenga nuestro DBF.

En general, cada campo proporciona la siguiente información a la sección de *descripción de campo*: (figura 2.36)

Byte	Información almacenada
0 al 10	Estos 11 bytes almacenan el nombre del campo. Los nombres de los campos pueden tener de 1 a 10 caracteres de longitud. El carácter que se utiliza para indicar la terminación del nombre de campo es un 00h.
11	Especifica el tipo de campo: Carácter 43h; Numérico 4Eh; Fecha 44h; Lógico 4Ch y Memo 4Dh.
12 al 15	Cuatro bytes que especifican el espacio que ocupará en la memoria el campo.
16	Especifica la longitud del campo
17	Especifica la longitud decimal del campo, si se trata de un campo numérico no entero
18 al 31	No se usan.

Figura 2.36 Sección descripción de campo de un archivo DBF

Después de la última descripción de campo, se escribe el carácter que indica la terminación, que es un 0Dh. A partir de este punto, y hasta el final del archivo se extiende el *Área de datos*.

En el *Área de Datos*, se almacena la información como tal de la base de datos, los datos son almacenados en bloques x+1 bytes donde x es la longitud total de todos los campos descritos y el byte adicional (+1) es la bandera de borrado. Este byte de borrado tiene un valor de 20h (espacio) cuando el registro está activo y un 2Ah (asterisco \*) cuando el registro ha sido marcado para borrar, esto en el caso de los drivers NTX y CDX, si se esta utilizando Advantage Database Server se utiliza para controlar los procesos de transacciones cuando éstas se manejan en entornos Cliente / Servidor.

Cada registro dentro del DBF tiene una longitud fija, como la descripción de campos. Los datos son recuperados siempre por su posición en relación al principio del registro; no se utiliza ninguna señal de delimitación para separar los campos, el motor de base de datos utiliza la información de la sección *Información de Archivo* bytes 10 y 11 para calcular en dónde comienza y dónde termina cada registro y recuperarlos de manera óptima, porque

en esta sección todo se almacena en formato ASCII, en otras palabras el *Área de datos* es un simple archivo de texto.

La forma de almacenamiento es como sigue:

- Las cadenas de caracteres se almacenan en formato ASCII.
- Los valores numéricos se almacenan en formato decimal usando el punto (.) como un separador entre la parte entera del número y la parte decimal. Un signo de menos (-) puede preceder a los valores numéricos.
- Los campos lógicos contienen un signo de interrogación (?) hasta que se modifique el campo, los valores válidos son Y, N, T, F, y, n, t, f.
- Los campos de fecha contienen 4 bytes. Los primeros dos bytes almacenan la porción correspondiente al año (incluyendo el siglo), el tercer byte contiene el mes y el cuarto contiene el día.
- Finalmente, los campos memo contienen una referencia de 10 bytes al bloque correspondiente dentro del archivo .FTP lo cual garantiza una comunicación de 2 vías entre DBF y FPT, reduciendo el riesgo de generar corrupción de datos en uno u otro archivo.

El algoritmo de recuperación de datos es bastante sencillo de imaginar, si se ejecuta desde el programa una instrucción `DBGOTO(5)`, internamente el motor de datos obtendrá la longitud de todo el registro a partir de la información contenida en los bytes 10 y 11 de la sección *Información de archivo*, una vez calculada la longitud total del registro, esta se multiplica por 5, y el motor de base de datos sabe el número de bytes que deberá desplazarse a partir del comienzo del *área de datos* hasta el comienzo del registro 5, una vez que ha llegado al principio del registro que queremos recuperar, nuevamente se utiliza la longitud del registro para saber cuantos bytes tiene que leer y acomodarlos en sus respectivas variables de nombre de campo. Como se puede apreciar, la estructura DBF provee de una solución sencilla y elegante para la manipulación de datos, aunque muchos la consideren vieja y obsoleta.

### 2.6.3.2 FRONT-END

FoxPro 2.6 es una herramienta que ya no tiene la actualidad de las nuevas herramientas como el Visual Fox, Visual Basic entre otras, pero esto no indica necesariamente que es mala o inferior. Por el contrario está programada en una plataforma lo suficientemente probada y estable como para garantizar la confianza de transacciones seguras y eficientes. Además tiene el respaldo de más de ocho años de estar con una presencia constante en el mercado.

Por otra parte, las aplicaciones desarrolladas en FoxPro 2.6 no tiene limitación de usuarios. Puede ser usada por uno, dos, tres usuarios y no se tienen que pagar licencias adicionales de uso. Es decir una sola licencia de uso, permite que la aplicación sea accesada por una cantidad ilimitada de usuarios. La única restricción es que se instale una sola aplicación en un servidor de archivos, que es accesada por los usuarios por medio de la red.

Las características más sobresalientes que ofrece este lenguaje para el desarrollo de aplicaciones de negocios son las siguientes:

#### Creación de Tablas y archivos de Índice:

Núm. máximo de registros por tabla	1.000 millones
Núm. máximo de caracteres por registro	65.500
Núm. máximo de campos por registro	255
Núm. máximo de archivos .DBF abiertos	225
Núm. máximo de caracteres por campo	254
Núm. máximo de caracteres por clave de Índice (IDX)	100
Núm. máximo de caracteres por clave de Índice (CDX)	240
Núm. máximo de archivos abiertos de Índice por tabla	ilimitado
Núm. máximo de archivos abiertos de Índice en todas las áreas de trabajo	ilimitado
Núm. máximo de relaciones	ilimitado
Longitud máxima de las expresiones relacionales	ilimitada

El tamaño de archivo real (en bytes) no puede exceder de 2 gigabytes para un solo usuario o para tablas multiusuario abiertas en exclusiva.

Las tablas compartidas sin índices o con índices -IDX no pueden exceder de 1 gigabyte.

Las tablas compartidas con índices estructurales .CDX no pueden exceder de 2 gigabytes.

#### Características de campos:

Tamaño máximo de los campos de caracteres 254

Tamaño máximo de los campos numéricos 20

Núm. máximo de caracteres de los nombres de campo 10

Dígitos de precisión en cálculos numéricos 16

#### Variables de memoria y matrices:

Núm. predeterminado de variables de memoria 256

Núm. máximo de variables de memoria 65.000

Núm. máximo de matrices 65.000

Núm. máximo de elementos por matriz 65.000

#### Archivos de programas y procedimientos:

Núm. máximo de líneas de los archivos de programa origen ilimitado

Tamaño máximo de una única FUNCION o PROCEDIMIENTO compilado 64K

Un programa o una aplicación pueden contener un número ilimitado de módulos de programa.

Núm. máximo de procedimientos por archivo ilimitado

Núm. máximo de llamadas DO anidadas 32

Núm. máximo de niveles de anidamiento READ 5

Núm. máximo de comandos anidados de programación estructurada 64

Núm. máximo de parámetros de procedimiento 24

### Capacidades del Generador de informes:

Núm. máximo de objetos de una definición de informe ilimitado

Núm. máximo de líneas de una definición de informe 255

Núm. máximo de niveles de agrupamiento 20

### Soporte de ventanas:

Núm. máximo de ventanas abiertas ilimitado

Núm. máximo de ventanas Examinar abiertas 225

### Capacidades varias:

Núm. máximo de caracteres por cadena 2 gigabytes

Núm. máximo de caracteres por línea de comandos 2.048

Núm. máximo de caracteres por sustitución de macro 2.048

Núm. máximo de archivos abiertos Límite del MS-DOS

Núm. máximo de pulsaciones por macro de teclado 1.024

Núm. máximo de campos que puede seleccionar una instrucción SELECT de SQL-255.

## **2.6.3.3 CARACTERÍSTICAS DEL SISTEMA**

### Características de la aplicación:

Sistema desarrollado en lenguaje Foxpro For windows.

Multiusuario

Ambiente Windows.

Bases de datos tipo xBase.

**Características de la instalación:**

En equipos PC se instalará el archivo ejecutable y las librerías correspondientes.  
En el servidor se instalarán las bases de datos.

**Requerimientos mínimos para el equipo pc:**

Equipo con procesador 486 ó mayor.  
Windows 95 ó mayor.  
12 Mb. de memoria RAM.  
10 Mb de espacio en disco duro.  
Conexión a la Red.

**Requerimientos para el servidor:**

Sistema operativo Windows NT.  
Procesador Pentium III, intel 400 Mhz.  
128 Mb de memoria RAM.  
100 Mb de espacio en disco duro.  
Grupo de usuarios para el acceso al sistema.



### 3.1 DISEÑO DE DATOS

Tomando como base el modelo de Entidad-Relación establecido previamente, aunado al levantamiento de información realizado durante las reuniones de trabajo con los usuarios del sistema y al análisis de la situación actual, se puede comenzar a definir el diseño de los datos, siendo éstos la parte medular para un correcto funcionamiento de cada uno de los módulos que integrarán el sistema.

#### 3.1.1 DISEÑO DE DATOS SISTEMA PARA EL SISTEMA TRÁMITE DE PAGOS

Tomando en cuenta el diagrama Entidad-Relación y aplicando la normalización de las tablas definidas, tenemos:

La tabla ST\_POLCA, en la cual se registrarán todos los datos concernientes a las facturas, quedando: (figura 3.1)

CAMPO	OBSERVACIÓN
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
BANCO	Clave del Banco al que pertenece la factura
CVE_DEPTO	Clave del departamento que recibe la factura para tramitar su pago.
PROVEEDOR	Clave del proveedor que solicita el pago de un servicio.
NO_DOCTO	Número de Factura.
FEC_DOCTO	Fecha de la Factura.
TIPO_GASTO	Tipo de Gasto al que se refiere la factura, Administrativo, Anticipo de Viáticos, Comprobación de Viáticos, Boleto de Avión.
VIÁTICO	Identifica si el pago solicitado corresponde a viáticos.

DETALLEDOC	Identifica si la factura se desglosará en varios movimientos.
ITEM	Campo que contiene la cuenta contable para la comprobación de un anticipo.
SERIE	Campo que identifica de forma individual el documento de la comprobación de un anticipo.
CONCEPTO	Clave que identifica el concepto de pago de la factura
DEPTOGASTO	Clave del departamento que origina el gasto.
FORMA_PAGO	Campo que identifica la forma de pago de la factura, ya sea cheque o abono a cuenta de cheques.
IMPORTE	Importe de la Factura.
DESCUENTO	Descuento aplicado.
IVA	Importe del IVA aplicado.
RET_IVA	Retención de IVA aplicado.
RET_ISR	Retención de ISR aplicado.
TOTAL	Importe Total de la Factura a pagar.
COMENT1	Comentarios que describen de forma detallada el concepto de pago.
COMENT2	Complemento de comentarios.
ANTICIPO	Importe que registra el anticipo otorgado.
SALDO_COMP	Importe que registra el saldo de las comprobaciones.
ESTATUS	Situación que tiene una factura dentro del proceso de generación de su pago.
CVE_TURNO	Clave que identifica a que persona se turno para su revisión.
FEC_POLIZA	Fecha que indica cuando se autorizo el pago de la factura.
NO_POLIZA	Número de referencia asignado cuando de autoriza la factura.
FEC_ENVIO	Fecha de envío de la factura al Área de Central de Pagos.
LOTE_ENVIO	Lote de envío al Área de Central de Pagos.

DEPTO_USU	Clave del Departamento del Usuario que captura la factura.
FEC_ALTA	Fecha de alta de la factura en el sistema.
FEC_VALOR	Indica la fecha del registro contable de los gastos originados por la factura, esto sucede cuando el pago se realiza fuera de tiempo.

Figura 3.1 Tabla ST\_POLCA

La tabla ST\_MOVCA, en la cual se registrarán todos los datos concernientes al detalle de las facturas cuando éstas afecten varios conceptos de pago, sean anticipos o comprobaciones de viajes y viáticos, quedando: (figura 3.2)

CAMPO	OBSERVACIÓN
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_DETALLE	Campo único que identifica cada detalle de la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
DEPTOGASTO	Clave del departamento que origina el gasto del detalle de la factura.
NO_DOCTO	Número de documento que forma parte de la Factura.
FEC_DOCTO	Fecha del documento.
CONCEPTO	Clave que identifica el concepto de pago de cada documento de la factura.
IMPORTE	Importe del documento.
POR_DESC	Porcentaje del descuento por aplicar.
DESCUENTO	Importe del Descuento aplicado.
POR_IVA	Porcentaje del IVA por aplicar.
IVA	Importe del IVA aplicado.
POR_RETIVA	Porcentaje de la Retención del IVA por aplicar.

RET_IVA	Importe de la Retención de IVA aplicado.
POR_RETISR	Porcentaje de la Retención del ISR por aplicar.
RET_ISR	Retención de ISR aplicado.
TOTAL	Importe Total del documento.

Figura 3.2 Tabla ST\_MOVCA

La tabla ST\_VIAJE, en la cual se registrarán todos los datos concernientes al detalle de los movimientos de viaje y viáticos, quedando: (figura 3.3)

CAMPO	OBSERVACIÓN
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_DETALLE	Campo único que identifica cada detalle de la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_VIAJE	Campo único que identifica el detalle de las ciudades que forman parte del viaje, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
CI_SALIDA	Ciudad Inicial (CI) de donde se origina el viaje.
CI_DESTINO	Ciudad destino del viaje.
CI_DIAS	Días del viaje.
CI_FECHA	Fecha de inicio del viaje.
CI_HORA	Hora de inicio del viaje.
CI_BOLETO	Identifica si se requiere boleto de avión.
CF_SALIDA	Ciudad Final (CF) de donde se concluirá el viaje.
CF_DESTINO	Ciudad de regreso del viaje.
CF_FECHA	Fecha de regreso del viaje.
CF_HORA	Hora de regreso del viaje.

CF_BOLETO	Identifica si se requiere boleto de avión para el regreso del viaje.
FINVIAJE	Identifica si este registro concluye la captura del viaje.
CIUDAD	Ciudad de la cual se esta realizando la comprobación de viáticos.
A_REALES	Días reales del viaje.
A_FECINI	Fecha real de inicio del viaje
A_FECFIN	Fecha real de finalización del viaje.

Figura 3.3 Tabla SC\_VIAJE

La tabla ST\_LOTE, en la cual se registrarán los lotes para el envío de facturas al Área de Central de Pagos, quedando: (figura 3.4)

CAMPO	OBSERVACIÓN
LOTE	Campo único que identifica el lote de envío de las <b>factura. Se tomará como Llave Principal.</b>
BANCO	Clave del banco al que pertenece el lote.
AÑO	Año de generación del lote.
INI_FIRMA	Iniciales del Funcionario que firma el lote
MES	Mes de generación del lote
FOLIO	Número de folio del lote.
DEPTO_USU	Clave del departamento que genero el lote.
FECHAENVIO	Fecha de envío de lote.
FEC_VALOR	Si las facturas son fuera de tiempo, la afectación contable queda establecida por la fecha indicada en este campo.
VALOR	Valor total de las facturas que conforman el lote.
FACTURAS	Número de facturas que conforman el lote.
CVE_DEPTO	Clave del departamento que autorizo las facturas.
NOM_DEPTO	Descripción del departamento que autorizo las facturas.
NOM_RESP	Nombre del funcionario que autorizo las facturas.
ESTATUS	Situación que determina la etapa del proceso de un lote.

Figura 3.4 Tabla SC\_LOTE

Adicionalmente a estas tablas, serán necesarios catálogos de solo consulta, para facilitar la captura y evitar errores en la información, la descripción de éstos se verá en el diseño de las tablas del Sistema Administrativo-Contable de Pagos, debido a que el Área de Central de Pagos será la responsable del mantenimiento, los catálogos que se utilizarán son: (figura 3.5)

CATÁLOGO	OBSERVACIÓN
SC_BANCO	Tabla que contiene la descripción del banco.
SC_DEPTO	Tabla con la descripción de los departamentos del banco.
SC_PROV	Tabla con la descripción de los proveedores y/o empleados, que prestan un servicio al banco.
SC_CONCE	Tabla con la descripción de todos los posibles conceptos de pago.
SC_ITEM	Tabla con la descripción de ITEMS para la afectación contable de anticipos y comprobaciones de viajes y viáticos.
SC_CIUUDA	Tabla con la descripción de todas las ciudades en las que se pueden realizar viajes de trabajo.
SC_ZONAS	Tabla con la descripción de zonas para determinar la tarifa de viáticos.
SC_DIAIN	Tabla con la descripción de los días inhábiles.
SC_USERS	Tabla que contiene los usuarios autorizados para utilizar el sistema.

Figura 3.5 Catálogos de consulta utilizados por el Sistema para el Trámite de Pagos

Para garantizar que los usuarios tendrán siempre la última versión de los catálogos, se deberá implementar un procedimiento de actualización automática, el cual se controlará a través de la tabla UP\_FILES, siendo ésta: (figura 3.6)

CAMPO	OBSERVACIÓN
SISTEMA	Especifica el sistema que se actualizará.
ARCHIVO	Indica el archivo que se actualizará.
TIPO	Indica el tipo de archivo para actualizar, por ejemplo, DBF (tabla), APP (Archivo del sistema), etc.
ORIGEN	Indica la ubicación física de los archivos que serán actualizados.
DESTINO	Determina la ubicación física de donde se tomarán las actualizaciones de los archivos.
FEC_ACTUAL	Indica la fecha de última actualización.
HOR_ACTUAL	Indica la hora de la última actualización.
MODULO	Indica el nivel del módulo que será actualizado.
ACTUALIZAR	Valor que indica si es necesario actualizar los catálogos.

Figura 3.6 Tabla UP\_FILES

Finalmente, para controlar y determinar los campos que se capturarán dependiendo del tipo de gasto, se usará la tabla STES\_POL, cuyo diseño es: (figura 3.7)

CAMPO	OBSERVACIÓN
NOMBRE	Especifica el nombre del campo que se esta capturando o modificando.
ADMINISTRA	Determina si el campo puede ser capturado en tipo de gasto administrativo.
ESANTICIPO	Determina si el campo puede ser capturado cuando el tipo de gasto es un anticipo.
COMP	Determina si el campo puede ser capturado cuando el tipo de gasto es una comprobación.
BOLETO	Determina si el campo puede ser capturado cuando el tipo de gasto es una solicitud de boleto de avión.

Figura 3.7 Tabla STES\_POL

### 3.1.2 DISEÑO DE DATOS SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

Tomando como base los diagramas de Entidad-Relación y aplicando la normalización sobre las entidades definidas en el capítulo anterior, comenzaremos con el diseño de los datos necesarios para el buen funcionamiento del Sistema Administrativo-Contable de Pagos, iniciaremos por definir los catálogos que servirán para poder controlar la información que los usuarios capturen, y que a su vez, nos permita tener codificada la información relevante para facilitar la explotación en reportes y consultas de forma rápida y veraz.

Los catálogos a definir son los siguientes:

La tabla SC\_BANCO, contendrá los datos que identificarán a la entidad Banco, quedando: (figura 3.8)

CAMPO	OBSERVACIÓN
CODIGO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
NOMBRE	Nombre del banco.
RFC_BANCO	Registro Federal de Contribuyentes del banco.
DOMICILIO	Domicilio fiscal del banco.
CHEQUERA	Identifica el tipo de chequera que tiene el banco.
CTA_CARGO	Cuenta contable de cargo para la erogación de los pagos.
LOTE_CONTA	Lote contable que identifica el banco.
ID_CONTA	Identificador contable para diferenciar los archivos que son enviados a Contabilidad.
LOGO	Logotipo corto del banco.
LOGO_LARGO	Logotipo largo del banco.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.8 Tabla SC\_BANCO

La tabla SC\_REG, contendrá los datos que identificarán las diversas regiones con las que cuenta un banco, quedando: (figura 3.9)

CAMPO	OBSERVACIÓN
CVE_BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CODIGO	Clave que identifica la región. Se tomará como <b>Llave Principal</b> .
NOMBRE	Nombre de la región.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.9 Tabla SC\_REG

La tabla SC\_PLAZA, contendrá los datos que identificarán las diversas plazas con las que cuenta un banco, quedando: (figura 3.10)

CAMPO	OBSERVACIÓN
CVE_BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CVE_REGION	Clave que identifica la región. Se tomará como <b>Llave Principal</b> .
CODIGO	Clave que identifica la plaza. Se tomará como <b>Llave Principal</b> .
NOMBRE	Nombre de la plaza.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.10 Tabla SC\_PLAZA

La tabla SC\_DEPTO, contendrá los datos que identificarán las distintas áreas con las que cuenta un banco, quedando: (figura 3.11)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
REGION	Clave que identifica la región. Se tomará como <b>Llave Principal</b> .
PLAZA	Clave que identifica la plaza. Se tomará como <b>Llave Principal</b> .
CVE_DEPTO	Clave que identifica al departamento. Se tomará como <b>Llave Principal</b> .
NOM_DEPTO	Descripción del departamento.
UBICACIÓN	Ubicación física del departamento.
NOM_RESP	Nombre del funcionario responsable del departamento.
C_COSTOS	Centro de costos del departamento, a través de este dato se llevará el control contable de los gastos efectuados por los departamentos, los cuales son registrados en el Sistema Central de Contabilidad.
CC_IVA	Centro de costos en donde contablemente se registra el gasto del IVA.
CC_ISR	Centro de costos en donde contablemente se registra el gasto del ISR.
CC_RETIVA	Centro de costos en donde contablemente se registra el gasto de la Retención del IVA.
CC_CHEQUE	Centro de costos en donde contablemente se registra el gasto de las erogaciones de cheques.
CVE_TURNO	Clave que identifica a la persona que se hará cargo de la revisión de los documentos en el Área de Central de Pagos.
AUTORIPAGO	Identificador que se utilizará para determinar si el departamento tendrá facultades para autorizar pagos.
INI_FIRMA	Iniciales del funcionario que autorizará los pagos.
MONTO_MAX	Monto máximo que el departamento tiene para autorizar pagos.

CAP_FECVAL	Determina si el departamento puede generar movimientos con fecha valor.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.11 Tabla SC\_DEPTO

La tabla SC PROV, contendrá los datos de todos los proveedores y empleados que otorgan algún servicio al banco, quedando: (figura 3.12)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
PROVEEDOR	Clave que identifica al proveedor. Se tomará como <b>Llave Principal</b> .
NOM_PROV	Nombre del proveedor.
RFC	Registro Federal de Causantes del proveedor.
CURP	Clave Única de Registro de Población del proveedor.
CALLE	Calle y número del domicilio del proveedor.
COLONIA	Colonia del proveedor.
CIUDAD	Ciudad del proveedor.
COD_POSTAL	Código postal del proveedor.
TELÉFONO	Número telefónico del proveedor.
FAX	Número de fax del proveedor
NOM_COBRA1	Nombre de la primer persona autorizada para recoger los pagos.
NOM_COBRA2	Nombre de la segunda persona autorizada para recoger los pagos.
NOM_COBRA3	Nombre de la tercer persona autorizada para recoger los pagos.
TIPO_PROV	Identifica el tipo de proveedor, si es empleado o proveedor externo.

PAGAR_PROV	Identifica si la emisión de cheques se hará por contrarrecibos.
LEYENDA_PROV	Determina si se incluirá la leyenda "Para Abono" en la generación del cheque.
ISR	Determina si se permitirá la captura de ISR.
RETIVA	Determina si se permitirá la captura de la Retención del IVA.
DESCUENTO	Determina si se permitirá la captura de descuentos.
P_CHEQUERA	Identifica si la emisión de cheques se hará a través de la chequera del mismo banco o un tercero.
ITEM_PROV	Indica si el proveedor cuenta con un ítem contable para la afectación de sus gastos.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.12 Tabla SC\_PROV

La tabla SC CONCE, contendrá los datos que identificarán los diversos conceptos de pago definidos para un banco, quedando: (figura 3.13)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CONCEPTO	Clave que identifica el concepto de pago. Se tomará como <b>Llave Principal</b> .
DESCRIP	Descripción del concepto de pago.
VENTANILLA	Identifica si el concepto de pago es de uso exclusivo de la Ventanilla del Área de Central de Pagos.
TIENEORDEN	Identifica si el concepto utiliza cuentas de orden.
ITEM_CARGO	Identifica si el concepto es una cuenta de ítem para realizar la aplicación del cargo.
APLIABO_DE	Determina si la aplicación del abono se realizará de la factura o del detalle de movimientos.

ABONO_CAJA	Determina la cuenta contable en donde se realizará el abono del cheque.
ABONO_OTRO	Determina una cuenta contable alterna para realizar el abono del gasto.
ABONO_ORDE	Identifica si el abono del gasto se realizará en una cuenta de orden.
GASTO1	Determina la cuenta contable que se afectará cuando el gasto cuente con IVA e ISR.
GASTO2	Determina la cuenta contable que se afectará cuando el gasto cuente solo con IVA.
GASTO3	Determina la cuenta contable que se afectará cuando el gasto cuente solo con ISR.
GASTO4	Determina la cuenta contable que se afectará cuando el gasto no cuente con IVA, ni con ISR.
GASTO_ORDE	Determina la cuenta contable que se afectará cuando el gasto contenga cuentas de orden.
APLICA_IVA	Determina la cuenta contable donde se afectará el IVA.
APLI_REIVA	Determina la cuenta contable donde se afectará la retención del IVA.
APLICA_RET	Determina la cuenta contable donde se afectará el ISR.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

3.13 Tabla SC\_CONCE

La tabla SC\_CTAS, contendrá los datos que identificarán las diversas cuentas contables definidas para un banco, quedando: (figura 3.14)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CLAVE	Clave que identifica la cuenta contable. Se tomará como <b>Llave Principal</b> .

NOMBRE	Descripción de la cuenta contable.
ICBS_TIPO	Determina si es cuenta regular o ítem.
CUENTA	Cuenta mayor.
SUB1	Subcuenta1.
SUB2	Subcuenta2.
SUB3	Subcuenta3.
SECTOR	Sector contable de la cuenta.
TIPODEMOVI	Indica si la cuenta es de cargo o abono.
CARGA_ITEM	Determina si para esta cuenta se permitirá la carga automática de ítems.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.14 Tabla SC\_CTAS

La tabla SC\_ITEMS, contendrá los datos que identificarán los diversos ítems de las cuentas contables definidas para un banco, quedando: (figura 3.15)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CUENTA	Cuenta contable a la que pertenece el ítem. Se tomará como <b>Llave Principal</b> .
C_COSTOS	Identifica el centro de costos que afecta el ítem. Se tomará como <b>Llave Principal</b> .
CVE_MAYOR	Clave que identifica la cuenta de mayor al que pertenece el ítem.
ITEM	Número contable del ítem.
SERIE	Número de serie asociado al ítem.
SERIE_FIJA	Identifica si la asignación de la serie es fija o variable.
NOMBRE	Descripción del ítem.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.15 Tabla SC\_ITEMS

La tabla SC\_CENCO, contendrá los datos que identificarán todos los centros de costos definidos para un banco, quedando: (figura 3.16)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
C_COSTOS	Identifica el centro de costos. Se tomará como <b>Llave Principal</b> .
CC_NOMBRE	Descripción del centro de costos.
CC_NIVEL	Determina el nivel del centro de costos.
CC_NIVEL1	Determina al centro de costos a nivel de banco.
CC_NIVEL2	Determina al centro de costos a nivel de Coordinación General.
CC_NIVEL3	Determina al centro de costos a nivel de Coordinador.
CC_NIVEL4	Determina al centro de costos a nivel de Coordinador Adjunto o Director.
CC_NIVEL5	Determina al centro de costos a nivel de Titular.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.16 Tabla SC\_CONCE

La tabla SC\_TURNO, contendrá los datos para identificar a los responsables de revisión dentro del Área de Central de Pagos, quedando: (figura 3.17)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CVE_TURNO	Clave que identifica al responsable de revisión. Se tomará como <b>Llave Principal</b> .
NOM_TURNO	Nombre del responsable de revisión.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.17 Tabla SC\_TURNO

La tabla **SC CIUDA**, contendrá los datos de todas las ciudades definidas para un banco, quedando: (figura 3.18)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
CIUDAD	Clave que identifica la ciudad. Se tomará como <b>Llave Principal</b> .
NOM_CIUADAD	Nombre de la ciudad.
ZONA	Clave de la zona a la que pertenece la ciudad.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.18 Tabla SC\_CIUADA .

La tabla **SC ZONA**, contendrá los datos de todas las zonas definidas para un banco, funcionará principalmente para determinar las tarifas máximas cuando el gasto sea de viaje y viáticos, quedando: (figura 3.19)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
ZONA	Clave que identifica la zona. Se tomará como <b>Llave Principal</b> .
CATEGORÍA	Identifica si la categoría es de empleado, funcionario o externo.
TARIFA	Tarifa máxima autorizada para realizar gastos.
HOTEL	Tarifa máxima autorizada para gastos de hospedaje.
ALIMENTOS	Tarifa máxima autorizada para gastos de alimentación.
EXTRAS	Tarifa máxima autorizada para gastos extras.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.19 Tabla SC\_ZONA

La tabla SC\_DIAIN, contendrá los días inhábiles definidos para las Instituciones Bancarias, quedando: (figura 3.20)

CAMPO	OBSERVACIÓN
FECHA	Día inhábil. Se tomará como <b>Llave Principal</b> .
NOM_CIUADAD	Descripción del día inhábil.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.20 Tabla SC\_DIAIN

La tabla SC\_BITA, contendrá los movimientos que realice cualquier usuario que tenga acceso al sistema, quedando: (figura 3.21)

CAMPO	OBSERVACIÓN
BIT_USUFOL	Clave que identifica al usuario que acceso el sistema. Se tomará como <b>Llave Principal</b> .
BIT_MAQ	Identifica el equipo desde el cual el usuario tuvo acceso al sistema.
BIT_FECHA	Fecha de acceso al sistema.
BIT_HORA	Hora de acceso al sistema.
BIT_OPERA	Clave que identifica el tipo de operación realizada por el usuario.
BIT_OBSERV	Breve descripción de la actividad realizada por el usuario.

Figura 3.21 Tabla SC\_BITA

La tabla SC\_CATBI, contendrá los diferentes tipos de operaciones que puede realizar un usuario dentro del sistema, quedando: (figura 3.22)

CAMPO	OBSERVACIÓN
CLAVE	Clave que identifica la operación realizada en el sistema. Se tomará como <b>Llave Principal</b> .
OPERACIÓN	Descripción de la operación realizada.

Figura 3.22 Tabla SC\_CATBI

La tabla SC\_USERS, contendrá los datos de todos los usuarios con acceso al sistema, quedando: (figura 3.23)

CAMPO	OBSERVACIÓN
USU_FOLIO	Clave única asignada automáticamente para identificar al usuario del sistema. Se tomará como <b>Llave Principal</b> .
USU_BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
USU_NOMINA	Número de empleado del usuario.
USU_USER	Clave de usuario.
USU_LOGIN	Contraseña del usuario.
USU_DEPTO	Departamento de adscripción del usuario.
USU_ACCESO	Nivel de acceso del usuario al sistema.
USU_NOMBRE	Nombre del usuario.
USU_UBICA	Ubicación física del usuario.
USU_TEL	Teléfono de localización del usuario.
USU_ACTIVADO	Identifica si el usuario esta utilizando el sistema.
FEC_ALTA	Fecha de alta del registro.
FEC_BAJA	Fecha de baja del registro.

Figura 3.23 Tabla SC\_USERS

Para garantizar que los usuarios tendrán siempre la última versión del sistema, se deberá implementar un procedimiento de actualización automática, la cual se controlará a través de la tabla SC\_FILES, quedando: (figura 3.24)

CAMPO	OBSERVACIÓN
SISTEMA	Especifica el sistema que se actualizará.
ARCHIVO	Indica el archivo que se actualizará.
TIPO	Indica el tipo de archivo para actualizar, por ejemplo, DBF (tabla), APP (Archivo del sistema), etc.
ORIGEN	Indica la ubicación física de los archivos que serán actualizados.
DESTINO	Determina la ubicación física de donde se tomarán las actualizaciones de los archivos.
FEC_ACTUAL	Indica la fecha de última actualización.
HOR_ACTUAL	Indica la hora de la última actualización.
MODULO	Indica el nivel del módulo que será actualizado.
ACTUALIZAR	Valor que indica si es necesario actualizar los catálogos.

Figura 3.24 Tabla SC\_FILES

Considerando las entidades que intervienen para el correcto funcionamiento de la **Ventanilla**, las cuales fueron descritas en el modelo entidad-relación, podemos determinar el diseño de los datos, quedando como se menciona a continuación.

La tabla **SC\_CONCA**, contendrá los datos de los contra-recibos entregados por la Ventanilla, quedando: (figura 3.25)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica el banco. Se tomará como <b>Llave Principal</b> .
NO_CONRE	Número de contra-recibo. Se tomará como <b>Llave Principal</b> .
NO_POLIZA	Número de referencia que identifica la factura.
CVE_DEPTO	Clave del departamento responsable de presentar el gasto en Ventanilla.
PROVEEDOR	Clave del proveedor beneficiario del pago.

TIPO_GASTO	Tipo de Gasto al que se refiere la factura: Administrativo, Anticipo de Viáticos, Comprobación de Viáticos, Boleto de Avión.
CONCEPTO	Clave que identifica el concepto de pago del contra-recibo.
FORMA_PAGO	Forma de pago del contra-recibo: cheque o abono a cuenta.
TOTAL	Importe total del contra-recibo a pagar.
SALDO_COMP	Importe que registra el saldo de las comprobaciones.
ANTICIPO	Importe que registra el anticipo otorgado.
COMENTARIO	Descripción breve del concepto de pago del contra-recibo.
FEC_ALTA	Fecha de registro del contra-recibo.
HORA_ALTA	Hora de registro del contra-recibo.
FEC_PAGO	Fecha programada de pago del contra-recibo.
ESTATUS	Situación que tiene el contra-recibo dentro del proceso de generación para su pago.
CVE_TURNO	Clave del responsable de revisión del contra-recibo.
VALIDO	Clave del usuario que autorizo el pago del contra-recibo.
FEC_CIERRE	Fecha Contable de pago del contra-recibo.
NO_CONTABI	Número de contabilización en el que se registro el pago.
ID_CIERRE	Número de identificación del contra-recibo en el proceso contable.
BLOQUEADO	Bandera que indica si el registro esta en uso.
LOTE_ENVIO	Identifica de que lote proceden las facturas que conforman el contra-recibo.
FEC_VALOR	Identifica si el contra-recibo se registrará en contabilidad con una fecha distinta al resto de los movimientos.

Figura 3.25 Tabla SC\_CONCA

La tabla SC\_POLCA será similar a la tabla ST\_POLCA, pero con la diferencia de que no solo almacenará la información de un departamento, sino que contendrá el histórico de todas las facturas recibidas por la Ventanilla, quedando: (figura 3.26)

CAMPO	OBSERVACIÓN
NO_CONRE	Número de contra-recibo. Se tomará como <b>Llave Principal</b> .
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
BANCO	Clave del Banco al que pertenece la factura.
CVE_DEPTO	Clave del departamento que recibe la factura para tramitar su pago.
PROVEEDOR	Clave del proveedor que solicita el pago de un servicio.
NO_DOCTO	Número de Factura.
FEC_DOCTO	Fecha de la Factura.
TIPO_GASTO	Tipo de Gasto al que se refiere la factura, Administrativo, Anticipo de Viáticos, Comprobación de Viáticos, Boleto de Avión.
VIATICO	Identifica si el pago solicitado corresponde a viáticos.
DETALLEDOC	Identifica si la factura se desglosará en varios movimientos.
ITEM	Campo que contiene la cuenta contable para la comprobación de un anticipo.
SERIE	Campo que identifica de forma individual el documento de la comprobación de un anticipo.
CONCEPTO	Clave que identifica el concepto de pago de la factura.
DEPTOGASTO	Clave del departamento que origina el gasto.
FORMA_PAGO	Campo que identifica la forma de pago de la factura, ya sea cheque o abono a cuenta de cheques.
IMPORTE	Importe de la Factura.
DESCUENTO	Descuento aplicado.
IVA	Importe del IVA aplicado.
RET_IVA	Retención de IVA aplicado.
RET_ISR	Retención de ISR aplicado.
TOTAL	Importe Total de la Factura a pagar.

COMENT1	Comentarios que describen de forma detallada el concepto de pago.
COMENT2	Complemento de comentarios.
ANTICIPO	Importe que registra el anticipo otorgado.
SALDO_COMP	Importe que registra el saldo de las comprobaciones.
ESTATUS	Situación que tiene una factura dentro del proceso de generación de su pago.
CVE_TURN0	Clave que identifica a que persona se turno para su revisión.
FEC_POLIZA	Fecha que indica cuando se autorizo el pago de la factura.
NO_POLIZA	Número de referencia asignado cuando de autoriza la factura.
FEC_ENVIO	Fecha de envío de la factura al Área de Central de Pagos.
LOTE_ENVIO	Lote de envío al Área de Central de Pagos.
DEPTO_USU	Clave del Departamento del Usuario que captura la factura.
FEC_ALTA	Fecha de alta de la factura en el sistema.
FEC_VALOR	Indica la fecha del registro contable de los gastos originados por la factura, esto sucede cuando el pago se realiza fuera de tiempo.

Figura 3.26 Tabla SC\_POLCA

La tabla SC\_MOVCA será similar a la tabla ST\_MOVCA, pero con la diferencia de que no solo almacenará la información de un departamento, sino que contendrá el histórico del detalle de todas las facturas recibidas por la Ventanilla, quedando: (figura 3.27)

CAMPO	OBSERVACIÓN
NO_CONRE	Número de contra-recibo. Se tomará como <b>Llave Principal</b> .
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_DETALLE	Campo único que identifica cada detalle de la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
DEPTOGASTO	Clave del departamento que origina el gasto del detalle de la factura.
NO_DOCTO	Número de documento que forma parte de la Factura.
FEC_DOCTO	Fecha del documento.
CONCEPTO	Clave que identifica el concepto de pago de cada documento de la factura.
IMPORTE	Importe del documento.
POR_DESC	Porcentaje del descuento por aplicar.
DESCUENTO	Importe del Descuento aplicado.
POR_IVA	Porcentaje del IVA por aplicar.
IVA	Importe del IVA aplicado.
POR_RETIVA	Porcentaje de la Retención del IVA por aplicar.
RET_IVA	Importe de la Retención de IVA aplicado.
POR_RETISR	Porcentaje de la Retención del ISR por aplicar.
RET_ISR	Retención de ISR aplicado.
TOTAL	Importe Total del documento.

Figura 3.27 Tabla SC\_MOVCA

La tabla **SC VIAJE** será similar a la tabla ST\_VIAJE, pero con la diferencia de que no solo almacenará la información de un departamento, sino que contendrá el histórico del detalle de todos los movimientos de viaje y viáticos recibidos por la Ventanilla, quedando: (figura 3.28)

CAMPO	OBSERVACIÓN
NO_CONRE	Número de contra-recibo. Se tomará como <b>Llave Principal</b> .
ID_POLIZA	Campo único que identifica la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_DETALLE	Campo único que identifica cada detalle de la factura, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
ID_VIAJE	Campo único que identifica el detalle de las ciudades que forman parte del viaje, se asignará de forma automática para garantizar la integridad de la información. Se tomará como <b>Llave Principal</b> .
CI_SALIDA	Ciudad Inicial (CI) de donde se origina el viaje.
CI_DESTINO	Ciudad destino del viaje.
CI_DIAS	Días del viaje.
CI_FECHA	Fecha de inicio del viaje.
CI_HORA	Hora de inicio del viaje.
CI_BOLETO	Identifica si se requiere boleto de avión.
CF_SALIDA	Ciudad Final (CF) en donde se concluirá el viaje.
CF_DESTINO	Ciudad de regreso del viaje.
CF_FECHA	Fecha de regreso del viaje.
CF_HORA	Hora de regreso del viaje.
CF_BOLETO	Identifica si se requiere boleto de avión para el regreso del viaje.
FINVIAJE	Identifica si este registro concluye la captura del viaje.
CIUDAD	Ciudad de la cual se esta realizando la comprobación de viáticos.
A_REALES	Días reales del viaje.
A_FECINI	Fecha real de inicio del viaje
A_FECFIN	Fecha real de finalización del viaje.

Figura 3.28 Tabla SC\_VIAJE

Por último, en lo referente a las entidades que intervienen en las **Áreas de Emisión de Pagos y Contable**, ambas concurren en algunas entidades por lo que el diseño de éstas se realizará conjuntamente tomando en consideración los modelos entidad-relación definidos anteriormente, quedando como se mencionan a continuación:

La tabla **SC\_CONTA**, almacenará la información concerniente a las emisiones de pagos y contabilizaciones efectuadas por el Área de Central de Pagos, cabe hacer mención que la emisión de pagos podrá efectuarse más de una vez al día y la generación de póliza contable solamente se efectuará una vez al día por banco, quedando: (figura 3.29)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica las diferentes emisiones de pagos que se efectúan en una misma fecha. Se tomará como <b>Llave Principal</b> .
LIBERACHEQ	Valor lógico que indica si fue efectuada correctamente la impresión de cheques.
FEC_POLIZA	Fecha en la cual quedarán registrados los pagos contablemente en el Sistema Central de Contabilidad.
FEC_EXPOR	Fecha en la que se efectúa la exportación de los movimientos al Sistema Central de Contabilidad.
FEC_VALOR	Fecha que indica si los movimientos se registrarán contablemente en una fecha distinta a la definida por la póliza.
REGION	Identifica si la emisión de pagos fue realizada a nivel regional.
PLAZA	Identifica si la emisión de pagos fue realizada a nivel plaza.
CVE_TURN0	Clave de la persona que efectuó la emisión de pago.

FORMA_PAGO	Identifica si la emisión de pagos, se realizó vía emisión de cheques o abono a cuenta.
CONTRA_PAG	Número de contra-recibos que fueron considerados en la emisión de pago.
NUM_CHEQUE	Número de cheques que se realizaron.
ABO_CHEQUE	Monto de pago que se realizó a través de emisión de cheques.
NUM_ABONOS	Número de abonos a cuenta que se realizaron.
ABO_CTACHE	Monto de pago que se realizó a través de abonos a cuenta.

Figura 3.29 Tabla SC\_CONTA

La tabla **SC\_CHEQ**, es la entidad en la cual se almacenará y controlará la información concerniente a los cheques emitidos, quedando: (figura 3.30)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica las diferentes emisiones de pagos que se efectúan en una misma fecha. Se tomará como <b>Llave Principal</b> .
ID_CIERRE	Número que identifica el cheque dentro de la emisión de pago. Se tomará como <b>Llave Principal</b> .
PROVEEDOR	Clave del beneficiario del cheque.
PAGAR_PROV	Identifica si para pagarle al proveedor se agruparán todos sus contra-recibos o se pagará por cada uno de ellos.
NOMBRE	Nombre que físicamente se imprimirá en el cheque.
NO_CHEQUE	Número que físicamente tiene el cheque emitido.
IMPORTE	Importe del cheque emitido.

FEC_ENTREG	Fecha en que se realiza la entrega del cheque.
RECIBIO	Nombre de la persona que recibe en cheque.
GENERO	Clave del usuario que genero el cheque.
ENTREGO	Clave del usuario que entrego el cheque.

Figura 3.30 Tabla SC\_CHEQ

La tabla SC\_ABONO, es la entidad en la cual se almacenará y controlará la información concerniente a los abonos a cuenta, quedando: (figura 3.31)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica los diferentes abonos a cuenta que se efectúan en una misma fecha. Se tomará como <b>Llave Principal</b> .
ID_CIERRE	Número que identifica el abono a cuenta dentro de la emisión de pago. Se tomará como <b>Llave Principal</b> .
PROVEEDOR	Clave del beneficiario del abono a cuenta.
NOMBRE	Nombre del titular de la cuenta a la cual se abonará.
IMPORTE	Importe del abono a cuenta emitido.
CHEQUERA	Identifica si la chequera en la que se realizará el cargo es propia o de un tercero.
CTA_CARGO	Cuenta de cheques en la que se efectuará el cargo:
GENERO	Clave del usuario que genero el abono a cuenta.

Figura 3.31 Tabla SC\_ABONO

La tabla SC\_CARAB, es la entidad que almacenará el detalle del movimiento contable originado por la emisión de cheques o abonos a cuentas, quedando: (figura 3.32)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica las diferentes emisiones de pagos efectuadas en una misma fecha. Se tomará como <b>Llave Principal</b> .
ID_CIERRE	Número que identifica el movimiento contable dentro de la emisión de pago. Se tomará como <b>Llave Principal</b> .
CVE_DEPTO	Departamento al que contablemente se aplicará el cargo o abono.
DESC_CTA	Descripción de la cuenta contable que se afectará.
ICBS_TIPO	Indica si la cuenta que se afectará es de control (item) o regular.
TIPODEMOVI	Identifica si el movimiento es de cargo o abono.
ICBS_CTA	Cuenta contable que se afectará.
CTA_MAYOR	Cuenta mayor a la que pertenece la cuenta contable que se esta afectando.
ICBS_SERIE	Número de serie que identifica el movimiento cuando se trata de una cuenta de control (item).
IMPORTE	Importe que se esta afectando en la cuenta contable.
NO_CHEQUE	Número de cheque que físicamente fue entregado.
C_COSTOS	Centro de costos que se afecta con la aplicación de la cuenta contable.

Figura 3.32 Tabla SC\_CARAB

La tabla SC\_ERROR, únicamente se utilizará cuando se realicen emisiones de pago o contabilizaciones de movimientos y registrará los errores generados durante éstos proceso, quedando: (figura 3.33)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica las diferentes emisiones de pagos efectuadas en una misma fecha. Se tomará como <b>Llave Principal</b> .
NO_CONRE	Número de contra-recibo en donde se detecto el error.
PROVEEDOR	Clave del proveedor.
NOM_PROV	Nombre del proveedor.
CONCEPTO	Clave del concepto de pago.
DESC_CONCE	Descripción del concepto de pago.
IMPORTE	Importe de pago.
ERROR	Motivo del error.

Figura 3.33 Tabla SC\_ERROR

La tabla **SC CHEQC**, contendrá la información de los cheques cancelados, con el fin de poder llevar un control al momento de realizar la conciliación de cheques emitidos, quedando: (figura 3.34)

CAMPO	OBSERVACIÓN
BANCO	Clave que identifica al banco. Se tomará como <b>Llave Principal</b> .
FEC_CIERRE	Fecha de emisión de pagos en la que se realizó la cancelación del cheque. Se tomará como <b>Llave Principal</b> .
NO_CONTABI	Número consecutivo que identifica la emisión de pago en la que se realizó la cancelación del cheque. Se tomará como <b>Llave Principal</b> .
NOMBRE	Nombre que físicamente se imprimió en el cheque.
NO_CHEQUE	Número que físicamente tiene el cheque cancelado.

IMPORTE	Importe del cheque cancelado.
MOTIVO	Motivo de cancelación del cheque.
CANCELO	Clave del usuario que cancelo el cheque.

Figura 3.34 Tabla SC\_CHEQC

Finalmente, la última tabla por definir será SC\_EXPOR, la cual no almacenará datos, solamente se utilizará para dar formato al archivo de transferencia de la póliza contable que será enviada al Sistema Central de Contabilidad, quedando: (figura 3.35)

CAMPO	OBSERVACIÓN
MONEDA	Indica si fue efectuado el pago en moneda nacional o moneda extranjera.
ICBS_CTA	Indica la cuenta contable que se afectó.
C_COSTOS	Centro de costos que se afecta con la aplicación de la cuenta contable.
ICBS_SERIE	Número de serie que identifica el movimiento cuando se trata de una cuenta de control.
FEC_POLIZA	Fecha contable en la cual quedaran registrados los pagos en el Sistema Central de Contabilidad.
TIPODEMOVI	Identifica si el movimiento es de cargo o de abono.
IMPORTE_MN	Importe en moneda nacional.
TIP_CAMBIO	Tipo de cambio, si el pago fue en moneda extranjera.
IMPORTE_EX	Importe en moneda extranjera.
DESC_CTA	Descripción de la cuenta contable.
IDENTIFICA	Identificador de archivo, utilizado por el Sistema Central de Contabilidad.

Figura 3.35 Tabla SC\_EXPOR

### 3.2 MÓDULOS DEL SISTEMA

De acuerdo al análisis establecido, a la definición, a la propuesta de solución del problema y siguiendo la estrategia planteada, es necesario iniciar con el diseño del sistema, a través del cual se establecerán todos los módulos que lo conformarán.

#### 3.2.1 MÓDULOS DEL SISTEMA PARA EL SISTEMA TRÁMITE DE PAGOS

El Sistema para el Trámite de Pagos que será utilizado por los departamentos que solicitan un pago constará de cuatro módulos principales, descritos en la figura 3.36:

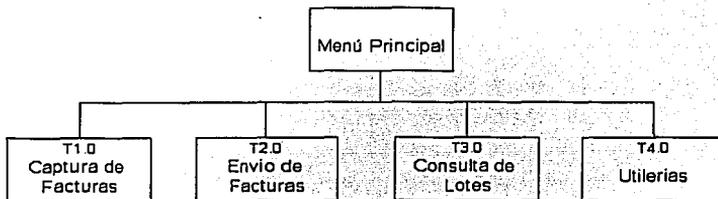


Figura 3.36 Menú Principal del Sistema para el Trámite de Pagos

El primer módulo del sistema "T1.0 Captura de Facturas", deberá ser capaz de abarcar todas las posibles situaciones de una solicitud de un pago, por lo que quedará integrado de la siguiente manera: (figura 3.37)

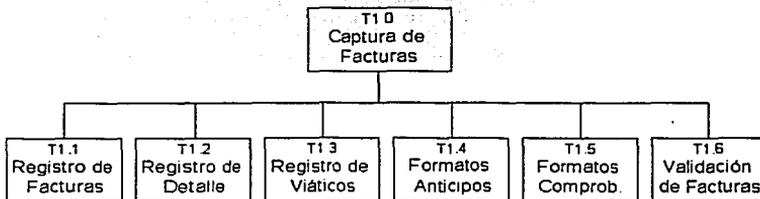


Figura 3.37 Módulo T1.0 Captura de Facturas

Ahora bien, será necesario desglosar cada una de las partes que conforman este primer módulo, por lo tanto comenzaremos con el "T1.1 Registro de Facturas", parte fundamental de la captura, a través del cual se permitirán registrar altas, bajas y cambios de facturas entregadas por los proveedores, así como contar con una opción de búsqueda de información dentro del sistema, quedando de la siguiente forma: (figura 3.38)

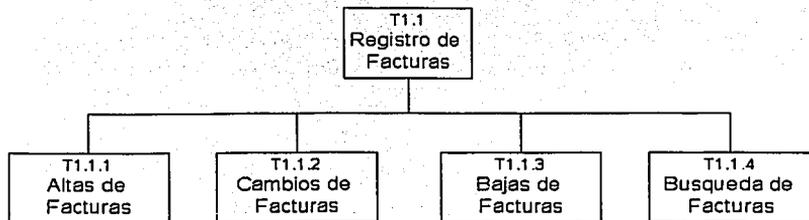


Figura 3.38 Apartado T1.1 Registro de Facturas

El apartado "T1.2 Registro de Detalle" será el elemento a través del cual se registrará el detalle con el que cuenta cada una de las facturas entregadas, permitiendo para tal efecto el registro de altas, bajas y cambios, así como búsqueda de movimientos, quedando definido de la siguiente forma: (figura 3.39)

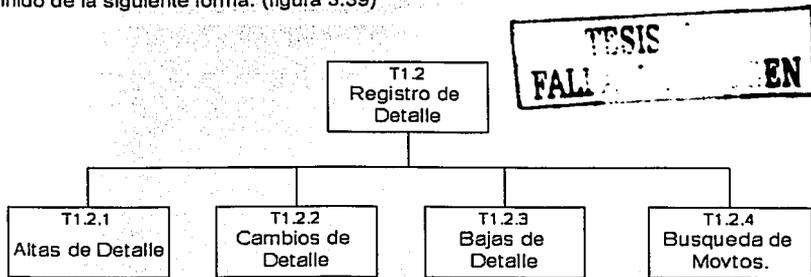


Figura 3.39 Apartado T1.2 Registro de Detalle

Cuando se trate de solicitudes de viaje y viáticos de empleados será necesario contar con un apartado denominado "T1.3 Registro de Viáticos", en el cual se permita registrar altas, bajas y cambios de este tipo de movimientos, y a su vez contar con la posibilidad de búsqueda de información previamente capturada, quedando definido de la siguiente forma: (figura 3.40)

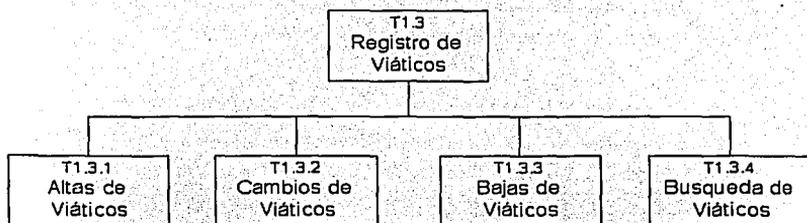


Figura 3.40 Apartado T1.3 Registro de Viáticos

Al momento del registro de un viático será necesario homologar los formatos de solicitud necesarios para tal efecto, por lo tanto el apartado "T1.4 Formatos Anticipos" contará con los dos posibles formatos, uno para la solicitud del gasto y otro para la solicitud del transporte aéreo, quedando definido de la siguiente forma: (figura 3.41)

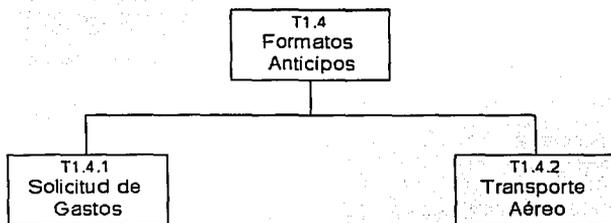


Figura 3.41 Apartado T1.4 Formatos Anticipos

En lo que se refiere a la comprobación de un gasto producto de un anticipo de viáticos, se deberá contar de igual manera con formatos únicos, los cuales quedarán definidos en el apartado "T1.5 Formatos Comprob.", consistentes en una carátula de comprobación y en un detalle de la misma, quedando de la siguiente forma: (figura 3.42)

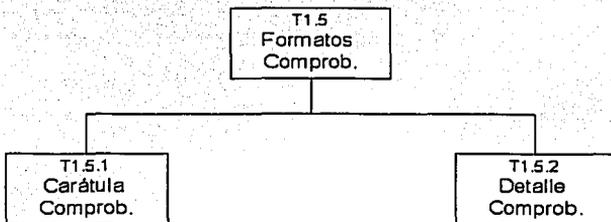


Figura 3.42 Apartado T1.5 Formatos Comprob.

Para concluir con el primer módulo del Sistema para el Trámite de Pagos, tenemos el apartado "T1.6 Validación de Facturas", a través del cual se podrán verificar los importes capturados y proporcionará la opción de autorización de facturas para el envío a firma del funcionario facultado, quedando definido de la siguiente forma: (figura 3.43)

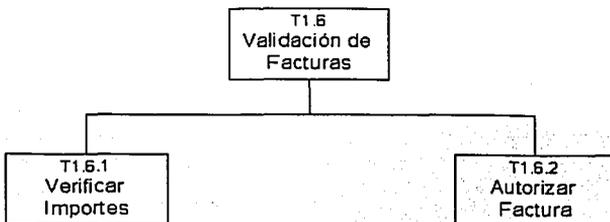


Figura 3.43 Apartado T1.6 Validación de Facturas

El segundo módulo del sistema "T2.0 Envío de Facturas", deberá ser capaz de permitir consultar las facturas autorizadas por el funcionario facultado, así como permitir la cancelación de éstas, por otra parte se deberá permitir asignar el lote de identificación y el envío de facturas al Área de Central de Pagos, la figura 3.44 ilustra esta situación:

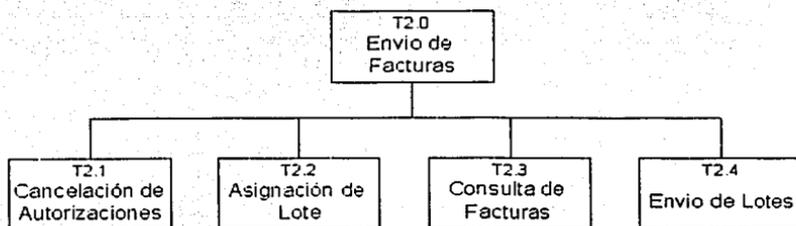


Figura 3.44 Módulo T2.0 Envío de Facturas

En lo referente al apartado "T2.2 Asignación de Lote", éste deberá ser capaz de asignar el lote de identificación de la factura, así como permitir la impresión previa o definitiva de la solicitud de pago, misma que será enviada al Área de Central de Pagos, en sustitución de las pólizas contables, gráficamente tenemos: (figura 3.45)



Figura 3.45 Apartado T2.2 Asignación de Lote

Para concluir con el segundo módulo del Sistema para el Trámite de Pagos, el apartado "T2.4 Envío de Lotes", será el encargado de seleccionar los lotes que se enviarán al Área de Central de Pagos; además, permitirá generar físicamente un archivo en formato ASCII que contendrá el detalle de las facturas enviadas, lo anterior con el fin de evitar la recaptura de información, gráficamente tenemos: (figura 3.46)

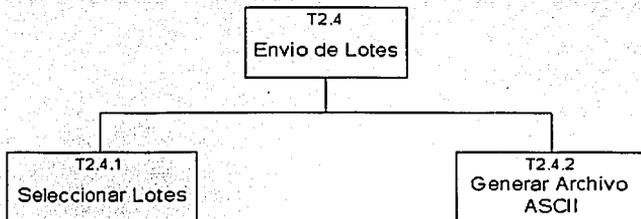


Figura 3.46 Apartado T2.4 Envío de Lotes

El tercer módulo del Sistema "T3.0 Consulta de Lotes", deberá permitir consultar el detalle de los lotes enviados; contará con la posibilidad de reenvío de lotes; tendrá la facilidad de generar un reporte con el detalle de los lotes enviados; proporcionará los elementos necesarios para recuperar los errores detectados por el Área de Central de Pagos, gráficamente tenemos: (figura 3.47)

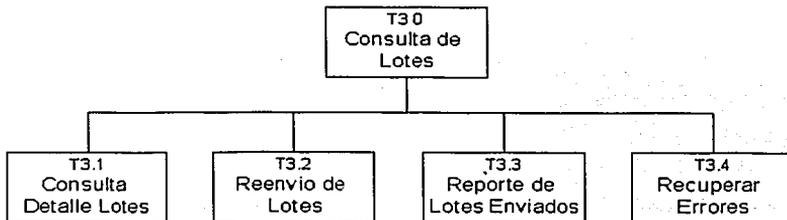


Figura 3.47 Módulo T3.0 Consulta de Lotes

El último módulo del Sistema "T4.0 Utilerías", proporcionará los elementos necesarios para su correcto funcionamiento, a través del cual se integrarán las herramientas de mantenimiento de la base de datos, definición de los parámetros del sistema y una serie de reportes útiles para el usuario; gráficamente tenemos: (figura 3.48)

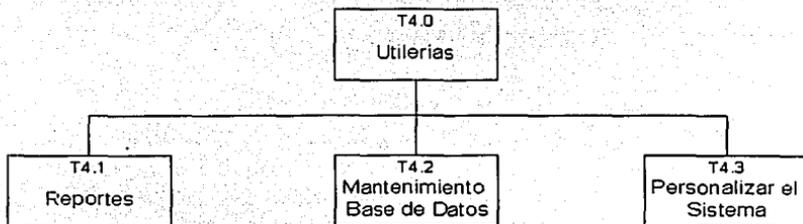


Figura 3.48 Módulo T4.0 Utilerías

En lo referente al apartado "T4.1 Reporte", éste contendrá una serie de reportes que le indicarán al usuario que facturas y anticipos de viáticos pendientes se encuentran registrados en el sistema, así como, permitirle la impresión de los formatos únicos para los gastos de viaje y viáticos, gráficamente tenemos: (figura 3.49)

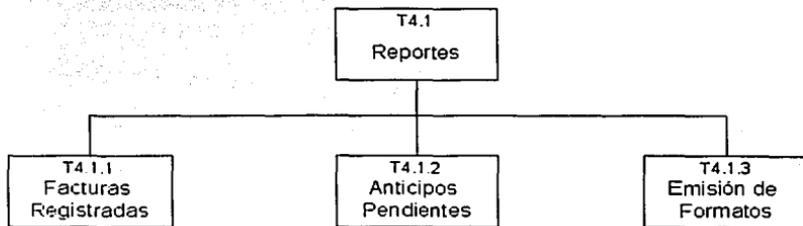


Figura 3.49 Apartado T4.1 Reportes

El apartado "T4.1.3 Emisión de Formatos", permitirá mandar a imprimir los formatos de solicitud de anticipos, comprobación de gastos y relación de pasajes, gráficamente tenemos: (figura 3.50)

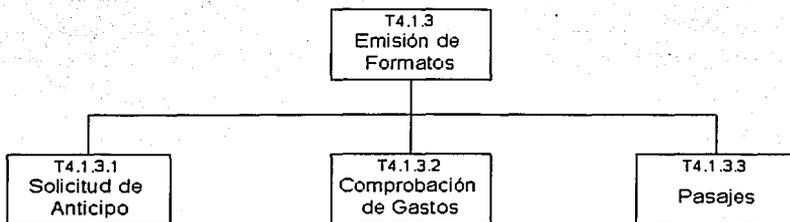


Figura 3.50 Apartado T4.1.3 Emisión de Formatos

El apartado "T4.2 Mantenimiento Base de Datos", proporcionará las herramientas necesarias para darle mantenimiento a los datos que conforman el sistema; regenerará los índices de las tablas; permitirá respaldar y restaurar la información de éstas, lo anterior con la finalidad de garantizar la integridad de la información contenida en el sistema, gráficamente tenemos: (figura 3.51)

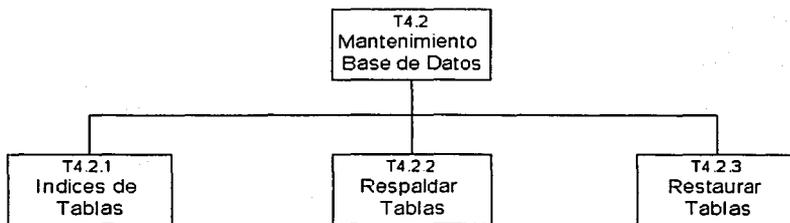


Figura 3.51 Apartado T4.2 Mantenimiento Base de Datos

El último apartado del módulo de Utilerías, es el denominado "T4.3 Personalizar Sistema", el cual contendrá los parámetros del sistema, tales como: ruta de bases de datos, ruta de envío de información, leyendas y logotipos de reportes, etc.; opción para que el usuario pueda cambiar su contraseña de acceso y finalmente información del sistema para que los usuarios puedan contactar al administrador del sistema en caso de cualquier error, gráficamente tenemos: (figura 3.52)

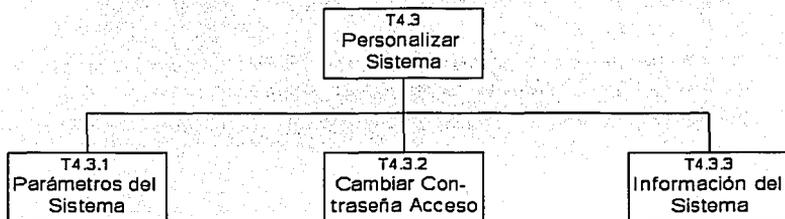


Figura 3.52 Apartado T4.3 Personalizar Sistema

### 3.2.2 MÓDULOS DEL SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

El Sistema Administrativo-Contable de Pagos será utilizado por el personal del Área de Central de Pagos, y tendrá como finalidad recibir solicitudes de pago, emitir contra-recibos, realizar emisión de pagos y efectuar la afectación contable de los pagos realizados; por otra parte, contará con una opción de mantenimiento a los catálogos utilizados por el sistema y utilerías para el correcto funcionamiento del mismo. Todo lo descrito anteriormente será agrupado en 6 principales módulos, los cuales se muestran en la figura 3.53:

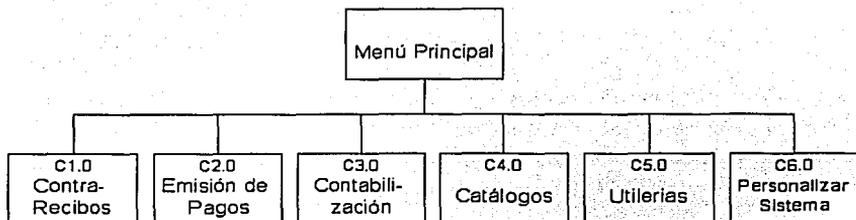


Figura 3.53 Menú Principal del Sistema Administrativo-Contable de Pagos

El primer módulo del sistema "C1.0 Contra-Recibos", deberá ser capaz de manipular los contra-recibos que son emitidos por el área de Ventanilla; permitirá la revisión de información y generará un previo contable de contra-recibos, por lo que quedará integrado de la siguiente forma: (figura 3.54)

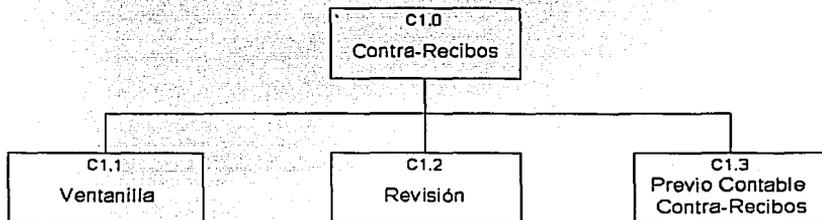


Figura 3.54 Módulo C1.0 Contra-Recibos

Ahora bien, será necesario desglosar cada una de las partes que conforman este primer módulo, por lo tanto comenzaremos con el apartado "C1.1 Ventanilla", éste será la parte medular de la Ventanilla, pues será aquí, en donde se registrarán y manipularán los contra-recibos emitidos, incluyendo la opción de poder autorizar o rechazar pagos, complementando estas actividades con la impresión de un reporte de contra-recibos emitidos, gráficamente tenemos: (figura 3.55)

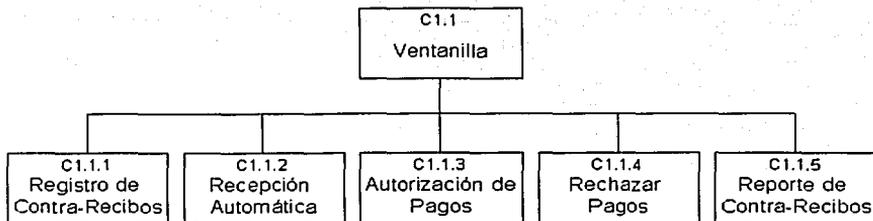


Figura 3.55 Apartado C1.1 Ventanilla

A través del apartado "C1.1.1 Registro de Contra-Recibos" se permitirán registrar altas, bajas y cambios de contra-recibos, así como contar con una opción de búsqueda de información dentro del sistema, quedando de la siguiente forma: (figura 3.56)

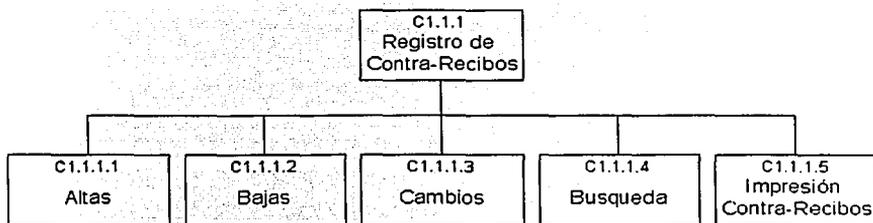


Figura 3.56 Apartado C1.1.1 Registro de Contra-Recibos

Para facilitar el registro de contra-recibos y evitar la recaptura de información de las facturas, se contará con el apartado "C1.1.2 Recepción Automática", mediante el cual se seleccionarán los lotes enviados a Ventanilla; se podrá verificar de forma automática la información recibida y si es correcta, se procederá a realizar la impresión de los contra-recibos correspondientes, gráficamente se tiene: (figura 3.57)

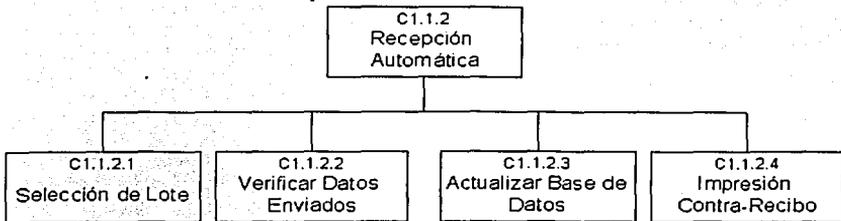


Figura 3.57 Apartado C1.1.2 Recepción Automática

El apartado "C1.2 Revisión", será el apoyo de la información recibida para los responsables de revisión, a través del cual, podrán verificar los datos y los importes de las facturas, tanto de gastos, como de viáticos, quedando de la siguiente forma: (figura 3.58)

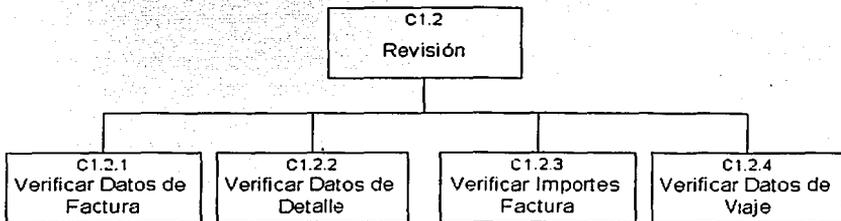
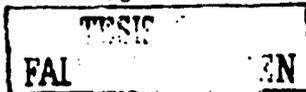


Figura 3.58 Apartado C1.2 Revisión

El último apartado del primer módulo del Sistema Administrativo-Contable de Pagos, definido como "C1.3 Previo Contable Contra-Recibos", realizará la generación de un previo contable, para garantizar que los movimientos que se van a procesar están correctos en cuanto a la definición del concepto de pago y a su afectación contable, así mismo, se podrá conocer con anticipación si existen errores en éstos, gráficamente tenemos: (figura 3.59)



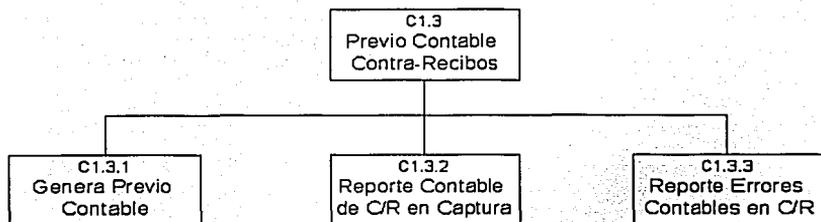


Figura 3.59 Apartado C1.3 Previo Contable Contra-Recibos

El segundo módulo del Sistema "C2.0 Emisión de Pagos", realizará la emisión de cheques; el previo contable de la emisión efectuada, se tendrá el control de la entrega de éstos, así como la impresión de reportes históricos, gráficamente tenemos: (figura 3.60)

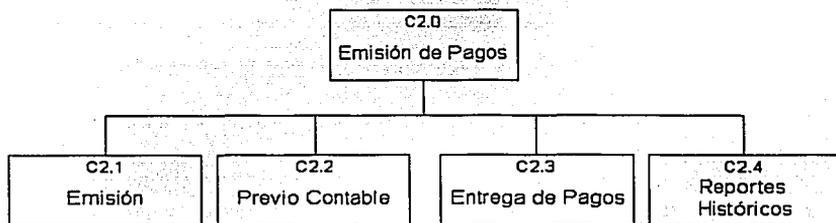


Figura 3.60 Módulo C2.0 Emisión de Pagos

Desglosando el módulo de Emisión de Pagos, tenemos el apartado "C2.1 Emisión", que realizará la selección de los contra-recibos a pagar; efectuará la impresión física de los cheques, así como la cancelación de los mismos en caso de haber errores; generará los abonos a las cuentas de cheques, la siguiente figura muestra lo descrito anteriormente: (figura 3.61)

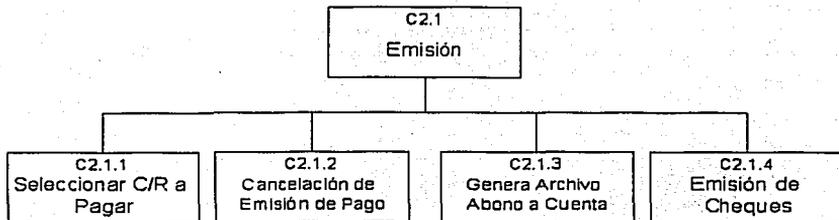


Figura 3.61 Apartado C2.1 Emisión

El apartado "C2.2 Previo Contable", permitirá contabilizar de manera previa los cheques o abonos generados; contará con la posibilidad de realizar la liberación de los pagos para su posterior afectación contable, gráficamente tenemos: (figura 3.62)

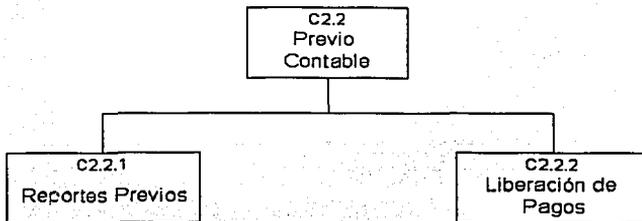
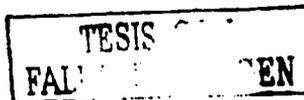


Figura 3.62 Apartado C2.2 Previo Contable

Los reportes previos a los que hace referencia el apartado anterior, serán descritos en "C2.2.1 Reportes Previos", los cuales conformarán el soporte de los pagos emitidos, así como los documentos soporte para solicitar los fondos necesarios al Área de Tesorería, gráficamente tenemos: (figura 3.63)



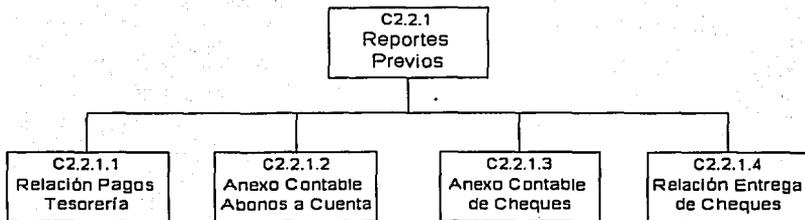


Figura 3.63 Apartado C2.2.1 Reportes Previos

Retomando el módulo "C2.0 Emisión de Pagos", tenemos el apartado "C2.3 Entrega de Pagos", a través del cual se realizará la búsqueda y entrega de cheques a los beneficiarios de un pago, realizando para tal efecto un control de entrega de cheques, gráficamente tenemos: (figura 3.64)

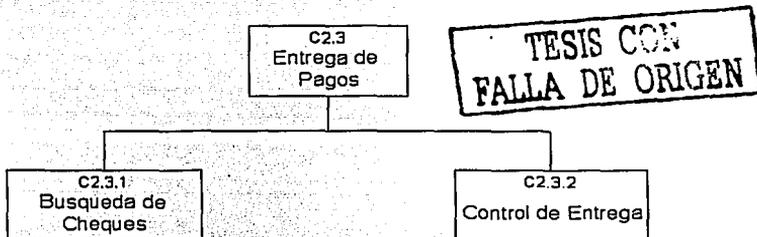


Figura 3.64 Apartado C2.3 Entrega de Pagos

El último apartado del módulo "C2.0 Emisión de Pagos", es el denominado "C2.4 Reportes Históricos", a través del cual se podrá realizar la emisión de reportes, estando clasificados en contables y en control interno, gráficamente tenemos: (figura 3.65)

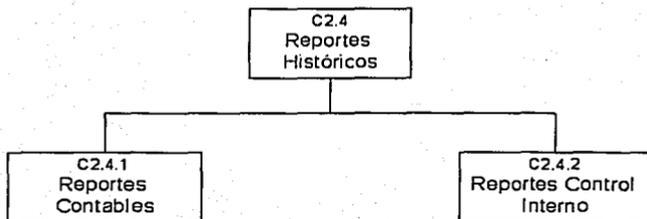


Figura 3.65 Apartado C2.4 Reportes Históricos

El apartado "C2.4.1 Reportes Contables", contendrá los informes contables que sirvieron de soporte para la emisión de un pago, gráficamente tenemos: (figura 3.66)

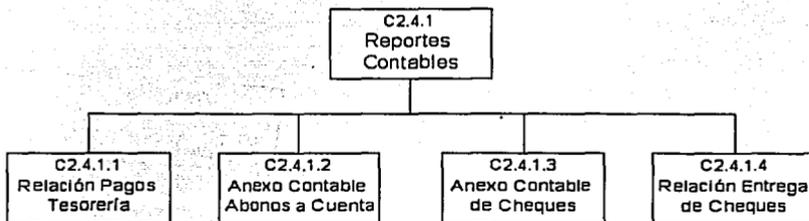


Figura 3.66 Apartado C2.4.1 Reportes Contables

El apartado "C2.4.2 Reportes Control Interno", contendrá los reportes que servirán para controlar los contra-recibos pagados; permitirán saber que cheques fueron emitidos y cuáles fueron cancelados, gráficamente tenemos: (figura 3.67)

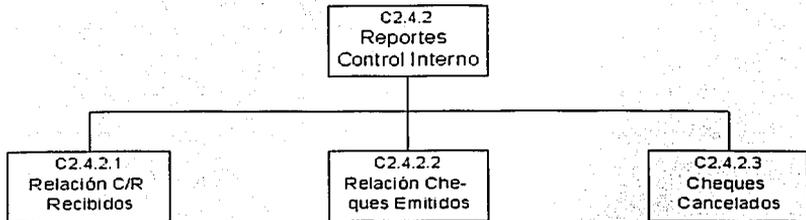


Figura 3.67 Apartado C2.4.2 Reportes Control Interno

El tercer módulo del Sistema "C3.0 Contabilización", realizará la afectación contable de los pagos efectuados; emitirá las pólizas contables y generará el archivo contable de transferencia que será enviado al Área Central de Contabilidad, la figura 3.68 ilustra lo mencionado anteriormente:

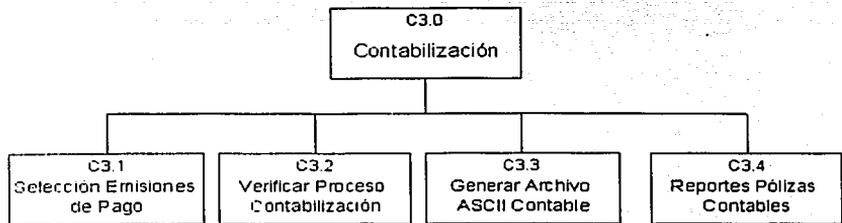


Figura 3.68 Módulo C3.0 Contabilización

Dentro del módulo de Contabilización existe el apartado "C3.2 Verificar Proceso Contabilización", mediante el cual se podrá verificar si aún persisten errores en la afectación contable de los pagos realizados; se podrá realizar la emisión previa de las pólizas contables para su revisión; y en el supuesto de errores, se podrá proceder a la cancelación contable de los movimientos, gráficamente tenemos: (figura 3.69)

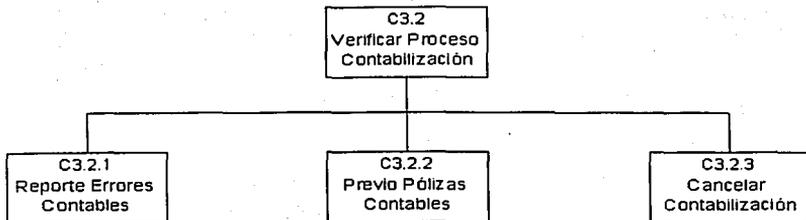


Figura 3.69 Apartado C3.2 Verificar Proceso Contabilización

El cuarto módulo del Sistema "C4.0 Catálogos", quedará dividido en tres grandes bloques, a través de los cuales se les dará mantenimiento a todos los catálogos que formarán parte del sistema, gráficamente tenemos: (figura 3.70)

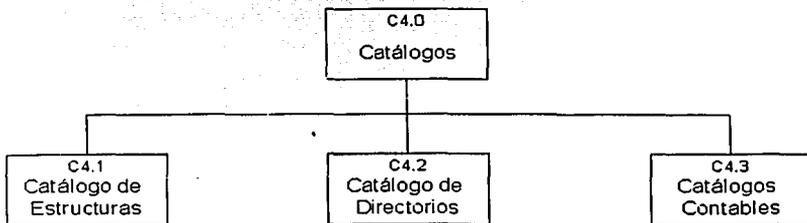


Figura 3.70 Módulo C4.0 Catálogos

El apartado "C4.1 Catálogos de Estructuras" abarcará el mantenimiento a las tablas de: Bancos, Regiones, Plazas, Departamentos, Ciudades y Zonas, el cual consistirá en dar altas, bajas, cambios y consultas de la información contenida en éstos, gráficamente tenemos: (figura 3.71)

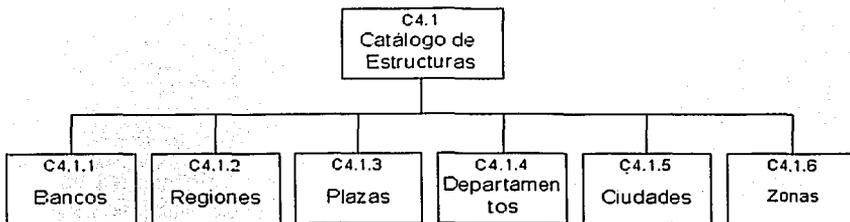


Figura 3.71 Apartado C4.1 Catálogo de Estructuras

El apartado "C4.2 Catálogos de Directorios" abarcará el mantenimiento a las tablas de: Proveedores, Responsables de Revisión y Días Inhábiles, el cual consistirá en dar altas, bajas, cambios y consultas de la información contenida en éstos, gráficamente tenemos: (figura 3.72)

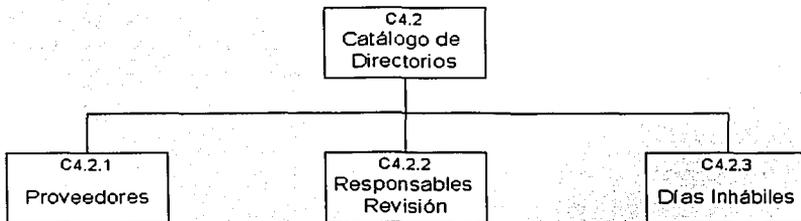


Figura 3.72 Apartado C4.2 Catálogo de Directorios

El apartado "C4.3 Catálogos Contables" abarcará el mantenimiento a las tablas de: Cuentas Contables, Ítems, Conceptos y Centros de Costos, el cual consistirá en dar altas, bajas, cambios y consultas de la información contenida en éstos, gráficamente tenemos: (figura 3.73)

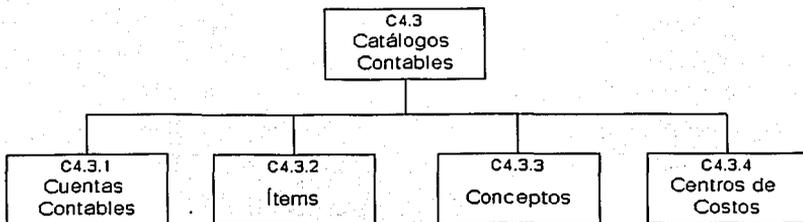


Figura 3.73 Apartado C4.3 Catálogos Contables

El quinto módulo del Sistema "C5.0 Utilerías", proporcionará los elementos necesarios para su correcto funcionamiento, a través del cual se integrarán las herramientas de mantenimiento de la base de datos, definición de los usuarios del sistema, y transferencia de información para la actualización del Sistema para el Trámite de Pagos, gráficamente tenemos: (figura 3.74)

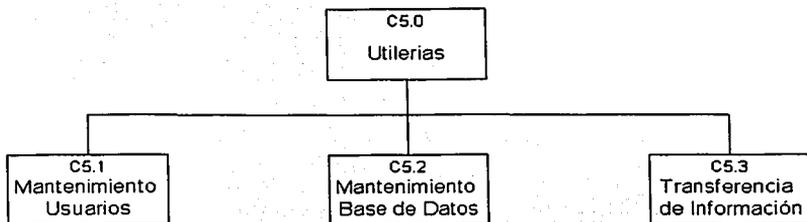


Figura 3.74 Módulo C5.0 Utilerías

El apartado "C5.1 Mantenimiento de usuarios", permitirá el registro y mantenimiento de los usuarios del sistema; llevará el control de una bitácora de accesos que permitirá conocer a los usuarios que utilizan el sistema, gráficamente tenemos: (figura 3.75)

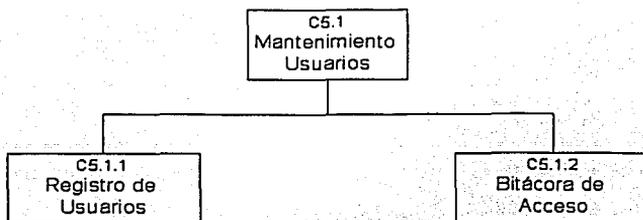


Figura 3.75 Apartado C5.1 Mantenimiento Usuarios

El apartado "C5.2 Mantenimiento Base de Datos", proporcionará las herramientas necesarias para darle mantenimiento a los datos que conforman el sistema; regenerará los índices de las tablas; permitirá respaldar y restaurar la información de éstas, lo anterior con la finalidad de garantizar la integridad de la información contenida en el sistema, gráficamente tenemos: (figura 3.76)

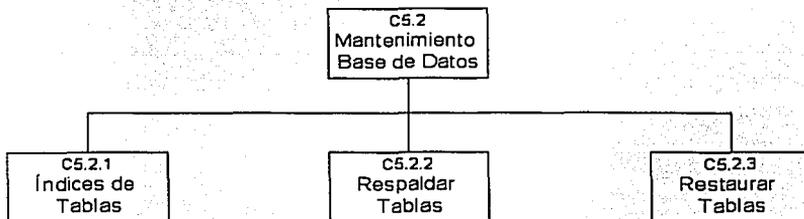


Figura 3.76 Apartado C5.2 Mantenimiento Base de Datos

El último apartado del módulo de utilerías es el denominado "C5.3 Transferencia de Información", será el encargado de actualizar los catálogos utilizados por el Sistema para el Trámite de Pagos; realizará la actualización automática de los ítems contables provenientes del Área Central de Contabilidad, por lo que quedará definido de la siguiente forma: (figura 3.77)

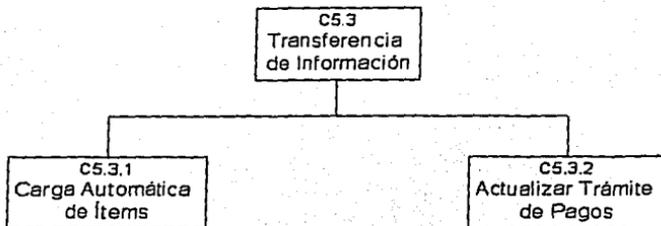


Figura 3.77 Apartado C5.3 Transferencia de Información

Finalmente, el último módulo del Sistema Administrativo-Contable de Pagos, es el denominado "C6.0 Personalizar Sistema", el cual contendrá los parámetros del sistema, tales como: ruta de bases de datos, ruta de envío de información, leyendas y logotipos de reportes, etc.; incluirá una opción para que el usuario pueda cambiar su contraseña de acceso; mostrará información del sistema para que los usuarios puedan contactar al administrador del sistema en caso de cualquier error, gráficamente tenemos: (figura 3.78)

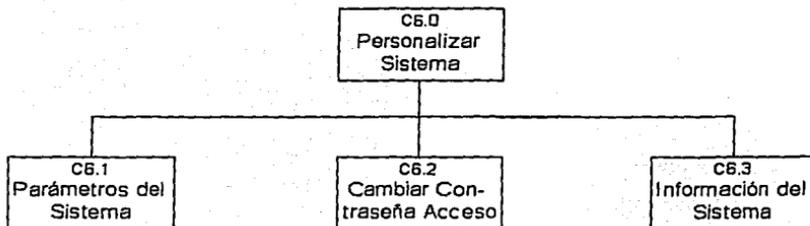


Figura 3.78 Módulo C6.0 Personalizar Sistema



## 4.1 CONSTRUCCIÓN SISTEMA PARA EL TRÁMITE DE PAGOS

Tomando como base el diseño de las bases de datos establecido previamente, aunado a la definición de los módulos que formarán parte del **Sistema para el Trámite de Pagos**, comienza la labor de construcción del sistema, el cual se implementará utilizando el modelo tecnológico definido durante el análisis del sistema, iniciando con la construcción de las relaciones y posteriormente con la construcción de los formularios, informes, menús y código diverso, tal como: seguridad, bitácora de acceso, transferencia de archivos y actualización del sistema.

### 4.1.1 RELACIONES

Con FoxPro se pueden crear tablas y manipular los datos utilizando instrucciones tipo SQL, tales como CREATE, INSERT, UPDATE y SELECT, dichas instrucciones se introducen a través de la ventana de comandos de FoxPro. Esta ventana mantiene un histórico de los comandos que se introduce durante una sesión de FoxPro. O mediante la utilización de asistentes gráficos definidos por el sistema de menús de FoxPro, la selección de opciones a partir de esta forma resulta ser simple e intuitiva mediante el uso de un ratón.

El menú principal se encuentra en la parte superior de la pantalla y la ventana de comandos está cerca de la esquina inferior derecha de la pantalla. (figura 4.1)

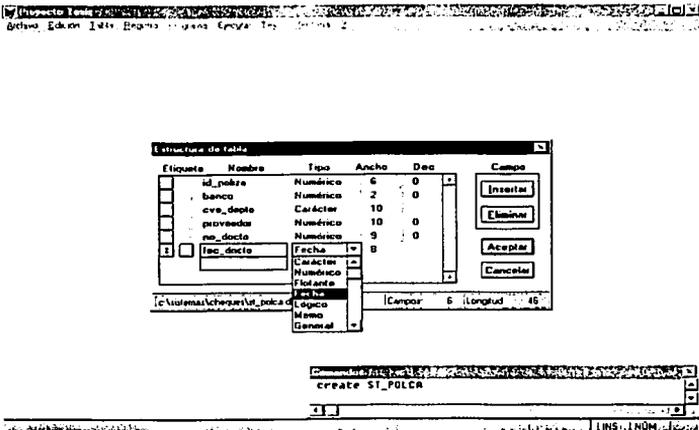
Ahora bien, para iniciar con la creación de las tablas tan solo bastará con introducir el comando CREATE TABLE en la ventana de comandos de FoxPro, lo cual nos desplegará la ventana de estructura de tabla para empezar con la definición de campos.

Los tipos de campos que soporta FoxPro son:

- Carácter (C):** Este tipo de dato puede contener letras, números y caracteres especiales, la amplitud máxima del campo es de 254 caracteres.
- Numérico (N):** Contiene sólo números y un punto decimal, además se puede comenzar con un signo, la amplitud máxima del campo es de 20 caracteres, considerando que el punto y signo ocupan cada uno un espacio.
- Flotante (F):** Es similar al tipo numérico, la diferencia radica en que fue diseñado para cálculos científicos, debido a que internamente almacena más dígitos significativos durante las operaciones efectuadas con este tipo de dato.
- Lógico (L):** Este tipo de dato contiene solo los valores T o F, para indicar True o False (Verdadero o Falso).
- Fecha (D):** Este tipo de dato contiene ocho caracteres de ancho, dos para definir el día, dos para el mes y 4 para el año. Para facilitar su captura se incluye el carácter de diagonal, pero internamente solo se almacenan los números.
- Memo (M):** El tipo de dato Memo puede contener cualquier cantidad y tipo de dato. El límite para la longitud del campo es la cantidad de espacio de disco que se tenga.
- General (G):** El tipo de dato General contiene objetos OLE, una característica que se introdujo en Windows 3.1 y que permite insertar objetos creados en una aplicación dentro de otra. Por ejemplo, se puede usar para incluir imágenes BMP, JPG, gráficas o sonidos creados en otra aplicación Windows dentro de la tabla FoxPro.

Finalmente, los archivos de bases de datos creados con FoxPro, adquieren de manera automática la extensión DBF (DataBase File) y si fueron definidos campos memos en su estructura, se creará un archivo con el mismo nombre, pero con la extensión FPT, que quiere decir "FoxPro Text".

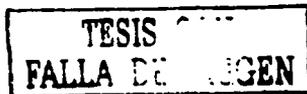
La figura 4.1 muestra la interfaz gráfica que nos permite crear las tablas que formarán parte de nuestra aplicación:



A continuación, se muestra solamente la definición de las principales tablas que conformarán el Sistema para el Trámite de Pagos:

La definición de la tabla ST\_POLCA, quedará de la siguiente forma:

```
CREATE TABLE ST_POLCA (
    ID_POLIZA      N(6),
    BANCO          N(2),
    CVE_DEPTO     C(10),
    PROVEEDOR     N(10),
    NO_DOCTO      N(9),
    FEC_DOCTO     D(8),
    TIPO_GASTO    N(1),
    VIÁTICO       L(1),
```



DETALLEDOC L(1),  
 ITEM C(12),  
 SERIE N(7),  
 CONCEPTO N(4),  
 DEPTOGASTO C(10),  
 FORMA\_PAGO N(1),  
 IMPORTE N(12,2),  
 DESCUENTO N(9,2),  
 IVA N(9,2),  
 RET\_IVA N(9,2),  
 RET\_ISR N(9,2),  
 TOTAL N(12,2),  
 COMENT1 C(250),  
 COMENT2 C(250),  
 ANTICIPO N(9),  
 SALDO\_COMP N(9,2),  
 ESTATUS C(1),  
 CVE\_TURNO C(4),  
 FEC\_POLIZA D(8),  
 NO\_POLIZA C(10),  
 FEC\_ENVIO D(8),  
 LOTE\_ENVIO C(8),  
 DEPTO\_USU C(10),  
 FEC\_ALTA D(8),  
 FEC\_VALOR D(8)

La definición de la tabla ST\_MOVCA, quedará de la siguiente forma:

```

CREATE TABLE ST_MOVCA (
  ID_POLIZA N(6),
  ID_DETALLE N(3),
  DEPTOGASTO C(10),

```

NO_DOCTO	N(9),
FEC_DOCTO	D(8),
CONCEPTO	N(4),
IMPORTE	N(10,2),
POR_DESC	N(6,2),
DESCUENTO	N(9,2),
POR_IVA	N(6,2),
IVA	N(9,2),
POR_RETIVA	N(6,2),
RET_IVA	N(9,2),
POR_RETISR	N(6,2),
RET_ISR	N(9,2),
TOTAL	N(10,2) )

La definición de la tabla ST\_VIAJE, quedará de la siguiente forma:

```
CREATE TABLE ST_VIAJE (
  ID_POLIZA      N(6),
  ID_DETALLE     N(3),
  ID_VIAJE      N(3),
  CI_SALIDA     N(3),
  CI_DESTINO    N(3),
  CI_DIAS       N(2),
  CI_FECHA      D(8),
  CI_HORA       C(5),
  CI_BOLETO     L(1),
  CF_SALIDA     N(3),
  CF_DESTINO    N(3),
  CF_FECHA      D(8),
  CF_HORA       C(5),
  CF_BOLETO     L(1),
  FINVIAJE      L(1),
  CIUDAD        N(3),
```

A_REALES	N(2),
A_FECINI	D(8),
A_FECFIN	D(8) )

La definición de la tabla ST\_LOTE, quedará de la siguiente forma:

```
CREATE TABLE ST_LOTE (
    LOTE          C(8),
    BANCO        N(2),
    AÑO          N(4),
    INI_FIRMA    C(3),
    MES          N(2),
    FOLIO        N(3),
    DEPTO_USU    C(10),
    FECHAENVIO   D(8),
    FEC_VALOR    D(8),
    VALOR        N(12,2),
    FACTURAS     N(4),
    CVE_DEPTO    C(10),
    NOM_DEPTO    C(60),
    NOM_RESP     C(40),
    ESTATUS      C(1) )
```

#### 4.1.2 FORMULARIOS

FoxPro incorpora un generador de pantallas, el cual es una herramienta de programación muy poderosa y nueva, similar a la de los lenguajes visuales, tales como: Visual Basic, Delphi, CA-Clipper, etc., a través del cual se pueden diseñar pantallas con botones, botones de radio, cuadros de selección, controles y listas despegables que se ven y funcionan como cualquier aplicación de Windows.

El generador de pantallas se invoca simplemente tecleando la instrucción CREATE SCREEN desde la ventana de comandos, en la figura 4.2 se muestra la ventana del generador de pantallas de FoxPro.

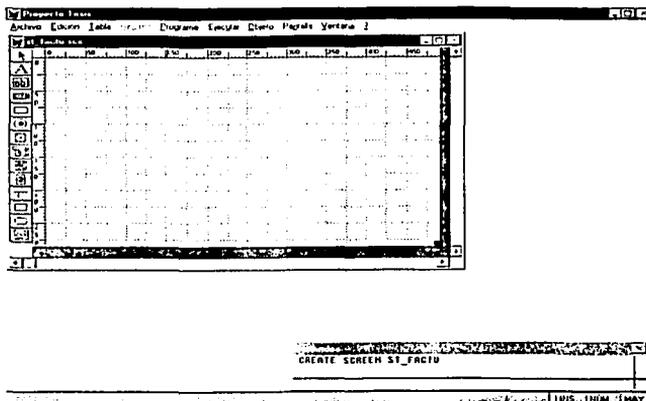


Figura 4.2 Generador de Pantallas de FoxPro

El cuadro de herramientas que está en el borde izquierdo, permite agregar todos los controles necesarios para que el usuario pueda realizar acciones específicas, aceptar entradas de datos y mostrar información, a continuación se describe de manera breve el significado de cada uno:

-  ▪ **Puntero de selección**  
Control que permite seleccionar objetos dentro de la pantalla para cambiarlos de posición o eliminarlos.
  
-  ▪ **Herramienta de texto**  
Control que permite agregar o editar texto (etiquetas) en la pantalla, el cual no podrá ser modificado por el usuario.



▪ **Herramienta de campo**

Es el control básico que permite a los usuarios agregar o modificar datos almacenados en un campo, con excepción de los campos memos y general.



▪ **Herramienta de región de edición**

Control que permite la edición de campos de texto largos o campos memo, permitiendo el ajuste automático de línea, ofrece la posibilidad de moverse por el texto con las teclas de dirección.



▪ **Herramienta de botón de comando**

Control que permite la creación de botones, éstos permiten a los usuarios realizar determinadas acciones que no tienen nada que ver con la captura de valores, por ejemplo: abrir o cerrar una ventana, guarda o cancelar modificaciones de datos, etc.



▪ **Herramienta de botón radio**

Control que permite a los usuarios especificar una opción entre varias en un cuadro de diálogo, en lugar de introducir los datos.



▪ **Herramienta de casilla de verificación**

Control empleado para permitir que un usuario especifique un estado booleano: Verdadero o Falso, Activado o Desactivado, Abierto o Cerrado.



▪ **Herramienta de emergencia**

Conocido como **ComboBox**, combina las características de un control de texto y un control de lista. Los usuarios pueden introducir información en la parte del cuadro de texto y seleccionar un elemento en la parte de cuadro de lista del control.

- 
  - Herramienta de lista  
 Conocido como ListBox, muestra una lista de elementos en la que el usuario puede seleccionar uno o más. Si el número de elementos supera el número que puede mostrarse, se agregará automáticamente una barra de desplazamiento al control.
  
- 
  - Herramienta control giratorio  
 Mediante este control se pueden introducir datos variando la posición del cursor, puede emplearse para permitir a los usuarios realizar selecciones mostrando los valores.
  
- 
  - Herramienta de línea  
 Control utilizado para dibujar líneas en pantalla.
  
- 
  - Herramienta de rectángulo  
 Control utilizado para dibujar un rectángulo en la pantalla.
  
- 
  - Herramienta de rectángulo redondeado  
 Control utilizado para dibujar un rectángulo con esquinas redondeadas en la pantalla.
  
- 
  - Herramienta de imagen  
 Control utilizado para incluir en la pantalla imágenes del tipo BMP o campos generales.

Una vez terminado el diseño de la pantalla, FoxPro permite agregar código extra, es decir, comandos de programación adicionales al código que es generado de forma automática por el generador de pantalla, la figura 4.3 ilustra el cuadro de diálogo que se muestra para agregar código:

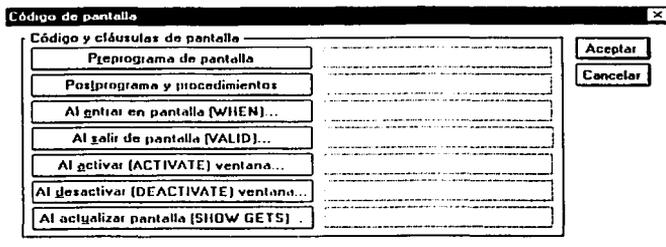


Figura 4.3 Cuadro de diálogo para agregar código de pantalla

Las cláusulas preprograma y postprograma se utilizan para agregar código que se ejecutará antes y después de que se desplegó la pantalla. Las cláusulas when y valid se utilizan para agregar fragmentos de código que van asociados al comando READ, el cual permite la entrada de datos tanto para la pantalla, como para cada objeto que forma parte de ésta. Las cláusulas activate y deactivate, son utilizadas para introducir código que será ejecutado cuando se realice la navegación entre varias pantallas. Por último, el código que se introduce en la cláusula show gets se ejecutará cuando se realice la actualización (refresh) de la pantalla.

Finalmente, para concluir con la forma en que son creados los formularios, solo falta mencionar como se controlan las características generales de la pantalla completa, para tal efecto, FoxPro incorpora un cuadro de diálogo, el cual se muestra en la figura 4.4:

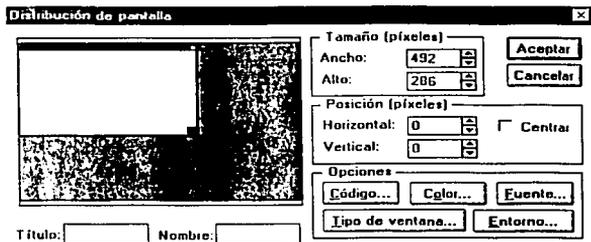


Figura 4.4 Cuadro de diálogo para la distribución de pantalla

A través de este cuadro de diálogo se podrá definir el tamaño, la posición, el color, el tipo de letra y el tipo de ventana, así como, el título que aparecerá en la barra de título de la ventana y el nombre que se empleará para referirse a ésta dentro de los programas.

La figura 4.5 muestra el estándar definido para la pantalla de escritorio utilizada por el sistema:

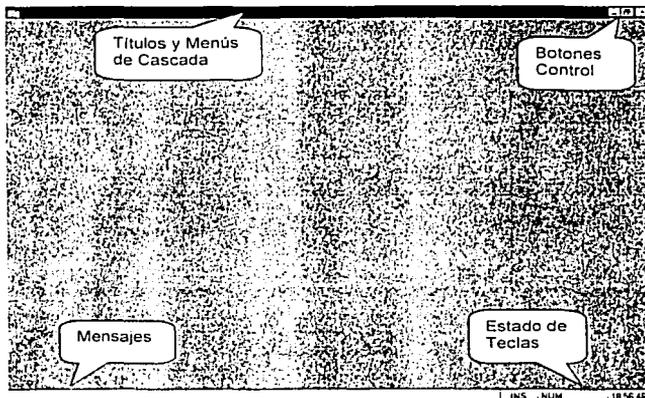


Figura 4.5 Estándar de la pantalla de escritorio

En la parte superior de la ventana se desplegará el nombre de la ventana, así como los menús de cascada del sistema; en la esquina superior derecha se mostrarán los botones de minimizar, maximizar y cerrar. En la barra inferior, se desplegará en la esquina izquierda los mensajes de ayuda para el usuario y en la esquina derecha se mostrará la hora y el estado de las teclas BloqMayús, BloqNúm e Insert; Por último, la parte central tendrá un fondo de color cian y será aquí en donde se desplegarán todos los formularios del sistema.

El formulario que se muestra en la figura 4.6, ejemplifica la forma en que son utilizadas todas las herramientas visuales descritas anteriormente, a parte de que será el formulario principal, a través del cual se realizará la captura de facturas:

The screenshot shows a window titled 'BANCA CREMI' with the following fields and options:

- Banco:** A dropdown menu.
- Beneficiario:** A text field with a search icon.
- Tipo de Cargo:** Radio buttons for 'Acreditac.', 'Acreditac.', 'Comprobar', 'B. Admin', 'Nada Documenta'.
- Concepto:** A dropdown menu with 'ITEM' selected.
- Depos. Cuentas:** A text field with a search icon.
- FORMA PAGO:** Radio buttons for 'Cheque', 'Abono en Cta', 'Sin Pago', 'Pago Fecha Valor'.
- DESCRIPCION DEL CONCEPTO:** A large text area with a search icon.
- Table of Payment Methods:**

Bitneta	0.00
Descuento	0.00
iva	0.00
Ret. IVA	0.00
Total	
- Navigation Bar:** Buttons for 'Anterior', 'Siguiente', 'Buscar', 'Guardar', 'Cancelar', 'Salir'.

Callout boxes indicate: 'Banco' (pointing to the dropdown), 'Desplazamiento entre registros' (pointing to the navigation buttons), and 'Llamada a Subformularios' (pointing to the search icons).

Figura 4.6 Formulario de captura de facturas

Invariably in the construction of the forms, always the navigation bar that will allow us to move between records and the buttons to make calls to other forms, will be found in the lower part, another point to highlight is the type of font to use, which will be Arial of 8 points.

The navigation bar will allow us to advance to the next, previous or search for a record, as well as through this one we will be able to give of high, low or modify information. (figure 4.7)

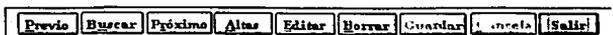


Figura 4.7 Barra de navegación entre registros

In what is referred to the search of records, this will be carried out through a BROWSE screen, in which the most important information will be shown; to move between records will be used the arrow keys and to select the information the ESC key, the figure 4.8 illustrates this situation:

TESIS CON  
FALLA DE ORIGEN

Seleccione un registro y oprima (Esc) para salir				
Tipo de Gasto	Cve.Prov.	Nom.Prov.	No.Docto.	Concepto
Anticipo	3	ACCOR SERVICIOS EMPRESARIALES, S.A. DE C.V.	100246	COMPRA DE CUPONES DE GASO
Comprobación	3	ACCOR SERVICIOS EMPRESARIALES, S.A. DE C.V.	0	COMPROBACION COMPRA DE C
Anticipo	423699	PRIEGO ROMERO MARTHA	100249	viaje a chihuahua por programa de
Comprobación	423699	PRIEGO ROMERO MARTHA	0	comprobacion viaje a chihuahua po
Anticipo	1046	SEGUROS COMERCIAL AMERICA, S.A. DE C.V.	100250	SEGURO CLIENTES HIPOTECAR
Comprobación	697	TARRAB MASLATON MAYER	0	comprobacion de renta del mes de
Anticipo	697	TARRAB MASLATON MAYER	100248	renta del mes de abril Edif. Reform.
Anticipo	1114	TELEFONOS DE MEXICO, S.A. DE C.V.	100247	recibo correspondiente de las cuen
Comprobación	1114	TELEFONOS DE MEXICO, S.A. DE C.V.	0	comprobacion del recibo correspon
Bolero de Avion	1208	VEMSA VIAJES EMPRESARIALES, S.A. DE C.V.	100249	Anticipo del viaje de PRIEGO ROM

Figura 4.8 Pantalla Browse para mostrar información

Para capturar la información de un campo que provenga de catálogo, se permitirá ingresar la clave o mediante el uso de un Combo Box se realizará su búsqueda, la figura 4.9 muestra esta situación:

Concepto 0

Figura 4.9 Combo Box para búsqueda de información

Por último, la forma de interactuar con el usuario será a través de ventanas del tipo "Message Box", las cuales servirán para solicitar la respuesta a una acción, ejemplo: (figura 4.10)

**ATENCIÓN**

¿Está seguro de cancelar esta referencia?

Figura 4.10 Ventana de mensaje solicitando respuesta a una acción

TESIS CON  
FALLA DE ORIGEN

Para informar al usuario, si hace falta algún elemento para el correcto funcionamiento de un proceso: (figura 4.11)

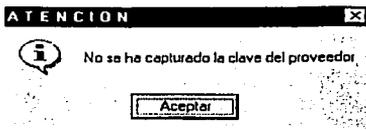


Figura 4.11 Ventana de mensaje indicando si hace falta información

Para mostrar mensajes de error de información mal capturada, o procesos incorrectos: (figura 4.12)

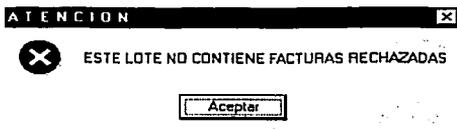


Figura 4.12 Ventana de mensaje indicando errores

#### 4.1.3 INFORMES

FoxPro incorpora un generador de informes muy potente que permite realizar en pantalla el diseño de los campos, etiquetas y gráficos de manera sencilla y rápida por el usuario.

El generador de informes se invoca utilizando la instrucción `CREATE REPORT` desde la ventana de comandos, la figura 4.13 muestra la ventana del generador de informes de FoxPro:



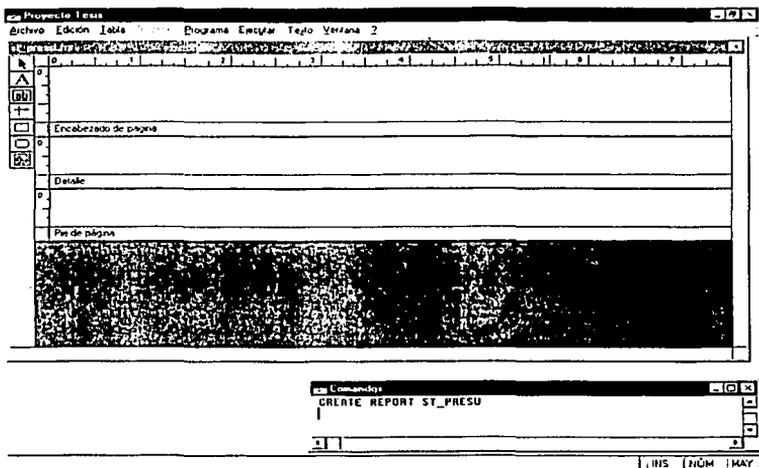


Figura 4.13 Generador de informes de FoxPro

Al igual que el generador de pantallas, el generador de informes cuenta con un cuadro de herramientas situado en el borde izquierdo, el cual facilita la creación de campos, etiquetas y gráficos, a continuación se describe de forma breve cada uno de ellos:

- ☞
  - **Puntero de selección**  
Se le llama herramienta de selección porque cuando el apuntador está habilitado, se puede seleccionar cualquier objeto dentro del informe.
  
- A
  - **Herramienta de texto**  
Herramienta que permite agregar o editar etiquetas (texto) en el informe.
  
- tbl
  - **Herramienta de campo**  
Herramienta que permite agregar expresiones, campos calculados o simplemente campos de una tabla dentro del informe, incluyendo campos memos y general.

- 
 ▪ Herramienta de línea  
 Control utilizado para dibujar líneas dentro del informe, las cuales solo podrán ser horizontales o verticales.
  
- 
 ▪ Herramienta de rectángulo  
 Control utilizado para dibujar un rectángulo dentro del informe.
  
- 
 ▪ Herramienta de rectángulo redondeado  
 Control utilizado para dibujar un rectángulo con esquinas redondeadas dentro del informe.
  
- 
 ▪ Herramienta de imagen  
 Control utilizado para incluir en el informe imágenes del tipo BMP o campos generales.

La ventana del generador de informes se encuentra dividida en tres bandas: encabezado, detalle y pie de página, los cuales sirven para incluir información que será desplegada al principio de cada página, los datos que formarán parte del informe y las anotaciones al final de cada página, respectivamente.

Adicionalmente, se podrán habilitar otras, siendo éstas:

- Título:** Permite incluir en la parte superior y solamente en la primer página un título.
  
- Resumen:** Permite incluir al final del informe una colección de datos, los cuales únicamente se presentarán en la última página.
  
- Agrupación:** Permite totalizar datos y mostrarlos dentro de cada agrupación realizada.

Cuando se ha concluido con el diseño del reporte, FoxPro permite realizar la distribución de la página, en donde se podrán establecer los márgenes, número de columna,

orientación de la hoja y otras características más, la figura 4.14 muestra ésta ventana de diálogo:

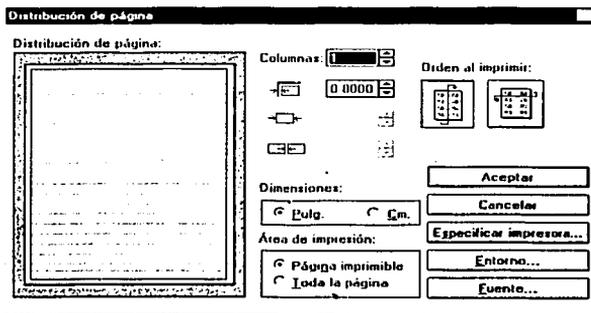


Figura 4.14 Ventana de diálogo Distribución de Página

Finalmente, para concluir con el diseño del informe, FoxPro incorpora una opción muy útil de presentación preliminar, la cual permite ver en pantalla cómo aparecerá el informe ya impreso, la figura 4.15 ilustra la ventana de diálogo página previa:

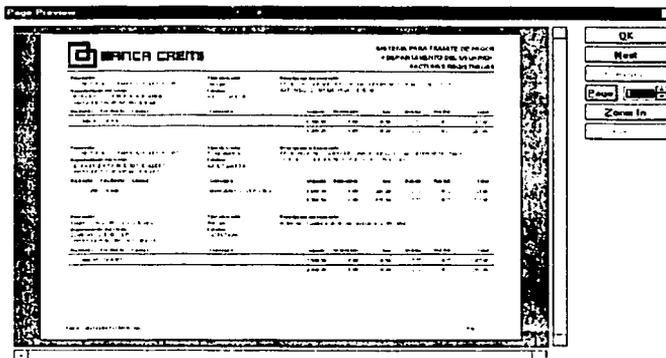


Figura 4.15 Ventana de diálogo Página Previa

Para iniciar con la creación de los informes, sólo bastará con seleccionar y arrastrar las herramientas al área de detalle, encabezado o pie de página.

Para la definición y creación de los informes que emitirá el sistema, se establecerá como estándar el mismo encabezado y pie de página.

Los encabezados quedarán definidos de la siguiente forma: (figura 4.16)



SISTEMA PARA TRAMITE DE PAGOS  
<DEPARTAMENTO DEL USUARIO>  
SOLICITUD DE TRAMITE DE PAGO

Figura 4.16 Estándar de Encabezados

Como se puede apreciar, del lado izquierdo se desplegará el logotipo del banco; del lado derecho se incluirán tres leyendas que identificarán el departamento que origina el gasto, el departamento del usuario que emite el reporte y el nombre del reporte, la definición de éstos estará parametrizada para que los usuarios puedan personalizar sus reportes.

El pie de página quedará definido de la siguiente forma: (figura 4.17)

Fuente: Sistema para Trámite de Pagos

Fecha de Emisión 30/03/2003

Página 1

Figura 4.17 Estándar de Pie de Página

Como se puede apreciar, del lado izquierdo se hará referencia a la fuente de donde se obtuvo la información, al centro la fecha de emisión y en la derecha el número de hoja del reporte.

Para la emisión de los informes se habilitará una ventana de diálogo que permita a los usuarios determinar hacia donde desean enviar sus reportes, la figura 4.18 ilustra esta situación:

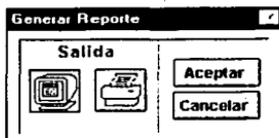


Figura 4.18 Ventana para generar los reportes

Si se elige la opción de pantalla, FoxPro tiene la opción de habilitar de manera similar a los aplicativos de Office una ventana de vista previa, la cual le permitirá a los usuarios navegar por todas las hojas del reporte, la figura 4.19 ilustra esta situación:

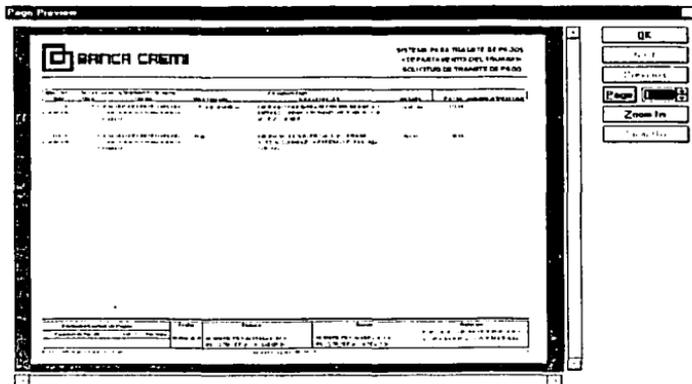


Figura 4.19 Vista previa de un reporte

Si se elige la opción de impresora, se habilitará la ventana de diálogo estándar de Windows para efectuar la impresión de los informes, la figura 4.20 muestra esta situación:

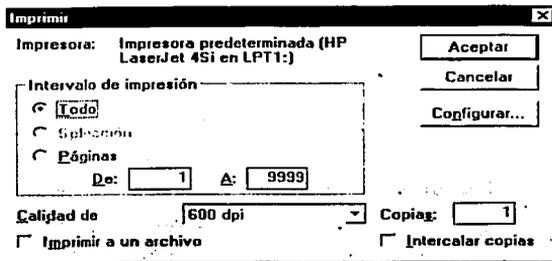


Figura 4.20 Ventana de diálogo para enviar a impresora un informe

#### 4.1.4 MENÚS

FoxPro incluye dentro de sus principales ventajas un generador de menús, a través del cual se pueden definir de manera sencilla y rápida los menús, evitando con esto, escribir una tediosa y repetitiva serie de comandos.

El generador de menús se invoca utilizando la Instrucción CRATE MENU desde la ventana de comandos, la figura 4.21 muestra la ventana del generador de menús:

TESIS CON  
FALLA DE ORIGEN

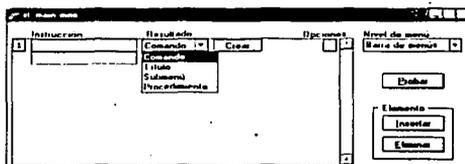


Figura 4.21 Generador de Menús de FoxPro

Es muy sencillo utilizar el generador de menús, solo bastará con teclear los indicadores del menú dentro de la columna izquierda (instrucción), para que se pueda desplegar la columna resultado, la cual incluirá cuatro posibles acciones a seguir.

- Comando:** Se podrá escribir solo un comando válido de FoxPro, normalmente llamada a un formulario o informe.
- Título:** Normalmente, se utiliza para incluir líneas de separación entre opciones, para tal efecto se teclean los símbolos \-.
- Submenú:** Opción utilizada para la creación de llamadas a otros menús.
- Procedimiento:** Permite escribir una serie de comandos que se incluirán dentro de la definición del menú.

**TESIS CON  
FALLA DE ORIGEN**

Otra de las características importantes del diseñador de menús, es el hecho de poder definir accesos rápidos, designando para tal efecto teclas de método abreviado para la instrucción, lo cual se logra agregado tan sólo los símbolos \< antes de la tecla que se utilizará para tal efecto.

Adicional a lo anterior, se pueden asociar opciones a cada instrucción del menú, las cuales se habilitan cuando se presiona el botón Opciones, la figura 4.22 muestra la ventana de diálogo "Opciones de Instrucción":

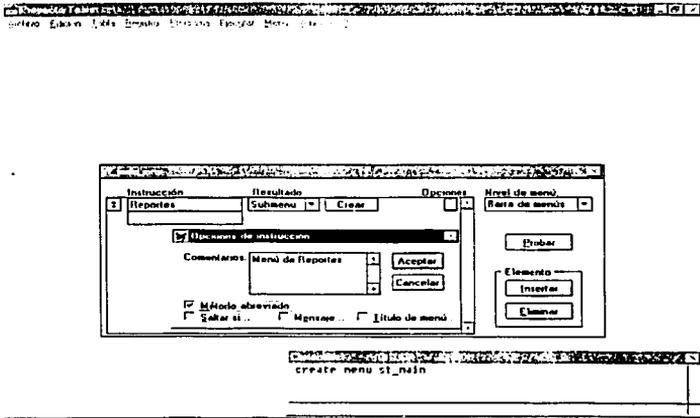


Figura 4.22 Ventana de diálogo "Opciones de Instrucción"

**Método Abreviado:** Se utiliza para especificar la combinación de teclas que formaran el atajo de la instrucción.

**Saltar si:** Tiene como finalidad desactivar la instrucción del menú, pero sólo bajo las condiciones que se le especificuen.

**Mensaje:** Sirve para especificar mensajes de ayuda, pero sólo cuando se tiene resaltada la instrucción.

**Título del Menú:** Se utiliza para asignarle un nombre a la instrucción, la cual se puede utilizar para hacer referencia a ésta dentro del código de cualquier programa.

Sólo resta mencionar, cómo se pueden habilitar los menús dentro del código de cualquier programa, para tal efecto se hace referencia a los siguientes comandos:

**SET SYSMENU OFF:** Esconde y desactiva el menú principal de FoxPro.

**SET SYSMENU TO <menú>:** Despliega cualquier menú previamente diseñado en el escritorio de FoxPro.

**SET SYSMENU TO DEFAULT:** Regresa el menú del sistema a su configuración normal.

Para iniciar con la definición de los menús que formarán parte del **Sistema para el Trámite de Pagos**, se utilizarán **menús de mosaico** para la creación del menú principal, el cual se implementará utilizando formularios, que incluirán el uso de botones de comando para la llamada a los menús internos, la figura 4.23 muestra el menú principal del sistema:

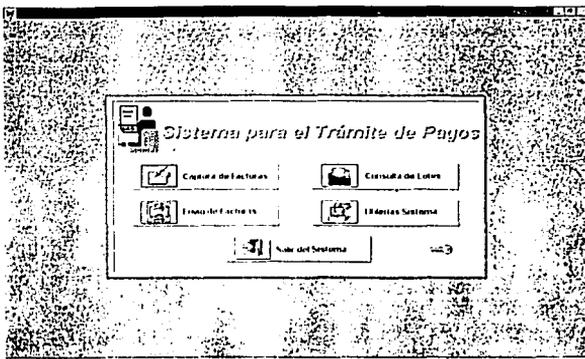


Figura 4.23 Menú Principal del Sistema para el Trámite de Pagos

Para la creación de los menús internos del sistema, se utilizarán menús de cascada, los cuales se implementarán con el generador de menús de FoxPro, la figura 4.24 muestra el menú interno para las utilerías del sistema:

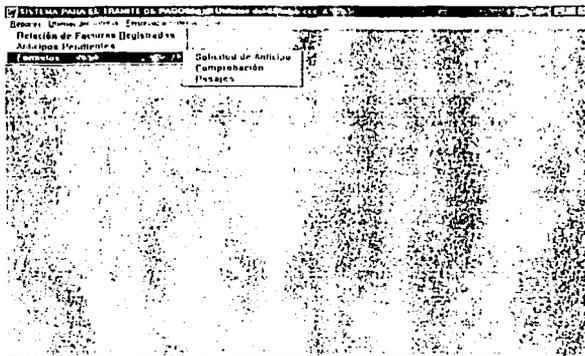


Figura 4.24 Menú de Cascada utilizado para las utilerías del sistema

#### 4.1.5 CÓDIGO

FoxPro incluye dentro de sus múltiples ventanas un administrador de proyectos, a través del cual se pueden integrar todos los archivos de una aplicación en un solo, ya sea en un APP, el cual puede correr utilizando FoxPro, o un archivo EXE que opere por sí mismo.

El administrador de proyectos se invoca utilizando la instrucción `CRATE PROJECT` desde la ventana de comandos, la figura 4.25 muestra la ventana del administrador de proyectos:

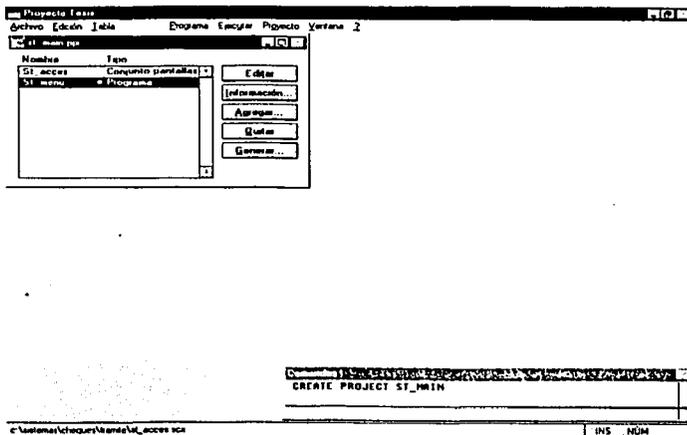


Figura 4.25 Administrador de Proyectos de FoxPro

FoxPro almacenará la información del proyecto en un archivo de base de datos con extensión PJX y en un archivo Memo asociado con la extensión PJT.

Para agregar un archivo al proyecto solo bastará con dar clic en el botón Agregar de la ventana Proyecto, con lo cual se desplegará la siguiente ventana de diálogo: (figura 4.26)

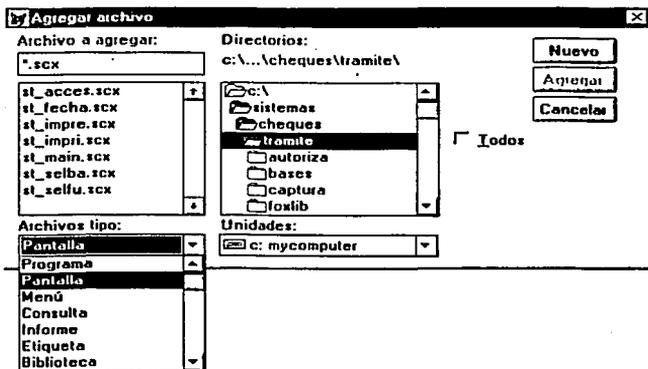


Figura 4.26 Ventana de diálogo Agregar Archivo.

Los tipos de archivos que se pueden agregar a un proyecto son: Programas (PRG), formularios.(SCX), informes (FRX), menús (MRX), librerías (FLL), imágenes (BMP), tablas (DBF) e incluso aplicaciones (APP).

Para crear la aplicación (APP o EXE) del proyecto, sólo bastará con dar clic en el botón generar de la ventana Proyecto, con lo cual se desplegará la siguiente ventana de diálogo: (figura 4.27)

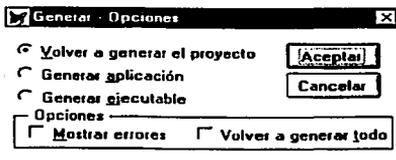


Figura 4.27 Ventana de diálogo Generar Aplicación

**Volver a generar:** Actualiza el proyecto. Agrega cualquier archivo nuevo que sea llamado por los programas en el proyecto y genera un nuevo código si el diseño de pantalla o menú se han alterado desde que se realizó la última construcción.

**Generar Aplicación:** Actualiza el proyecto y crea un archivo APP.

**Generar Ejecutable:** Actualiza el proyecto y crea un archivo EXE.

Una de las características más útiles del administrador de proyectos, es que si se incorpora un nuevo archivo, al seleccionar cualquiera de las opciones anteriores, en forma automática se agregarán todos los elementos asociados a este archivo en particular, en lugar de incluirlos de uno en uno.

Finalmente, para concluir con la forma de generar código en FoxPro, tan sólo resta mencionar que cuando se realiza un sistema, lo que menos se quiere es que aparezca la pantalla de presentación del lenguaje, afortunadamente FoxPro incluye una ventana de diálogo denominada Opciones, a través de la cual se puede suprimir el despliegado de ésta, adicionalmente se puede incluir información relevante del proyecto, tales como: información general del autor, ubicación del código generado, el directorio origen de FoxPro, entre otros, a continuación se muestra esta ventana de diálogo: (figura 4.28)

**Opciones**

Información del programador

Autor: \_\_\_\_\_

Empresa: \_\_\_\_\_

Dirección: \_\_\_\_\_

Ciudad/Prov.: \_\_\_\_\_

País: \_\_\_\_\_ Cód. postal: \_\_\_\_\_

Ubicación del código generado

Guardar código

Con pantalla o menú

Con proyecto

En directorio: \_\_\_\_\_

Estado de comentarios

Cuadros ahedados

Asteriscos ahedados

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

Opciones del proyecto

Info para depuración  Códificar programas

Mostrar logotipo al ejecutar la aplicación

Directorio de origen... \_\_\_\_\_

c:\j\p=26

Aceptar

Cancelar

Guardar como predeterminado

Figura 4.28 Ventana de diálogo Opciones

TESIS CON  
FALLA DE ORIGEN

A continuación, sólo se mostrarán algunas partes de código que conformarán el Sistema para el Trámite de Pagos.

El siguiente código ejemplifica y muestra, entre otras cosas:

- Como se define la ventana que funcionará como el escritorio de nuestra aplicación,
- las llamadas a librerías y procedimientos externos,
- la declaración de funciones y procedimientos internos,
- la declaración de variables privadas y públicas,
- el llamado a las rutinas de control de errores,
- el llamado a formularios,
- la forma de cómo se utiliza el comando READ, necesario para el ingreso de información por parte del usuario, y
- las instrucciones necesarias para restablecer el ambiente de FoxPro.

```
.....  
* Sistema: SISTEMA PARA TRAMITE DE PAGOS  
* Programa: ST_MENU.PRG Menú Principal de Sistema  
* Autor: Israel Mendoza Pacheco  
.....
```

```
MODIFY WINDOW SCREEN TITLE "Sistema para Trámite de Pagos" ;  
FONT "Arial", 8 ;  
ICON FILE 'ST_LOGO.ICO' COLOR RGB(...,0,128,128) ;  
FILL FILE 'FONDO.BMP'
```

```
CLEAR
```

```
SET LIBRARY TO FOXTOOLS.FLL ADITIVE  
SET PROCEDURE TO ST_COMUN  
SET SYSMENU OFF
```

```
PUBLIC dropdead,tobedone,Inicio,m.hayerror  
PUBLIC nUsuFolio,cUser,cLogin,cAcceso,lAcceso,cDeptoUser,bancoUser ;  
nomBanco,cPathRoot,cDbfParam  
PUBLIC p_dbfcaptu,p_dbfred,p_dbfenvio,p_dbftempo ;  
p_titre1,p_titre2,p_nomelabo,p_pueelabo,p_nomrevi,p_puerevi ;  
p_moneda,p_nom_moné,p_iva,p_retiva,p_retsr
```

```
dropdead = .F.  
tobedone = "  
Inicio = .F.
```

```
cPathRoot = SYS(5)+SYS(2003)+'\'
```

```

nUsuFolio = 0
cUser      = ""
cLogin     = ""
cAcceso    = ""
cDeptoUser = ""
bancoUser  = 0
nomBanco   = SPACE(30)

IAcceso    = .F.

cDbfParam = SYS(5)+SYS(2003)+'1ST_PARAM'

=Def_ambte()
IF !Parametros(cDbfParam)
  CLOSE ALL
  CLEAR ALL
  RETURN
  *QUIT
ENDIF

CLEAR

DO PASSWORD.PRG WITH cUser,cLogin,cAcceso,IAcceso,cDeptoUser

IF !Acceso
  cPathUpFile = m.p_dbfCaptu
  DO SC_UPVER.PRG WITH PROGRAM(),m.BancoUser,m.cDeptoUser,cPathRoot,cPathUpFile

  ON ERROR DO manejaerror :
    WITH ERROR(),MESSAGE(),MESSAGE(1),PROGRAM(),LINENO()
  =BITACORA('01;')
  MODIFY WINDOW SCREEN TITLE ""
  KEYBOARD {ENTER} CLEAR
  READ VALID myhandler()
ENDIF

CLEAR WINDOW ALL
CLOSE PROCEDURE
CLOSE ALL
CLOSE DATABASES

*** Restablece el ambiente original de FOX ***

MODIFY WINDOWS SCREEN  &&Restablece la pantalla original de Fox
SET SYSMENU TO DEFA    &&Restablece el menu original de fox
ON ERROR               &&Restablece los mensajes de errores de fox
SET STATUS BAR ON     &&Restablece la barra de estatus
ON KEY LABEL ESC      &&Restablece la tecla ESC

RETURN

```

```
.....  
* Function: MYHANDLER()  
.....
```

```
FUNCTION myhandler  
PRIVATE m.temp, m.x
```

```
IF dropdead  
  RETURN .T.  
ENDIF  
IF LEN(tobedone) > 0  
  m.temp = tobedone  
  tobedone = ""  
  DO (m.temp)  
ELSE  
  DO ST_MAIN.SPR  
  Inicio = .F.  
ENDIF  
  
RETURN .F.
```

```
.....  
* Procedure: MHIT  
.....
```

```
PROCEDURE mhit  
PARAMETER PROG  
IF RLEVEL() > 1  
  tobedone = PROG  
  CLEAR READ  
ELSE  
  IF NIV_ACCESO(cacceso,PROG)  
    DO (PROG)  
  ENDIF  
ENDIF
```

```
.....  
* Procedure: MEXIT  
.....
```

```
PROCEDURE mexit  
dropdead = .T.
```

```
IF NotUse('SC_USERS',.F.,0)  
  SET ORDER TO TAG FOLIO  
  SEEK nUsuFolio  
  IF FOUND()  
    IF RLOCK()  
      REPLACE USU_ACTIVADO WITH .F.  
      UNLOCK  
    ENDIF  
  ENDIF  
  USE  
ENDIF  
CLEAR READ ALL  
=BITACORA('00','')  
RETURN
```

A continuación se muestra cómo generar un formulario sin utilizar el generador de pantallas de FoxPro, la figura 4.29 ilustra esta situación:

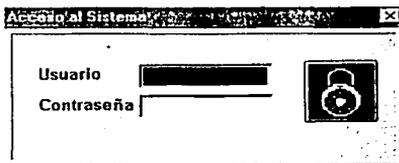


Figura 4.29 Pantalla de Acceso al Sistema

El siguiente segmento de código muestra las instrucciones que se necesitaron para poder generar la ventana de la figura 4.29, a través de la cual se controla el acceso al sistema, utilizando rutinas de encriptación y desencriptación de cadenas:

```
.....
*Sistema: SEGURIDAD DEL SISTEMA
*Programa: PASSWORD.PRG Versión: Sistema Cheques
*Autor: Israel Mendoza Pacheco
.....
```

```
PARAMETERS cUser,cLogin,cAcceso,lAcceso,cDeptoUser
```

```
*-:SCXDRV5 C:\FOXLIB\3D.PRG
*-:ALL3D
```

```
PRIVATE cDbfUser,cUsuario,cPassword,nVeces,nLeeVar
```

```
DEFINE WINDOW password ;
    AT 0.000,0.000 ;
    SIZE 8.929,60.000 ;
    TITLE "Acceso al Sistema" ;
    FONT "Arial", 8 ;
    STYLE "B" ;
    NOFLOAT ;
    NOCLOSE ;
    NOMINIMIZE ;
    COLOR RGB(...,192,192,192)
MOVE WINDOW password CENTER
```

```
cDbfUser = 'SC_USERS'
cUsuario = ""
cPassword = ""
nVeces = 1
```

```

IF IFILE(cDbfUser+'.CDX')
SELECT 0
USE (LOCFILE("SC_USERS.DBF","DBF","Buscar SC_USERS.DBF"));
AGAIN ALIAS SC_USERS EXCLUSIVE
IF USED('SC_USERS')
    INDEX ON usu_user+usu_login TAG CLAVE ADDITIVE
    INDEX ON STR(usu_banco,2,0)+usu_nombre TAG NOMBRE ADDITIVE
    INDEX ON usu_folio TAG FOLIO ADDITIVE
USE
ELSE
IAcceso = .F.
CLEAR READ
RETURN
ENDIF
ENDIF

IF WVISIBLE("password")
    ACTIVATE WINDOW password SAME
ELSE
    ACTIVATE WINDOW password NOSHOW
ENDIF

DO 3DBOX WITH 0.000,0.000,8.929,60.000,1.000,255,255,255,128,128,128
@ 0.000,0.000 SAY "" ;
FONT "", 0
DO 3DBOX WITH 0.357,1.000,8.286,58.200,-1.000,255,255,255,128,128,128
@ 0.000,0.000 SAY "" ;
FONT "", 0
@ 2.429,5.000 SAY "Usuario" ;
FONT "Arial", 10 ;
STYLE "BT"
@ 4.429,5.000 SAY "Contraseña" ;
FONT "Arial", 10 ;
STYLE "BT"
DO 3DBOX WITH 2.366,20.133,1.552,20.333,-1.8.255,255,255,96,96,96
DO 3DBOX WITH 4.366,20.133,1.481,20.333,-1.8.255,255,255,96,96,96

DO PINTA_GETS

IF NOT WVISIBLE("password")
    ACTIVATE WINDOW password
ENDIF

IAcceso = .F.
nLeeVar = 1
DO WHILE !IAcceso
    IF nLeeVar = 1
        @ 2.571,20.800 GET cUsuario ;
        SIZE 1.030,15.550 ;
        DEFAULT " " ;
        FONT "Arial", 10 ;
        STYLE "B" ;
        PICTURE "@KI XXXXXXXX" ;
        COLOR ,RGB(,,,255,255,255)

    READ
    IF READKEY() = 15 OR READKEY() = 271
        nLeeVar = 2
    ENDIF
ENDIF
ENDIF

```

```

IF nLeeVar = 2
  = Llena(4.571,20.800,@cPassword,8)
  !Acceso = ACCESO(@nVeces)
  nLeeVar = 1
ENDIF
IF nVeces > 3
  EXIT
ENDIF
ENDDO
IF !Acceso
  @ 2.143,45.000 GET btn_in ;
  PICTURE "@*BHN " + ;
  (LOCFILE("secur02.bmp","BMP|ICO|PCT|ICN","¿Dónde está secur02?"));
  SIZE 4.000,11.200,0.800 ;
  DEFAULT 1 ;
  FONT "Arial", 8 ;
  STYLE "B"
  CLEAR GET
  SET CURSOR OFF
  =INKEY(1)
  SET CURSOR ON
ENDIF
RELEASE WINDOW password
RETURN

```

```

.....
*Función: PINTA_GETS
*Objetivo: Despliega en blanco los campos get de Usuario y Password
*Autor: Israel Mendoza Pacheco
.....

```

```

PROCEDURE PINTA_GETS
*DO 3DBOX WITH 2.366,20.133,1.552,20.333,-1.8,255,255,255,96,96,96
@ 2.571,20.800 GET cUsuario ;
SIZE 1.030,15.550 ;
DEFAULT " " ;
FONT "Arial", 10 ;
STYLE "B" ;
PICTURE "@*K! XXXXXXXX" ;
COLOR ,RGB(...,255,255,255)
*DO 3DBOX WITH 4.366,20.133,1.481,20.333,-1.8,255,255,255,96,96,96
@ 4.571,20.800 GET cPassword ;
SIZE 1.000,9.500 ;
DEFAULT " " ;
FONT "FoxPrint", 11 ;
STYLE "B" ;
PICTURE "@*K! XXXXXXXX" ;
COLOR ,RGB(...,255,255,255)
@ 2.143,45.000 GET btn_out ;
PICTURE "@*BHN " + ;
(LOCFILE("secur01.bmp","BMP|ICO|PCT|ICN","¿Dónde está secur01?"));
SIZE 4.000,11.200,0.800 ;
DEFAULT 1 ;
FONT "Arial", 8 ;
STYLE "B"
CLEAR GETS
RETURN

```

```

.....
*Función: ACCESO
*Objetivo: Verifica que el usuario este registrado
*Autor: Israel Mendoza Pacheco
.....

```

```

FUNCTION ACCESO
PARAMETERS nVeces
PRIVATE lReturn,nPass,cError,cClave1,cClave2,m.busca

```

```

lReturn = .T.
cError = 'NO TIENE ACCESO AL SISTEMA'
cPass = Encrypt(cPassword)
IF NetUse(cDbfUser,.F.,0)
  SET ORDER TO TAG CLAVE
  SEEK cUsuario+cPass
  IF FOUND()
    IF USU_ACTIVO
      cError = 'EL USUARIO YA ESTA DENTRO DEL SISTEMA'
      lReturn = .f.
    ELSE
      IF RLOCK()
        nUsuFolio = USU_FOLIO
        cUser = USU_USER
        cLogin = Decrypt(ALLTRIM(USU_LOGIN))
        cAcceso = Decrypt(ALLTRIM(USU_ACCESO))
        cDeptoUser = '000000000'
        IF EMPTY(USU_DEPTO)
          cClave1 = decrypt(SUBSTR(USU_DEPTO,1,4))
          cClave2 = decrypt(SUBSTR(USU_DEPTO,5,4))
          cDeptoUser = SUBSTR(cclave1,2,5)+SUBSTR(cclave2,2,5)
          cDeptoUser = STRTRAN(cDeptoUser,'A','')
        ENDIF
        IF subs(cAcceso,3,2) = 'TB' or cDeptoUser = '9999999999'
          do st_acces.spr
        else
          m.busca = .f.
          m.BancoUser = int(val(subs(cAcceso,3,2)))
          if not used('sc_banco')
            IF netuse('sc_banco',.F.,0)
              m.busca = .t.
            ENDIF
          else
            m.busca = .t.
          endif
          if m.busca
            select sc_banco
            IF AbreIndice('SC_BANCO','CODIGO')
              seek m.bancoUser
            ELSE
              LOCATE FOR codigo = m.bancoUser
            ENDIF
            if found()
              m.nomBanco = sc_banco.nombre
            endif
          endif
          IF USED('SC_BANCO')
            USE IN SC_BANCO
          ENDIF
        endif
      ENDIF
    endif
  endif
endif

```

```

if m.bancoUser = 0
  cError = 'NO SE HA INDICADO EL BANCO QUE SE VA A OPERAR'
  IReturn = .F.
endif
if cDeptoUser = '0000000000'
  cError = 'EL USUARIO NO TIENE ACCESO A NINGUN DEPARTAMENTO'
  IReturn = .F.
endif
if cDeptoUser = '9999999999' OR EMPTY(cDeptoUser)
  cError = 'NO SE HA INDICADO EL DEPARTAMENTO QUE SE VA A OPERAR'
  IReturn = .F.
endif
SELECT SC_USERS
IF IReturn
  REPLACE USU_ACTIVADO WITH .T.
UNLOCK
ENDIF
ELSE
  IReturn = .f.
ENDIF
ENDIF
IF LEN(cAcceso) <> 5
  cAcceso = SPACE(01)+cAcceso+SPACE(01)
ENDIF
ELSE
  IReturn = .f.
ENDIF
USE
IF !IReturn
  ?CHR(7)+CHR(7)
  IF MsgBox(cError,'A T E N C I O N',21) = 4
    nVeces = nVeces + 1
    cUsuario = ""
    cPassword = ""
    DO PINTA_GETS
  ELSE
    nVeces = 4
  ENDIF
ELSE
  nVeces = 4
  IReturn = .f.
ENDIF
RETURN(IReturn)

```

```

.....
*Función: Encrypt
*Objetivo: Encripta expresión carácter
*Parámetros: cPassword : Expresión carácter a encriptar
*Autor: Israel Mendoza Pacheco
.....

```

```

FUNCTION ENCRYPT
PARAMETERS cPassword
PRIVATE nl,nLongPass,nPassword,nValCar,nPeso,nBase,nBase1,cEncrypt

```

\*\*\*\*\* Primer Paso de Encriptamiento \*\*\*\*\*

```
nBase = 37
nPassword = 0
nPeso = 0
nValCar = 0
nLongPass = LEN(ALLTRIM(cPassword))

FOR nl = 1 TO nLongPass
  IF SUBSTR(cPassword, nl, 1) $ '0123456789'
    nPeso = ASC(SUBSTR(cPassword, nl, 1)) - 47
  ELSE
    nPeso = ASC(SUBSTR(cPassword, nl, 1)) - 54
  ENDIF
  nValCar = nPeso * POTENCIA(nBase,(nl-1))
  nPassword = nPassword + nValCar
ENDFOR
```

\*\*\*\*\* Segundo Paso de Encriptamiento \*\*\*\*\*

```
nBase1 = 200
cEncrypt = ""
nValCar = 0

DO WHILE nPassword > 0
  nValCar = INT(MOD(nPassword,nBase1))+33
  cEncrypt = cEncrypt + CHR(nValCar)
  nPassword = INT(nPassword /nBase1)
ENDDO

RETURN( cEncrypt )
```

.....  
\*Función: DESENCRYPT  
\*Objetivo: Descripta una expresión carácter  
\*Parámetros: cPassword: Expresión encriptada a convertir a carácter  
\*Autor: Israel Mendoza Pacheco  
.....

```
FUNCTION DESENCRYPT
PARAMETERS cPassword
PRIVATE nl,nLongPass,nPassword,nValCar,nBase,nBase1,cDesencrypt
```

\*\*\*\*\* Primer Paso de Desencriptamiento \*\*\*\*\*

```
nBase1 = 200
nPassword = 0
nValCar = 0
nLongPass = LEN(ALLTRIM(cPassword))

FOR nl = nLongPass TO 1 STEP-1
  nValCar = ASC(SUBSTR(cPassword,nl,1))-33
  nPassword = nPassword+nValCar
  IF nl <> 1
    nPassword = nPassword*nBase1
  ENDIF
ENDFOR
```

\*\*\*\*\* Segundo Paso de Descriptamiento \*\*\*\*\*

```
.nBase = 37
cDesencrypt = "
nValCar = 0
```

```
DO WHILE nPassword > 0
  nValCar = INT(MOD(nPassword,nBase))
  IF nValCar <= 10
    nValCar = nValCar + 47
  ELSE
    nValCar = nValCar + 54
  ENDIF
  cDesencrypt = cDesencrypt + CHR(nValCar)
  nPassword = INT(nPassword/nBase)
ENDDO
RETURN( cDesencrypt )
```

.....

\*Función: POTENCIA  
\*Objetivo: Eleva la base de un número a la potencia N. Esta función es llamada por ENCRYPT()  
\*Parámetros: nBase : Número Base ; nPotencia : Potencia  
\*Autor: Israel Mendoza Pacheco

.....

```
FUNCTION POTENCIA
PARAMETER nBase,nPotencia
PRIVATE nValor
nValor = 1
IF nPotencia <> 0
  FOR nl = 1 TO nPotencia
    nValor = nValor * nBase
  ENDFOR
ELSE
  nValor = 1
ENDIF
RETURN(nValor)
```

.....

\*FUNCIÓN: LLENA  
\*OBJETIVO: Convierte el carácter tecleado en un asterisco (\*)  
\*Autor: Israel Mendoza Pacheco

.....

```
FUNCTION LLENA
PARAMETERS nRen,nCol,cVar,nLongCampo
PRIVATE nl,nKey,iRetVal,nColx
```

```
nl = 0
nKey = 0
iRetVal = .T.
```

```
SET CURSOR OFF
```

```
DO WHILE .T.
  IF EMPTY(cVar)
    @ nRen,nCol SAY Space(1)
  ENDIF
```

TESIS CON  
FALLA DE ORIGEN

```

nl= nl+1
nKey=inkey(0)
DO CASE
CASE (nKey>=48.And.nKey<=57).Or.(nKey>=65.And.nKey<=90).Or.(nKey>=97.And.nKey<=122)
cVar =IIF(nl<=nLongCampo,Alltrim(cVar+Upper(Chr(nKey))),Alltrim(cVar))
@ nRen,nCol+IIF(nl<=nLongCampo,nl,0)-1 Say Chr(42)
CASE (nKey==8 .Or. nKey==127).And.nl>1
nl = nl-2
cVar = Left(cVar,nl)
@ nRen,nCol+IIF(nl<=nLongCampo,nl,0) Say Space(1)
CASE nKey==13
If Empty(cVar)
nl = nl-1
Else
Exit
Endif
OTHERWISE
nl=nl-1
ENDCASE
If nl>=nLongCampo
Exit
Endif
ENDDO
SET CURSOR ON
Return(IRetVal)

```

Para concluir con el apartado de generación de código, se mostrarán algunas funciones que se utilizarán de manera general a lo largo de toda la creación de programas; la primer función muestra como se realiza la apertura de tablas, controlando los accesos para red; la segunda convierte una fecha en formato dd-mmm-aa; y por último, la tercera genera nombres únicos para la creación de archivos temporales.

```

.....
*Función: NetUse
*Objetivo: Apertura de Bases de Datos
*Parámetros:
* archivo - nombre del archivo a abrir
* modo - modo de apertura del archivo (.T.=Exclusivo,.F.=Compartido)
* área - área de trabajo del archivo
*Autor: Israel Mendoza Pacheco
.....

```

```

FUNCTION NetUse
PARAMETERS archivo,modo,area
PRIVATE IReturn,nVeces,cBase
IReturn = .F.
nVeces = 1

IF area <> 0
SELECT(area)

```

```

ELSE
  SELECT(area)
  area = SELECT()
ENDIF
cBase = IIF(RAT('DBF',archivo)=0,ALLTRIM(archivo)+'DBF',;
  ALLTRIM(archivo))
IF !FILE(cBase)
  =MsgBox('NO EXISTE EL ARCHIVO'+CHR(13)+ ;
    UPPER(cBase),'A T E N C I O N',16)
ELSE
  DO WHILE nVeces > 0
    IF modo=.T.
      USE (archivo) EXCLUSIVE
    ELSE
      USE (archivo) SHARED
    ENDIF
    IF USED(ALIAS(area))
      IReturn = .T.
      EXIT
    ELSE
      =INKEY(1,'H')
      nVeces = nVeces - 1
    ENDIF
  ENDDO
  IF !IReturn
    =MsgBox('NO SE TIENE ACCESO AL ARCHIVO'+CHR(13)+ ;
      UPPER(archivo),'A T E N C I O N',16)
  ENDIF
ENDIF
RETURN(IReturn)

```

.....

\*Función: FEC\_CORTA.PRG  
 \*Objetivo: Rutina que Cambia el Formato Fecha a un Formato Corto  
 \*Autor: Israel Mendoza Pacheco  
 .....

```

FUNCTION FEC_CORTA
PARAMETER mfecha
mmesletra = Space(03)
mfechac = "
mmeses = 'EneFebMarAbrMayJunJulAgoSepOctNovDic'
IF mfecha <> CTOD("")
  mmesletra = LTRIM(RTRIM(SUBSTR(mmeses,MONTH(mfecha)*3-2,3)))
  mfechac = STR(DAY(mfecha),2,0)+'-' + mmesletra+'-' + STR(YEAR(mfecha),4,0)
ENDIF
RETURN(mfechac)

```

.....

\*Función: ArchTempo  
 \*Objetivo: Genera el nombre de un archivo temporal  
 \*Parámetros: cPrefijo - Primer letra del archivo temporal  
 \*Autor: Israel Mendoza Pacheco  
 .....

```

FUNCTION ArchTempo
PARAMETERS cPrefijo
RETURN( cPrefijo+SUBSTR(SYS(3),2) )
*RETURN( cPrefijo+SUBSTR(STR(SECOND(),9,3),2,4)+;
  SUBSTR(STR(SECOND(),9,3),7,3) )

```

## 4.2 CONSTRUCCIÓN SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

Para la construcción del **Sistema Administrativo-Contable de pagos**, se seguirá la misma metodología que se implementó durante la construcción del Sistema para el Trámite de Pagos, pero enfocados en el Administrativo-Contable, iniciando de igual manera con la construcción de las relaciones, seguido de la construcción de los formularios, informes, menús y código diverso.

### 4.2.1 RELACIONES

A continuación, sólo se muestra como quedaron definidas las principales tablas del sistema:

La definición de la tabla SC\_BANCO, quedará de la siguiente forma:

```
CREATE TABLE SC_BANCO (  
    CODIGO          N(2),  
    NOMBRE          C(50),  
    RFC_BANCO      C(15),  
    DOMICILIO      C(80),  
    CHEQUERA       N(1),  
    CTA_CARGO      C(12),  
    LOTE_CONTA     C(1),  
    ID_CONTA       C(2),  
    LOGO           G(10),  
    LOGO_LARGO    G(10),  
    FEC_ALTA       D(8),  
    FEC_BAJA       D(8) )
```

La definición de la tabla SC\_DEPTO, quedará de la siguiente forma:

```
CREATE TABLE SC_DEPTO (  
    BANCO          N(2),  
    REGION         N(4),  
    PLAZA          N(6),  
    CVE_DEPTO     C(10),  
    NOM_DEPTO     C(60),  
    UBICACIÓN     C(40),  
    NOM_RESP      C(40),  
    C_COSTOS      C(5),  
    CC_IVA        C(5),  
    CC_ISR        C(5),  
    CC_RETIVA     C(5),  
    CC_CHEQUE     C(5),  
    CVE_TURNO    C(4),  
    AUTORIPAGO   L(1),  
    INI_FIRMA     C(3),  
    MONTO_MAX     N(6),  
    CAP_FECVAL   L(1),  
    FEC_ALTA     D(8),  
    FEC_BAJA     D(8) )
```

La definición de la tabla SC\_PROV, quedará de la siguiente forma:

```
CREATE TABLE SC_PROV (  
    BANCO          N(2),  
    PROVEEDOR     N(10),  
    NOM_PROV      C(50),  
    RFC           C(15),  
    CURP          C(18),  
    CALLE         C(60),  
    COLONIA       C(40),
```

CIUDAD	C(40),
COD_POSTAL	C(5),
TELÉFONO	C(12),
FAX	C(12),
NOM_COBRA1	C(40),
NOM_COBRA2	C(40),
NOM_COBRA3	C(40),
TIPO_PROV	N(1),
PAGAR_PROV	N(1),
LEYENDA_PROV	N(1),
ISR	L(1),
RETIVA	L(1),
DESCUENTO	L(1),
P_CHEQUERA	L(1),
ITEM_PROV	C(12),
FEC_ALTA	D(8),
FEC_BAJA	D(8)

La definición de la tabla SC\_CONCE, quedará de la siguiente forma:

```
CREATE TABLE SC_CONCE (
    BANCO          N(2),
    CONCEPTO     N(4),
    DESCRIP        C(50),
    VENTANILLA     L(1),
    TIENEORDEN     L(1),
    ITEM_CARGO     L(1),
    APLIABO_DE     C(1),
    ABONO_CAJA     N(4),
    ABONO_OTRO     N(4),
    ABONO_ORDE     N(4),
    GASTO1         N(4),
```

GASTO2	N(4),
GASTO3	N(4),
GASTO4	N(4),
GASTO_ORDE	N(4),
APLICA_IVA	N(4),
APLI_REIVA	N(4),
APLICA_RET	N(4),
FEC_ALTA	D(8),
FEC_BAJA	D(8) )

La definición de la tabla SC\_USERS, quedará de la siguiente forma:

```
CREATE TABLE SC_USERS (
    USU_FOLIO      N(4),
    USU_BANCO     N(2),
    USU_NOMINA    N(10),
    USU_USER      C(8),
    USU_LOGIN     C(10),
    USU_DEPTO    C(10),
    USU_ACCESO    C(10),
    USU_NOMBRE    C(40),
    USU_UBICA     C(40),
    USU_TEL       C(12),
    USU_ACTIVO    L(1),
    FEC_ALTA      D(8),
    FEC_BAJA      D(8) )
```

La definición de la tabla SC\_CONCA, quedará de la siguiente forma:

```
CREATE TABLE SC_CONCA
    BANCO         N(2),
    NO_CONRE      N(9),
```

NO_POLIZA	C(10),
CVE_DEPTO	C(10),
PROVEEDOR	N(10),
TIPO_GASTO	N(1),
CONCEPTO	N(4),
FORMA_PAGO	N(1),
TOTAL	N(12,2),
SALDO_COMP	N(9,2),
ANTICIPO	N(9,2),
COMENTARIO	C(250),
FEC_ALTA	D(8),
HORA_ALTA	C(5),
FEC_PAGO	D(8),
ESTATUS	C(1),
CVE_TURNO	C(4),
VALIDO	C(4),
FEC_CIERRE	D(8),
NO_CONTABI	C(3),
ID_CIERRE	N(6),
BLOQUEADO	L(1),
LOTE_ENVIO	C(8),
FEC_VALOR	D(8) )

La definición de la tabla SC\_CONTA, quedará de la siguiente forma:

```
CREATE TABLE SC_CONTA (
    BANCO          N(2),
    FEC_CIERRE     D(8),
    NO_CONTABI     C(3),
    LIBERACHEQ    L(1),
    FEC_POLIZA     D(8),
    FEC_EXPOR     D(8),
```

FEC_VALOR	D(8),
REGION	N(4),
PLAZA	C(6),
CVE_TURNO	C(4),
FORMA_PAGO	N(1),
CONTRA_PAG	N(4),
NUM_CHEQUE	N(4),
ABO_CHEQUE	N(14,2),
NUM_ABONOS	N(4),
ABO_CTACHE	N(14,2)

La definición de la tabla SC\_CHEQ, quedará de la siguiente forma:

```

CREATE TABLE SC_CHEQ (
    BANCO          N(2),
    FEC_CIERRE     D(8),
    NO_CONTABI     C(3),
    ID_CIERRE      N(6),
    PROVEEDOR     N(10),
    PAGAR_PROV    N(1),
    NOMBRE         C(60),
    NO_CHEQUE     N(10),
    IMPORTE       N(12,2),
    FEC_ENTREG    D(8),
    RECIBIO       C(40),
    GENERO        N(4),
    ENTREGO       N(4) )

```

La definición de la tabla SC\_CARAB, quedará de la siguiente forma:

```
CREATE TABLE SC_CARAB (  
    BANCO          N(2),  
    FEC_CIERRE     D(8),  
    NO_CONTABI     C(3),  
    ID_CIERRE      N(6),  
    CVE_DEPTO     C(10),  
    DESC_CTA       C(30),  
    ICBS_TIPO      C(1),  
    TIPODEMOVI    C(1),  
    ICBS_CTA       C(12),  
    CTA_MAYOR     C(12),  
    ICBS_SERIE     C(7),  
    IMPORTE        N(12,2),  
    NO_CHEQUE      C(10),  
    C_COSTOS       C(5) )
```

La definición de la tabla SC\_EXPOR, quedará de la siguiente forma:

```
CREATE TABLE SC_EXPOR (  
    MONEDA         C(3),  
    ICBS_CTA       C(12),  
    C_COSTOS       C(5),  
    ICBS_SERIE     C(7),  
    FEC_POLIZA     C(7),  
    TIPODEMOVI    C(1),  
    IMPORTE_MN     N(15),  
    TIP_CAMBIO     N(11),  
    IMPORTE_EX     N(15),  
    DESC_CTA       C(30),  
    IDENTIFICA     C(1) )
```

## 4.2.2 FORMULARIOS

El formulario mostrado en la figura 4.30, ejemplifica claramente que el estándar definido para la construcción de éstos se mantiene, como se puede observar, en la parte superior de la ventana se muestra el título y el menú de cascada del sistema; En la esquina superior derecha se muestran los botones de control. Así mismo, en la barra inferior se visualizan en la esquina izquierda los mensajes de ayuda y en la esquina derecha la hora y el estado de las teclas BloqMayús, BloqNúm e Insert. Por otra parte, en el formulario se puede apreciar que la barra de navegación y los botones para realizar llamadas a otros formularios se localizan en la parte inferior.

**SISTEMA ADMINISTRATIVO CONTABLE DE PAGOS >>> Recepcion de Contra Recibos <<<**

Ver pantalla | Devolucion | Salir

Títulos y Menús de Cascada | Botones Control

Banco

**Captura de Contra Recibos**

Banco	SINCA CREMI	REFERENCIA	CONTRA RECIBO
HOMBRE O RAZON SOCIAL	3	DEPTO RESPONSABLE	CENTRO COSTOS
RECIBIMOS PARA SU REVISION PAGO <input type="radio"/> Anticipo <input type="radio"/> Anticipo <input type="radio"/> Comprob. <input type="radio"/> B. Avanz.			
CONCEPTO	FORMA PAGO	IMPORTE	COMENTARIO
	<input type="radio"/> Cheques <input type="radio"/> Abono a Cta <input type="radio"/> Sin Pago	0.00	
	Anticipo	0.00	
	Saldo Comprob.	0.00	
FECHA RECEPCION	FECHA VALOR	Imp.	Tipica
FECHA DE PAGO			Comprobante
RESPONSABLE DE REVISION			

Mensajes | Desplazamiento entre registros | Llamada a Subformularios | Estado de Teclas

Inicio | Borrar | Devolucion | Alta | Baja | Guardar | Ayuda | Impresión | Salir

C43700 Record None Exchome | Ins | Num | Caps | D1014

Figura 4.30 Formulario de captura de contra-recibos

El formulario de la figura 4.31, muestra cómo a través de botones de radio y ListBox se especifican los parámetros necesarios para la selección de contra-recibos, así mismo, se puede apreciar como en la parte superior se define el título del formulario y en la parte inferior se encuentran los botones que en este caso no realizan llamadas a formularios, ni realizan el desplazamiento de registro, sino que son botones de comando que le indican al sistema una acción a realizar.

Selección de Contra-Recibos a Pagar

Por Banco  Por Región  Por Plaza

Banco: BANCA CREMI

Región:

Plaza:

Responsable de Revisión: ---TOCOS---

Fecha Programada de pago De: / / A: / /

Forma de Pago:  Cheques  Abonos  Sin Pago

Fecha de Liberación: 09/03/2003

Aceptar Cancelar Salir

Figura 4.31 Formulario de Selección de contra-recibos a pagar

Finalmente, en el formulario de la figura 4.32, se ejemplifica la manera mediante la cual se solicita al usuario ingrese el banco y la fecha de liberación de pagos o contabilización de movimientos:

TESIS CON  
FALLA DE ORIGEN

Contabilización

Banco: BANCA CREMI

Fecha de Liberación	Número

Aceptar    Salir

Figura 4.32 Formulario para Seleccionar la contabilización de pagos.

#### 4.2.3 INFORMES

Para la creación de los informes en el **Sistema Administrativo-Contable de Pagos**, se seguirán los mismos estándares que se definieron en el Sistema para el Trámite de Pagos.

Los encabezados quedarán definidos de la siguiente forma: (figura 4.33)



TESIS CON  
FALLA DE ORIGEN

COORDINACION DE CENTRAL DE PAGOS  
AREA METROPOLITANA

Figura 4.33 Estándar de Encabezados

Al igual que en los informes creados en el Sistema para el Trámite de Pagos, del lado izquierdo se desplegará el logotipo del banco; del lado derecho se mostrarán las leyendas que identificarán el informe; permitiéndole al usuario personalizar la información que se mostrará.

El pie de página quedará definido de la siguiente forma: (figura 4.34)

Figura 4.34 Estándar de Pie de Página

Como se puede observar, el pie de página sigue exactamente la misma estructura que el definido en el Sistema para el Trámite de Pagos.

La ventana de diálogo que le permitirá a los usuarios determinar hacia donde enviarán sus reportes, mantendrá el estándar definido con anterioridad, la figura 4.35 ilustra esta situación:

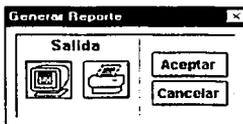


Figura 4.35 Ventana para generar los reportes

Si se elige la opción de pantalla, se mostrará la ventana de vista previa: (figura 4.36)

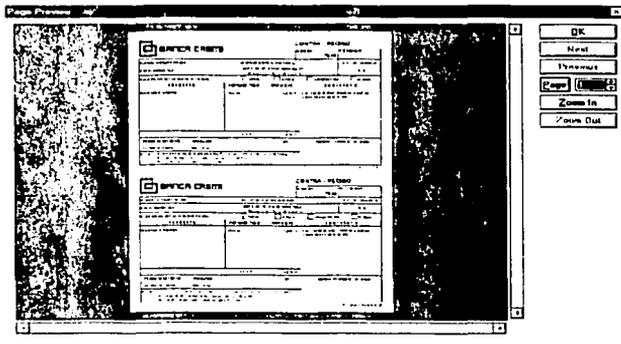


Figura 4.36 Vista previa de un reporte

Y si se elige la opción de impresora, se mostrará la ventana de diálogo para impresión de información de Windows: (figura 4.37)

Figura 4.37 Ventana de diálogo para enviar a impresora un informe

A continuación se muestra el diseño del uno de los principales informes que se implementará durante la creación del Sistema Administrativo-Contable de Pagos.

Este informe es el contra-recibo que será entregado a los proveedores que solicitan el pago de sus facturas en la Ventanilla del Área de Central de Pagos: (figura 4.38)

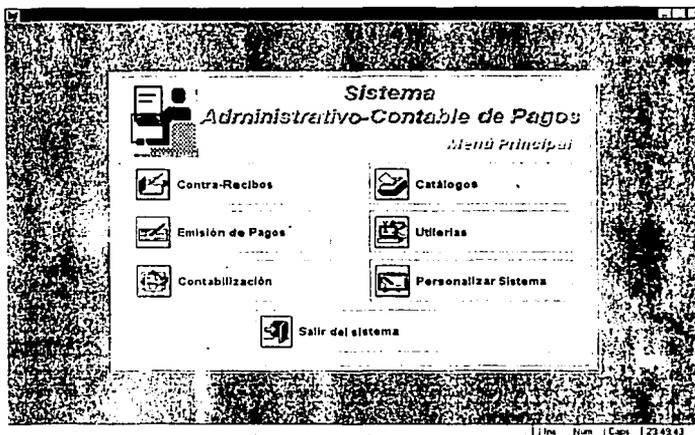
BANCA CREMI		CONTRA - RECIBO	
NOMBRE O RAZON SOCIAL		DEPARTAMENTO RESPONSABLE	C.C. QUE AUTORIZA
ANA SEGUROS, S.A.		COORD. DE RECURSOS MATERIALES HERNANDEZ Y FUENTE JORGE ALBERTO	914
RECHIMOS PARA SU REVISION Y PAGO:		<input checked="" type="checkbox"/> Adhivo	<input type="checkbox"/> Arribo <input type="checkbox"/> Cargo obsecibiv <input type="checkbox"/> B. Audn
CONCEPTO	FORMA DE PAGO	IMPORTE	COMENTARIO
SEGUROS Y FIANZAS	Checkue	1,826.11	FAC 098888803851 CORRESP A BAÑOS TAXIS POR EL MES DE OCT.
TOTAL		1,826.11	
FECHA DE RECIBIDO:	28 Oct. 2008	VIC	NOMBRE Y FIRMA DE RECIBIDO
FECHA DE PAGO:	Mier 7 Nov		
NOTA: EL RECIBO Y PAGO DE COMPROMANTES SOLO ES VÁLIDO DE ACUERDO A LO PROGRAMADO Recepción de Echeuneral Lunes a Jueves de 9:00 a 13:30 hrs. Entrega de Checkues Lunes a Jueves de 9:00 a 13:30 hrs			

Figura 4.38 Contra-recibo que entregará la Ventanilla a los Proveedores

**TESIS CON  
FALLA DE ORDEN**

#### 4.2.4 MENÚS

Para la creación del menú principal del **Sistema Administrativo-Contable de Pagos**, se utilizará un **menú de mosaico**, el cual se implementará utilizando formularios, que incluirán el uso de botones de comando para la llamada a los menús internos, a continuación se muestra el menú principal del sistema: (figura 4.39)



4.39 Menú Principal del Sistema Administrativo-Contable de Pagos

Para la creación de los menús internos del sistema, se utilizarán **menús de cascada**, los cuales se implementarán con el generador de menús de FoxPro, la figura 4.40 muestra el menú interno para la operación de la Ventanilla:

TESIS CON  
FALLA DE ORIGEN

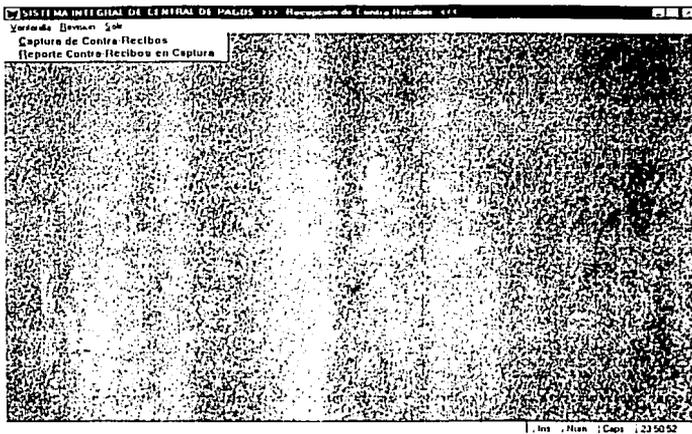


Figura 4.40 Menú de Cascada utilizado para la operación de la Ventanilla

#### 4.2.5 CÓDIGO

A continuación sólo se muestran algunos segmentos de programas utilizados dentro del Sistema Administrativo-Contable de Pagos.

Una de las principales funciones que se utilizarán para el Área de Emisión de Cheques, resulta ser el algoritmo que convierte una cantidad en letras. Este algoritmo puede ser utilizado por cualquier programa realizado en FoxPro, y si se requiere, puede importarse a otro lenguaje, cuidando se realicen las adecuaciones necesarias para la definición y sintaxis de sus comandos específicos.

A continuación se detalla el código:

TESIS CON  
FALLA DE ORIGEN

.....  
\*Función: NUM\_LETRA  
\*Objetivo: Convierte una cantidad en carácter  
\*Parámetros: nImporte : Número a convertir  
\*Autor: Israel Mendoza Pacheco  
.....

FUNCTION NUM\_LETRA  
PARAMETERS importe

PRIVATE unidades,decenas,centenas,separadores,DiezUnit  
PRIVATE Cdu1,Cdu2,Cdu3,centavos,cTxt

DIMENSION unidades(9),decenas(9),centenas(9),;  
separadores(4),diezunit(5),VeinteUnit(9)

Unidades(1) = " Un"  
Unidades(2) = " Dos"  
Unidades(3) = " Tres"  
Unidades(4) = " Cuatro"  
Unidades(5) = " Cinco"  
Unidades(6) = " Seis"  
Unidades(7) = " Siete"  
Unidades(8) = " Ocho"  
Unidades(9) = " Nueve"

Decenas(1) = " Diez"  
Decenas(2) = " Veinte"  
Decenas(3) = " Treinta"  
Decenas(4) = " Cuarenta"  
Decenas(5) = " Cincuenta"  
Decenas(6) = " Sesenta"  
Decenas(7) = " Setenta"  
Decenas(8) = " Ochenta"  
Decenas(9) = " Noventa"

Centenas(1) = " Ciento"  
Centenas(2) = " Doscientos"  
Centenas(3) = " Trescientos"  
Centenas(4) = " Cuatrocientos"  
Centenas(5) = " Quinientos"  
Centenas(6) = " Seiscientos"  
Centenas(7) = " Setecientos"  
Centenas(8) = " Ochocientos"  
Centenas(9) = " Novecientos"

Separadores(1) = " Cien"  
Separadores(2) = " Mil"  
Separadores(3) = " Millon"  
Separadores(4) = " Billion"

DiezUnit(1) = " Once"  
DiezUnit(2) = " Doce"  
DiezUnit(3) = " Trece"  
DiezUnit(4) = " Catorce"  
DiezUnit(5) = " Quince"

```
VeinteUnit(1) = " Veintiun"  
VeinteUnit(2) = " Veintidos"  
VeinteUnit(3) = " Veintitres"  
VeinteUnit(4) = " Veinticuatro"  
VeinteUnit(5) = " Veinticinco"  
VeinteUnit(6) = " Veintiseis"  
VeinteUnit(7) = " Veintisiete"  
VeinteUnit(8) = " Veintiocho"  
VeinteUnit(9) = " Veintinueve"
```

```
cdu1 = 0  
cdu2 = 0  
cdu3 = 0  
centavos = 0
```

```
cTxt = ""  
Centavos = INT(VAL(SUBSTR(ALLTRIM(STR(importe,15,2)),:  
RAT('.',ALLTRIM(STR(importe,15,2)))+1,2)))
```

```
Cdu1 = INT(INT( MOD(importe/1000,1000000) )/1000)  
Cdu2 = INT( MOD( importe / 1000 , 1000 ) )  
Cdu3 = INT( MOD( importe , 1000 ) )
```

\* los millones

```
If Cdu1 <> 0  
  cTxt = Trans_cdu( Cdu1, cTxt, 3 )  
  cTxt = cTxt + Separadores(3)  
  If Cdu1 <> 1  
    cTxt = cTxt + "ES"  
  ENDIF  
ENDIF
```

\* los miles

```
If Cdu2 <> 0  
  cTxt = Trans_cdu(Cdu2, cTxt, 2)  
  cTxt = cTxt + Separadores(2)  
ENDIF
```

\* los cientos, decenas y unidades

```
If Cdu3 <> 0  
  cTxt = Trans_cdu(Cdu3, cTxt, 1)  
ENDIF
```

```
If Cdu1 <> 0 And Cdu2 = 0 And Cdu3 = 0  
  cTxt = cTxt + " DE"  
ENDIF
```

```
cTxt = cTxt + " PESO"
```

```
If importe <> 1  
  cTxt = cTxt + "S"  
ENDIF
```

```

If Centavos = 0 Then
cTxt = cTxt + " 00/100 M.N"
Else
cTxt = cTxt + " " + STR(Centavos,2,0) + "/100 "
If Centavos > 1
cTxt = cTxt + "M.N."
ENDIF
ENDIF
cTxt = UPPER(cTxt)
RETURN(cTxt)

```

```

FUNCTION Trans_cdu
PARAMETERS Cdu,cTxt,nivel

```

```

PRIVATE u,d,c,du

```

```

u = 0
d = 0
c = 0
du = 0

```

```

u = INT( MOD( Cdu , 10 ) )
d = INT( MOD( (Cdu / 10), 10 ) )
c = INT( Cdu / 100.)
du = INT( MOD( Cdu, 100) )

```

```

If c <> 0 Then
If du = 0 And c = 1
cTxt = cTxt + Separadores(1)
Else
cTxt = cTxt + Centenas(c)
ENDIF
ENDIF

```

```

If du <> 0
DO CASE
CASE du > 10 And du < 16
cTxt = cTxt + DiezUnit(Mod(du,10))
CASE du > 20 AND du < 30
cTxt = cTxt + VeinteUnit(Mod(du,10))
OTHERWISE
If d <> 0
cTxt = cTxt + Decenas(d)
ENDIF
If u <> 0 And d <= 0
cTxt = cTxt + " Y"
ENDIF
If u <> 0 And Not(Cdu = 1 And nivel = 2)
cTxt = cTxt + Unidades(u)
ENDIF
ENDCASE
ENDIF
return(ctxt)

```

Otro programa importante es el creado para realizar la actualización del sistema, el cual se ejecuta cada vez que un usuario ingresa al sistema, este procedimiento actualiza cualquier módulo en forma automática, evitando realizar esta labor de manera personal en cada computadora.

Este mismo código incluye una sección que realiza la actualización de los catálogos utilizados dentro del Sistema para el Trámite de Pagos, la manera de identificar esta situación es enviando un parámetro que le indique al programa que sistema lo está llamando.

Desde el programa principal del sistema, se llamará el programa de actualización a través del siguiente código:

```
DO SC_UPVER.PRG WITH PROGRAM(),0,SPACE(10),cPathRoot,cPathUpFile
```

Los parámetros que se pasan son: el sistema, banco y departamento de adscripción del usuario, directorio raíz donde se encuentra instalado el sistema y por último el directorio donde se encuentran los archivos actualizados.

A continuación se detalla el código:

```
.....  
*SISTEMA : PROCEDIMIENTO DE ACTUALIZACION VERSION SISTEMA DE PAGOS  
*PROGRAMA: SC_UPVER.PRG  
*Autor: Israel Mendoza Pacheco  
.....
```

```
PARAMETERS cSistema,cBcoUser,cDeptoUser,cPathRaiz,cPathUpFile
```

```
PRIVATE IReturn,cDbfUpFiles
```

```
IReturn = .T.  
cDbfUpFiles = cPathUpFile+'UP_FILES.DBF'
```

```
WAIT WINDOWS 'UN MOMENTO POR FAVOR, VERIFICANDO SISTEMA....' NOWAIT
```

```
IReturn = IIF(IReturn,NetUse('SC_FILES',F.,0),IReturn)
```

```
IF IReturn AND !FILE(cDbfUpFiles)  
  SELECT SC_FILES  
  COPY STRUC TO &cDbfUpFiles  
ENDIF
```

```
IReturn = IIF(IReturn,NetUse('UP_FILES',.F.,0),JReturn)
```

```
IF IIReturn
```

```
=MsgBox('NO FUE POSIBLE ACTUALIZAR EL SISTEMA'+CHR(13)+;  
'VERIFIQUE CON EL ADMINISTRADOR DEL SISTEMA, LA UBICACION '+  
'DEL ARCHIVO SC_FILES.DBF';A T E N C I O N',16)
```

```
ELSE
```

```
SELECT SC_FILES
```

```
GO TOP
```

```
SCAN FOR ALLTRIM(sistema) = ALLTRIM(cSistema)
```

```
SCATTER MEMVAR
```

```
SELECT UP_FILES
```

```
LOCATE FOR sistema=m.sistema AND archivo=m.archivo AND tipo=m.tpo
```

```
IF !FOUND()
```

```
IF FLOCK()
```

```
INSERT INTO UP_FILES FROM MEMVAR
```

```
UNLOCK
```

```
ENDIF
```

```
ELSE
```

```
IF RLOCK()
```

```
REPLACE origen WITH SC_FILES.origen, ;
```

```
destino WITH SC_FILES.destino, ;
```

```
modulo WITH SC_FILES.modulo
```

```
UNLOCK
```

```
ENDIF
```

```
ENDIF
```

```
SELECT SC_FILES
```

```
ENDSCAN
```

```
IActualizaDbf = .F.
```

```
SELECT UP_FILES
```

```
REPLACE actualizar WITH .T. ALL
```

```
GO TOP
```

```
SCAN FOR ALLTRIM(sistema) = ALLTRIM(cSistema)
```

```
SCATTER MEMVAR
```

```
WAIT WINDOWS 'UN MOMENTO POR FAVOR, VERIFICANDO SISTEMA....' NOWAIT
```

```
cRutaOrigen = IIF(!EMPTY(origen),&origen,")
```

```
IF !EMPTY(cRutaOrigen)
```

```
IF !EMPTY(destino)
```

```
cRutaDestino = &destino
```

```
ELSE
```

```
cRutaDestino = cPathRaiz
```

```
ENDIF
```

```
IF tipo <> 'FPT'
```

```
cFile_ori = ALLTRIM(cRutaOrigen)+ALLTRIM(archivo)+'.'+ALLTRIM(tipo)
```

```
cFile_des = ALLTRIM(cRutaDestino)+ALLTRIM(archivo)+'.'+ALLTRIM(tipo)
```

```
ELSE
```

```
cFile_ori = ALLTRIM(cRutaOrigen)+ALLTRIM(archivo)+'.DBF'
```

```
cFile_des = ALLTRIM(cRutaDestino)+ALLTRIM(archivo)+'.DBF'
```

```
ENDIF
```

```
DIMENSION aFile_ori(1)
```

```
DIMENSION aFile_des(1)
```

```
DIMENSION aRuta_ori(1)
```

```
DIMENSION aRuta_des(1)
```

```
aFile_ori = F.
```

```
aFile_des = F.
```

```
aRuta_ori = F.
```

```
aRuta_des = .F.
```

TESIS CON  
FALLA DE ORIGEN

```

=ADIR(aRuta_ori,cRutaOrigen+*.*)
=ADIR(aRuta_des,cRutaDestino+*.*)
=ADIR(aFile_ori,cFile_ori)
=ADIR(aFile_des,cFile_des)
IF TYPE('aRuta_ori(1,1)')='C' AND TYPE('aRuta_des(1,1)')='C' AND TYPE('aFile_ori(1,1)')='C'
  IF TYPE('aFile_des(1,1)')<>'C'
    DIMENSION aFile_des(1,5)
    aFile_des(1,1) = "
    aFile_des(1,2) = 0
    aFile_des(1,3) = {}
    aFile_des(1,4) = '00:00:00'
    aFile_des(1,5) = "
  ENDIF
  IF aFile_ori(1,2)<>aFile_des(1,2) OR ;
  aFile_ori(1,3)<>m.fec_actual OR aFile_ori(1,4)<>m.hor_actual
  IF TYPE('aRuta_des(1,1)')='C' AND modulo=0 AND actualizar
    WAIT WINDOWS 'ACTUALIZANDO ARCHIVO '+ALLTRIM(archivo)+'+ALLTRIM(tipo) NOWAIT
  DO CASE
    CASE m.tipo = 'DBF'
      !ActualizaDbf = .T.
      COPY FILE &cFile_ori TO &cFile_des
      IF NetUse(cFile_des,.T.,0)
        DELETE TAG ALL
      USE
      cFile_des = SUBSTR(cFile_des,1,RAT('.',cFile_des))+CDX'
      IF FILE(cFile_des)
        ERASE (cFile_des)
      ENDIF
    CASE m.tipo = 'FPT'
      !ActualizaDbf = .T.
      COPY FILE &cFile_ori TO &cFile_des
      cFile_ori = SUBSTR(cFile_ori,1,RAT('.',cFile_ori))+FPT'
      cFile_des = SUBSTR(cFile_des,1,RAT('.',cFile_des))+FPT'
      COPY FILE &cFile_ori TO &cFile_des
      cFile_des = SUBSTR(cFile_des,1,RAT('.',cFile_des))+DBF'
      IF NetUse(cFile_des,.T.,0)
        DELETE TAG ALL
      USE
      cFile_des = SUBSTR(cFile_des,1,RAT('.',cFile_des))+CDX'
      IF FILE(cFile_des)
        ERASE (cFile_des)
      ENDIF
    OTHERWISE
      COPY FILE &cFile_ori TO &cFile_des
  ENDCASE
  SELECT UP_FILES
  IF RLOCK()
    REPLACE fec_actual WITH aFile_ori(1,3) ;;
    hor_actual WITH aFile_ori(1,4) ;;
    actualizar WITH .F.
  UNLOCK
  ENDIF
ENDIF
WAIT CLEAR
ELSE
  SELECT UP_FILES
  IF RLOCK()

```

**TESIS CON  
FALLA DE ORIGEN**

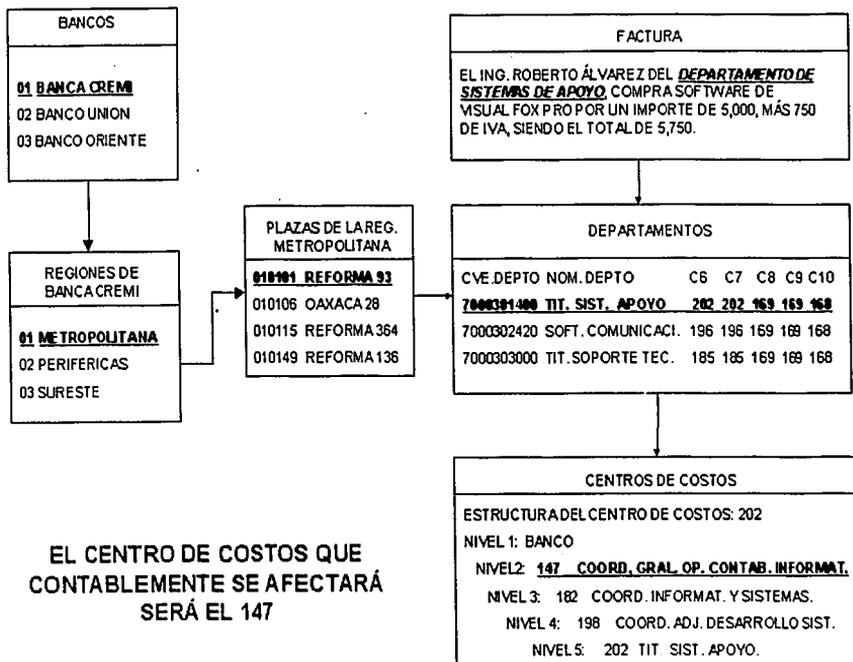
```

REPLACE actualizar WITH .F.
UNLOCK
ENDIF
ENDIF
ENDIF
IF m.tipo = 'ERR'
  !ExisteErr = .F.
  cFile_ori = SUBSTR(cFile_ori,1,RAT(';',cFile_ori))+'.DBF'
  IF NetUse(cFile_ori,.F.)
    SELECT * FROM &cFile_ori ;
    WHERE banco = cBcoUser AND depto_usu = cDeptoUser AND ;
      estatus<>'R' ;
    INTO CURSOR dbfError
    SELECT dbfError
    GO TOP
    IF !EOF()
      !ExisteErr = .T.
    ENDIF
    USE
    = SELECT(cFile_ori)
    USE
  ENDIF
  IF !ExisteErr
    =MsgBox('EXISTEN ERRORES EN LOS LOTES ENVIADOS A CENTRAL DE PAGOS. '+'
      'VERIFIQUE SUS LOTES.!' + ' T E N C I O N',16) .
  ENDIF
ENDIF
ENDIF
SELECT UP_FILES
ENDSCAN
IF !ActualizaDbf
  =MsgBox('SE HAN ACTUALIZADO LAS BASES DE DATOS DEL SISTEMA, '+'CHR(13)+'
    'POR FAVOR GENERE LOS INDICES RESPECTIVOS.!' + ' T E N C I O N',64)
ENDIF
ENDIF
IF USED('UP_FILES')
  USE IN UP_FILES
ENDIF
IF USED('SC_FILES')
  USE IN SC_FILES
ENDIF
WAIT CLEAR
RETURN (.T.)

```

Para concluir con este apartado, sólo se mostrará parte del código utilizado para realizar la afectación contable; primeramente se mostrará gráficamente la forma en que se obtendrán los centros de costos (figura 4.41) y la forma en que se realizará la afectación de las cuentas contables (figura 4.42), para posteriormente mostrar partes del código.

# Obtención Centro de Costos

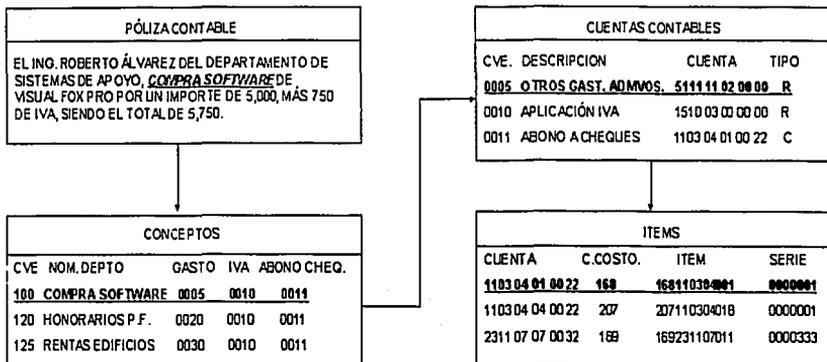


EL CENTRO DE COSTOS QUE  
CONTABLEMENTE SE AFECTARÁ  
SERÁ EL 147

Figura 4.41 Procedimiento para la obtención del Centro de Costos

TESIS CON  
FALLA DE ORIGEN

## Aplicación de Cuentas Contables



<u>C.C.</u>	<u>CUENTA CONTABLE</u>	<u>CARGOS</u>	<u>ABONOS</u>
147	5111 11 02 00 00	5,000	
147	1510 03 00 00 00	750	
168	168 1103 04 001		5,750

Figura 4.42 Procedimiento para realizar la aplicación de las Cuentas Contables

TESIS CON  
 FALLA DE ORIGEN

El siguiente segmento de código muestra como se obtienen los centros de costos:

**\*\*APLICACION DE CENTROS DE COSTO DE ACUERDO AL DEPARTAMENTO**

```
SELECT SC_DEPTO
SEEK STR(mBancoConta,2,0)+mCveDeplo
IF FOUND()
  mcc_gasto = SC_DEPTO.c_costos
  mcc_iva   = SC_DEPTO.cc_iva
  mcc_isr   = SC_DEPTO.cc_isr
  mcc_retiva = SC_DEPTO.cc_retiva
  mcc_cheque = SC_DEPTO.cc_cheque
  mBanco    = SC_DEPTO.banco
  mPlaza    = SC_DEPTO.plaza
  mRegion   = SC_DEPTO.region
  IF EMPTY(mcc_gasto)
    DO HAY_ERROR WITH 15,'M',mgasto,mdesc_conce,;
    mproveedor,mnom_prov
  ENDIF
ELSE
  DO HAY_ERROR WITH 8,'M',mgasto,mdesc_conce,mproveedor,mnom_prov
ENDIF

IF mbanco = 0 OR mregion=0 OR EMPTY(mplaza)
  DO HAY_ERROR WITH 9,'M',mgasto,mdesc_conce,mproveedor,mnom_prov
ENDIF
```

El siguiente segmento de código muestra como se realiza la afectación contable del gasto, del IVA, de la retención de IVA e ISR, así como la manera de almacenar esta información en la tabla de Cargos y Abonos:

**&& APLICACION DEL GASTO ORIGINADO**

```
DO CASE
CASE SP_MOVTO.iva<>0 AND SP_MOVTO.ret_isr<>0
  ncvcuenta = SC_CONCE.gasto1
CASE SP_MOVTO.iva<>0 AND SP_MOVTO.ret_isr=0
  ncvcuenta = SC_CONCE.gasto2
CASE SP_MOVTO.iva=0 AND SP_MOVTO.ret_isr<>0
  ncvcuenta = SC_CONCE.gasto3
CASE SP_MOVTO.iva=0 AND SP_MOVTO.ret_isr=0
  ncvcuenta = SC_CONCE.gasto4
OTHERWISE
  ncvcuenta = 0
ENDCASE
```

**TESIS CON  
FALLA DE ORIGEN**

```
IF ncveuenta <> 0
  DO movcontable WITH 'G',mBancoConta,mCveDepto,ncveuenta,;
    mgasto,mNo_cheque,mcc_gasto,;
    mnom_prov,mFec_valor,.F.
ELSE
  DO HAY_ERROR WITH 1,'M',mgasto,SC_CONCE.descripcion,mproveedor,mnom_prov
ENDIF
```

#### && APLICACION DEL IVA

```
ncveuenta = 0
IF SP_MOVTO.iva<>0
  ncveuenta = SC_CONCE.aplica_iva
  IF ncveuenta <> 0
    DO movcontable WITH 'IVA',mBancoConta,mCveDepto,ncveuenta,;
      miva,mNo_cheque,mcc_iva,;
      mnom_prov,mFec_valor,.F.
  ELSE
    DO HAY_ERROR WITH 2,'M',miva,SC_CONCE.descripcion,mproveedor,mnom_prov
  ENDIF
ENDIF
```

#### && APLICACION DE LA RETENCION DEL IVA

```
ncveuenta = 0
IF mretiva AND IEMPTY(mlva)
  ncveuenta = SC_CONCE.apli_reiva
  IF ncveuenta <> 0
    DO movcontable WITH 'RIVA',mBancoConta,mCveDepto,ncveuenta,;
      mret_iva,mNo_cheque,mcc_retiva,;
      mnom_prov,mFec_valor,.F.
  ELSE
    DO HAY_ERROR WITH 12,'M',mret_iva,SC_CONCE.descripcion,;
      mproveedor,mnom_prov
  ENDIF
ENDIF
```

#### && APLICACION DE LA RETENCION

```
ncveuenta = 0
IF SP_MOVTO.ret_isr<>0
  ncveuenta = SC_CONCE.aplica_ret
  IF ncveuenta <> 0
    DO movcontable WITH 'ISR',mBancoConta,mCveDepto,ncveuenta,;
      mretencion,mNo_cheque,mcc_isr,;
      mnom_prov,mFec_valor,.F.
  ELSE
    DO HAY_ERROR WITH 3,'M',mretencion,SC_CONCE.descripcion,;
      mproveedor,mnom_prov
  ENDIF
ENDIF
```

```

.....
* APLICACION DE MOVIMIENTOS CONTABLES
cTipoMov = G: GASTO, IVA, RIVA, ISR
*
*   A: ABONO, ACV: ABONO COMPROBACION VIATICOS
*   SACV: SALDO ABONO COMPROBACION VIATICOS
*   AOC : ABONO OTRA COMPROBACION
*   SAOC : SALDO ABONO OTRA COMPROBACION

```

```

*Autor: Israel Mendoza Pacheco
.....

```

```

PROCEDURE movcontable

```

```

PARAMETERS cTipoMov,mBancoConta,mCveDepto,ncv cuenta,nimporte,mNo_cheque,;
ccostos,mnom_prov,mFec_valor,ICheque

```

```

PRIVATE mItem,mSerie,mCuenta,mCta_Mayor,micbs_tipo,mTipodemovi

```

```

mItem      = ""
mSerie     = ""
mCta_Mayor = ""

```

```

IF EMPTY(nimporte)
RETURN
ENDIF

```

```

SELECT SC_CTAS
SEEK STR(mBancoConta,2,0)+STR(ncvCuenta,4,0)
micbs_tipo = SC_CTAS.icbs_tipo
mTipodemovi = SC_CTAS.tipodemovi

```

```

IF SC_CTAS.icbs_tipo='C'
mCuenta = cuenta+sub1+sub2+sub3+sector
SELECT SC_ITEMS
DO CASE

```

```

CASE mTiene_item AND (cTipoMov='G' OR cTipoMov='ACV')
SET ORDER TO TAG ITEM
IF mItem_prov = mItem_docto
SEEK STR(mBancoConta,2,0)+mItem_prov
ELSE
SEEK STR(mBancoConta,2,0)+mItem_docto
ENDIF
ENDIF
IF FOUND()

```

```

mCta_Mayor = SC_ITEMS.cuenta
mItem      = SC_ITEMS.item
ccostos    = SC_ITEMS.c_costos
IF SC_ITEMS.serie_fija
mSerie     = SC_ITEMS.serie
ELSE
mSerie     = mSerie_Fac
ENDIF
ENDIF

```

```

ELSE
DO HAY_ERROR WITH 17,'M',nimporte,SC_CTAS.nombre,;
mproveedor,mnom_prov
ENDIF

```

```

CASE ( mTieneOrden AND (cTipoMov='CORDEN') ) OR ;
( !EMPTY(mItem_docto) AND cTipoMov='G' ) OR ;
( !EMPTY(mItem_docto) AND cTipoMov='AOC' )
SET ORDER TO TAG ITEM
SEEK STR(mBancoConta,2,0)+mItem_docto
IF FOUND()

```

TESIS CON  
 FALLA DE ORIGEN

```

mCta_Mayor = SC_ITEMS.cuenta
mItem      = SC_ITEMS.item
ccostos   = SC_ITEMS.c_costos
IF SC_ITEMS.serie_fija
  mSerie = SC_ITEMS.serie
ELSE
  mSerie = mSerie_Fac
ENDIF
ELSE
DO HAY_ERROR WITH 22,'M',nimporte,SC_CTAS.nombre,;
  mproveedor,mnom_prov
ENDIF
OTHERWISE
SET ORDER TO TAG CUENTA
SEEK STR(mBancoConta,2,0)+mCuenta+STR(nCveCuenta,4,0)+ccostos
IF FOUND()
  mCta_Mayor = SC_ITEMS.cuenta
  mItem      = SC_ITEMS.item
  mSerie     = SC_ITEMS.serie
ELSE
DO HAY_ERROR WITH 16,'M',nimporte,SC_CTAS.nombre,;
  mproveedor,mnom_prov
ENDIF
ENDCASE
ENDIF
IF (cTipoMov = 'SACV' AND SP_FACTU.tipo_gasto = 3 AND SP_FACTU.viatico) OR;
cTipoMov = 'SAOC'
IF nimporte > 0
  IF mTipodemovi='6'
    mTipodemovi='0'
  ELSE
    mTipodemovi='6'
  ENDIF
ENDIF
ENDIF
IF mTieneOrden AND cTipoMov='AORDEN'
  SELECT SC_ITEMS
  SET ORDER TO TAG ITEM
  SEEK STR(mBancoConta,2,0)+mItem_docto
  IF FOUND()
    ccostos = SC_ITEMS.c_costos
  ELSE
    DO HAY_ERROR WITH 22,'M',nimporte,SC_CTAS.nombre,;
      mproveedor,mnom_prov
  ENDIF
ENDIF
SELECT SP_CARAB
APPEND BLANK
REPLACE banco WITH mBancoConta, ;
fec_cierre WITH mfec_cierre, ;
no_contabi WITH mno_contabi, ;
id_cierre WITH mid_cierre, ;
fec_valor WITH mFec_valor, ;
region WITH mregion, ;

```

```

plaza WITH mplaza ;
cve_depto WITH mCveDepto ;
importe WITH ABS(nImporte) ;
desc_cta WITH IIF(EMPTY(mnom_prov),mnom_prov,SC_CTAS.nombre) ;
no_cheque WITH mNo_cheque ;
icbs_tipo WITH mlcbs_tipo ;
tipodemovi WITH mTipodemovi ;
c_costos WITH PADL(ALLTRIM(ccostos),5,'0') ;
cta_cheque WITH .F.

IF ICheque
  REPLACE desc_cta WITH mNo_cheque ;
  cta_cheque WITH .T.
ENDIF
IF mlcbs_tipo = 'R'
  REPLACE icbs_cta WITH SC_CTAS.cuenta+SC_CTAS.sub1+SC_CTAS.sub2+
    SC_CTAS.sub3+SC_CTAS.sector
ELSE
  REPLACE cve_mayor WITH nCveCuenta ;
  cta_mayor WITH mCta_mayor ;
  icbs_cta WITH mltem ;
  icbs_serie WITH mSerie
ENDIF
RETURN

```

El siguiente segmento de código muestra la forma en que son registrados los errores contables, cuándo se aplican mal los movimientos:

```

* GRABA LOS ERRORES ENCONTRADOS EN LOS MOV. CONTABLES
*Autor: Israel Mendoza Pacheco
*
PROCEDURE HAY_ERROR
PARAMETERS merror.cTipo,mimporte,mdesc_conce,mproveedor,mnom_prov

SELECT SP_ERROR
APPEND BLANK
REPLACE banco WITH SP_FACTU.banco ;;
  fec_cierre WITH mFec_cierre ;;
  no_contabi WITH mNo_contabi ;;
  no_conre WITH SP_FACTU.no_conre ;;
  no_poliza WITH SP_FACTU.no_poliza

IF cTipo='F'
  REPLACE concepto WITH SP_FACTU.concepto
ELSE
  REPLACE concepto WITH SP_MOVTO.concepto
ENDIF
REPLACE proveedor WITH SP_FACTU.proveedor
REPLACE nom_prov WITH mnom_prov
REPLACE desc_conce WITH mdesc_conce
REPLACE importe WITH mimporte

```

```

DO CASE
CASE merror = 1
  REPLACE ERROR WITH 'GASTO NO APLICADO (VER. CAT. CONCEPTOS)'
CASE merror = 2
  REPLACE ERROR WITH 'IVA NO APLICADO (VER. CAT. CONCEPTOS)'
CASE merror = 3
  REPLACE ERROR WITH 'RETENCION NO APLICADA (VER. CAT. CONCEPTOS)'
CASE merror = 4
  REPLACE ERROR WITH 'ABONO A CTA CHEQ. NO APLICADO (VER. CAT. CONCEPTOS)'
CASE merror = 5
  REPLACE ERROR WITH 'ABONO A CHEQUES NO APLICADO (VER. CAT. CONCEPTOS)'
CASE merror = 6
  REPLACE ERROR WITH 'CLAVE DE CONCEPTO NO EXISTE (VER. CAT. CONCEPTOS)'
CASE merror = 7
  REPLACE ERROR WITH 'CLAVE DE PROVEEDOR NO EXISTE'
CASE merror = 8
  REPLACE ERROR WITH 'CLAVE DE DEPARTAMENTO NO EXISTE'
CASE merror = 9
  REPLACE ERROR WITH 'REGIONALIZACION DEL DEPARTAMENTO INCORRECTA'
CASE merror = 10
  REPLACE ERROR WITH 'NO EXISTEN ABO. CTA CHEQ PROV'
CASE merror = 11
  REPLACE ERROR WITH 'NO EXISTEN ABO. CTA CHEQ EMPLEADO'
CASE merror = 12
  REPLACE ERROR WITH 'RETENCION DEL IVA NO APLICADO'
CASE merror = 13
  REPLACE ERROR WITH 'NUMERO DE NOMINA NO EXISTE'
CASE merror = 14
  REPLACE ERROR WITH 'CVE. DEPTO. MAL REGISTRADA EN EL CAT. DE PLAZAS'
CASE merror = 15
  REPLACE ERROR WITH 'CEN.COSTO MAL REGISTRADO EN EL CAT. DE DEPTOS.'
CASE merror = 16
  REPLACE ERROR WITH 'ITEM NO EXISTE'
CASE merror = 17
  REPLACE ERROR WITH 'EL ITEM INDICADO EN EL CAT. DE PROV. NO EXISTE'
CASE merror = 18
  REPLACE ERROR WITH 'NO EXISTE EL ANTICIPO DEL MOVIMIENTO INDICADO'
CASE merror = 19
  REPLACE ERROR WITH 'ABONO EN OTRA CTA NO APLICADO (VER.CAT.CONCEPTOS)'
CASE merror = 20
  REPLACE ERROR WITH 'CARGO EN CTA.ORDEN NO APLICADO (VER.CAT.CONCEPTOS)'
CASE merror = 21
  REPLACE ERROR WITH 'ABONO EN CTA.ORDEN NO APLICADO (VER.CAT.CONCEPTOS)'
CASE merror = 22
  REPLACE ERROR WITH 'EL ITEM INDICADO EN LA FACTURA NO EXISTE'
ENDCASE
RETURN

```



## 5.1 PRUEBAS Y AJUSTES

La fase de pruebas de cualquier sistema es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación.

Los objetivos que se deben cumplir en la fase de pruebas, quedan perfectamente establecidos siguiendo una serie de reglas propuestas por Glen Myers:

- a) La prueba es un proceso de ejecución de un programa con la intención de descubrir cualquier error.
- b) Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto.
- c) Una prueba tiene éxito si descubre un error no descubierto hasta entonces.

Considerando lo antes expuesto, el objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

El flujo de información deberá contar con dos clases de entradas al proceso de pruebas:

- a) Una configuración del software que incluya la especificación de los requisitos del sistema, y
- b) Una configuración de pruebas que incluya un plan y procedimiento a seguir.

El proceso de prueba deberá estar apoyado en métodos de diseño de casos de prueba, que proporcionen un enfoque sistemático a ésta, y que proporcionen los mecanismos de ayuda para asegurar su completitud, para conseguir la mayor probabilidad de descubrimiento de errores en el software.

### 5.1.1 TIPOS DE PRUEBAS

Antes de la liberación del sistema y como última fase del desarrollo, el sistema fue sometido a una serie de procesos de validación, con el propósito de determinar el grado de cumplimiento de las especificaciones definidas por el usuario, para tal efecto se llevaron a cabo diversos tipos de pruebas, siendo las más importantes:

#### **Pruebas de unidad y de integración**

Las pruebas de unidad, también conocidas como de "Caja Blanca", se enfocan hacia la estructura de control del programa, es decir, conociendo el funcionamiento del programa, se deberá validar que la operación interna se ajuste a las especificaciones inicialmente definidas y que todos los componentes internos se hayan comprobado de forma adecuada.

Utilizando este método se derivaron los siguientes casos de prueba:

- Para cada módulo del sistema (Captura de documentos, validación y firma, generación de lotes, envío de información al Área de Central de Pagos, Operación de la Ventanilla, Emisión de Pagos, Afectación contable) se ejercitaron todos los posibles caminos independientes de cada uno de éstos.
- Se ejercitaron todas las decisiones lógicas en sus valores verdaderos y falsos, haciendo énfasis en los ciclos de búsquedas de información.
- Con la ayuda del manejador de datos de Foxpro se verificaron las estructuras de datos que son utilizadas por cada módulo para asegurar su validez.

Una vez que se han realizado las pruebas por separado de cada módulo, el siguiente paso es realizar pruebas de integración, que básicamente consisten en conjuntar incrementalmente los módulos con la finalidad de detectar posibles errores.

Para realizar la integración de los módulos, se utilizó la técnica de integración incremental descendente, que consistió en tomar el módulo principal y a partir de este añadir los módulos subordinados, para realizar estas pruebas se fueron creando formularios en blanco que sustituyeron a los formularios originales, con lo cual se garantizó que las llamadas y regresos entre formularios fueran correctos, posteriormente se añadieron los formularios originales y se verificó que las funciones de los botones de comando operaran adecuadamente.

### Pruebas de validación

Se refieren a las pruebas que se llevan a cabo sobre la interface, las cuales son utilizadas para demostrar que las funciones operativas del sistema son correctas, es decir, que las entradas de información se realicen de forma correcta y que las salidas sean las esperadas, a este tipo de pruebas también se les conoce como de "Caja Negra", y tienen como finalidad encontrar errores de las siguientes categorías:

- Funciones incorrectas o inexistentes.
- Errores de la interface.
- Errores en las estructuras de datos o accesos erróneos a bases de datos externas.
- Errores de rendimientos.
- Errores de inicialización o terminación.

Para la elaboración de los casos de pruebas del **Sistema para el Trámite de Pagos** se seleccionaron documentos de los diferentes movimientos que originan gasto (pago de luz, agua, predial, estacionamiento, papelería, copias certificadas, pasajes, viáticos, comprobaciones, etc.) y se adaptó un equipo para simular la captura que deberán hacer los Departamentos. Además, se capacitó a una persona sobre los procesos que se deberán seguir para la captura, validación y autorización de facturas, hasta llegar al último proceso de asignación de los lotes de envío.

Para la elaboración de los casos de pruebas del **Sistema Administrativo-Contable de Pagos** se seleccionó el lote de facturas generado en las pruebas del Sistema para el Trámite de Pagos, se realizó la carga de estos datos en un equipo del Área de Central de Pagos que se acondicionó para simular la Ventanilla, la selección y generación de cheques, la afectación contable y la generación de las pólizas contables, hasta culminar con la elaboración del archivo de transferencia contable, para tal efecto, se capacitó al personal responsable de cada proceso y se verificó que los datos obtenidos por el sistema fueran los correctos.

## **5.2 CAPACITACIÓN DEL USUARIO**

Durante la implementación del software creado, los analistas de sistemas se deberán involucrar en un proceso educacional con los usuarios llamado capacitación.

En esta etapa será fundamental transmitir a los usuarios del nuevo software las características y virtudes del sistema, por lo que se deberá capacitar a todas las personas que tendrán uso primario o secundario del sistema, considerando para tal efecto, desde el personal de captura de datos, hasta aquellos que usarán los reportes de salida para la toma de decisiones, aunque éstos últimos no hagan uso personal del sistema.

Para tal efecto, hay que asegurarse de que estén separados usuarios de diferentes niveles de habilidades e intereses de trabajo. Es ciertamente problemático incluir novatos en las mismas sesiones de capacitación con los expertos, debido a que los novatos se pierden rápidamente y los expertos rápidamente se aburren con los puntos básicos, dando como resultado que ambos grupos queden perdidos.

La capacitación deberá tener como objetivo principal el documentar a los usuarios a cerca del funcionamiento y recorrido a través de las pantallas, así como darles a conocer la forma de cómo podrán utilizar el equipo para la ejecución del sistema.

Para lograr los objetivos de la capacitación, se deberán establecer los siguientes lineamientos:

- Establecimiento de objetivos mensurables.
- Uso de métodos de capacitación adecuados.
- Selección de lugares de capacitación adecuados.
- Empleo de materiales de capacitación comprensibles.

Para llevar a cabo la consecución de los lineamientos establecidos, el analista se deberá apoyar en:

- Un Diagrama General del Sistema: Este diagrama deberá ser presentado gráficamente y de forma sencilla, explicando clara y brevemente todos los módulos del sistema.
- Diagramas de Pantallas: Deberán mostrar el flujo de información que es utilizado en cada pantalla, considerando:
  - Una explicación del recorrido que se debe seguir para llegar a éstas.
  - El formato de los datos a capturar.
  - Los datos e informes de salida.

Para la capacitación de los usuarios se realizaron dos grupos principales uno para el Sistema para el Trámite de Pagos y otro para el Sistema Administrativo-Contable de Pagos.

Como material de apoyo se presentó una serie de diapositivas en Power Point, en las cuales se mostraron los antecedentes que originaron la creación del sistema, los objetivos que se pretenden obtener, diagramas de flujo de información y de procesos, haciendo hincapié en las ventajas que ofrece el sistema.

Se acondicionó una oficina en el Área de Central de Pagos dotándola de un proyector con el objeto de poder mostrar la operación del sistema, obteniendo así la atención de todos los usuarios, con la intención de recabar sus dudas y comentarios.

En lo que se refiere al grupo de usuarios del Sistema para el Trámite de Pagos, solamente se capacitó a cierto personal del Área de Central de Pagos, los cuales, posteriormente con base a un calendario establecido por ellos, dieron inicio con la capacitación de los diferentes usuarios de los Departamentos que originan gasto.

### 5.3 DOCUMENTACIÓN DEL SISTEMA

La documentación de sistemas es el conjunto de información que nos dice qué hacen los sistemas, cómo lo hacen y para quién lo hacen.

La documentación consiste en material escrito que explica las características técnicas y la operación de un sistema. Es esencial para proporcionar entendimiento de un sistema a quien lo vaya a usar para mantenerlo, para permitir auditoria del sistema y para enseñar a los usuarios como interactuar con éste y a los analistas de sistemas como hacerlo funcionar.

Existen varios tipos de documentación. La de programas, que explica la lógica de un programa e incluye descripciones, diagramas de flujo, listados de programas y otros documentos; la de los usuarios, que describen de forma general la naturaleza y capacidades del sistema y cómo usarlo.

La importancia de la documentación bien podría ser comparada con la importancia de la existencia de una Póliza de Seguro; mientras todo va bien no existe la precaución de confirmar si nuestra Póliza de Seguros está o no vigente.

La documentación adecuada y completa de una aplicación que se desea implantar, mantener y actualizar en forma satisfactoria, es esencial en cualquier Sistema de Información, sin embargo, frecuentemente es la parte a la cual se dedica el menor tiempo y se le presta menos atención.

### 5.3.1 MANUAL DEL USUARIO

Expone los procesos que el usuario puede realizar con el sistema implantado. Para lograr esto, es necesario que se detallen todas y cada una de las características que tienen los programas y la forma de acceder e introducir información. Permite a los usuarios conocer el detalle de las actividades que ellos deberán desarrollar para la consecución de los objetivos del sistema. Reúne la información, normas y documentación necesaria para que el usuario conozca y utilice adecuadamente la aplicación desarrollada.

Los objetivos de este manual son:

- Que el usuario conozca cómo preparar los datos de entrada.
- Que el usuario aprenda a obtener los resultados y los datos de salida.
- Deberá funcionar como manual de aprendizaje.
- Deberá servir como manual de referencia.
- Deberá definir las funciones que debe realizar el usuario.
- Deberá informar al usuario de la respuesta a cada mensaje de error.

La importancia del Manual de Usuario radica en que facilita el conocimiento de:

- Los documentos a los que se puede dar entrada por computadora.
- Los formatos de los documentos.
- Las operaciones que utilizan entradas y salidas de los datos.
- El orden del tratamiento de la computadora con los datos introducidos.
- El momento en que se debe solicitar una operación deseada.
- Los resultados de las operaciones realizadas a partir de los datos introducidos.

Al elaborar el Manual de Usuario hay que tener en cuenta a quién va dirigido, es decir, debe redactarse de forma clara y sencilla para que lo entienda cualquier tipo de usuario.

### 5.3.1.1 MANUAL DE USUARIO DEL SISTEMA PARA EL TRÁMITE DE PAGOS

El manual de usuario que se describirá a continuación representa un compendio de las funciones y operaciones más importantes del sistema, esto debido a que el manual que se elaboró tuvo 118 hojas, resultando imposible incorporarlo en este trabajo, por lo que sólo se mostrarán las opciones que permitan conocer la forma de solicitar y realizar el pago de gastos, anticipos y comprobaciones.

#### ACCESO AL SISTEMA

Para ingresar al sistema, sólo bastará dar doble clic en el icono **Sistema Trámite de Pagos**, ubicado en el escritorio de la computadora: (figura 5.1)

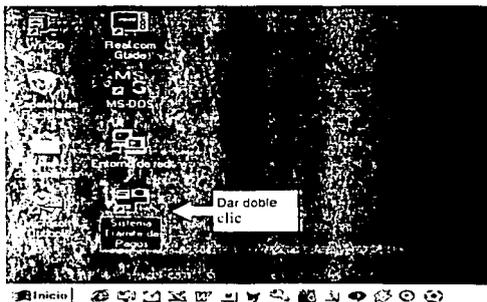


Figura 5.1 Ubicación del Sistema para el Trámite de Pagos

La pantalla inicial que se desplegará es la referente a la seguridad del sistema, a través de la cual se solicitará se ingrese la clave de usuario y contraseña: (figura 5.2)

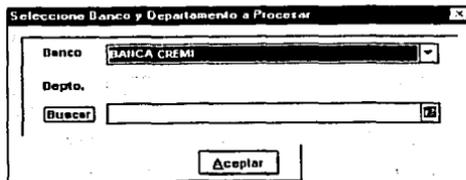


Figura 5.2 Ventana de acceso al sistema

TESIS CON  
FALLA DE ORIGEN

La clave de usuario delimitará el registro de la información a capturar, tanto por banco, como por área; por otra parte, al momento de ingresar, el sistema verificará si existen facturas rechazadas, adecuaciones a los catálogos o adecuaciones a los módulos de éste, para proceder a su actualización.

Si el usuario tiene acceso al sistema para ver más de un Banco o Área, se desplegará un cuadro de selección de Banco y Área: (figura 5.3)



Selección Banco y Departamento a Procesar

Banco: BANCA CREMI

Depto.:

Buscar

Aceptar

Figura 5.3 Ventana de selección de Banco y Departamento a procesar

## MENÚ PRINCIPAL

Cuando se autoriza el acceso al sistema, internamente se identifica el Banco y el Área a la que pertenece el usuario, delimitando la información a utilizar.

El sistema desplegará la pantalla de menú principal (figura 5.4), que cuenta con las siguientes funciones:

- Captura de Facturas: Registro de los documentos que amparan y originan gasto.
- Envío de Facturas: Envío de documentos autorizados para su trámite al Área de Central de Pagos.
- Consulta de Lotes: Consultas de la información enviada al Área de Central de Pagos, el conjunto de información se identifica a través de un lote.
- Utilerías del Sistema: Opciones que permiten dar mantenimiento a la base de datos, así como poder respaldar y recuperar información.
- Salir del Sistema: Opción que finaliza la sesión de trabajo.

TESIS CON  
FALLA DE ORIGEN

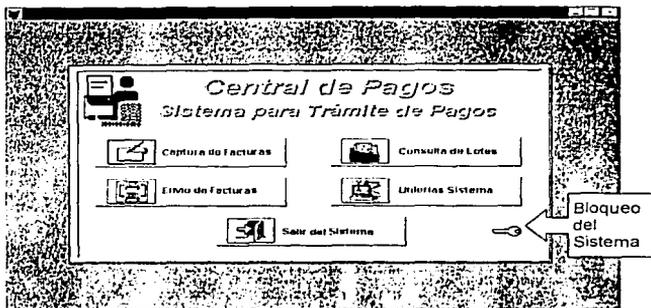


Figura 5.4 Menú Principal del Sistema para el Trámite de Pagos

### CAPTURA DE DOCUMENTOS

Para activar esta opción se deberá seleccionar el botón  Captura de Facturas ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.5)

**BANCA CREMI** FACTURA CAPTURADA

Beneficiario: 132901

Tipo de Gasto:  
 Adquis.  Anticipo  Comprobar  B. Avión  Otros Documento  
 Es Válido  Fecha Documento: 16 04 2007

Concepto: 323  ITEM: 168451202116  
 Serie:

Deppto. Gasto: 2000101100

**FORMA PAGO** Importe: 5,940.00 **DESCRIPCION DEL CONCEPTO**   
 Cheque  Descuento  Instalacion DEL SISTEMA  
 Abono en Cla.  Iva  
 Sin Pago  Ret. Iva  
 Pago Fecha Water:  Ret. ISR  
 Total:

Anticipo < Comprobar >  Abajo de la Comprobación

Figura 5.5 Captura de Facturas

Esta pantalla permite al usuario registrar las facturas, recibos de honorarios, anticipos y comprobaciones de proveedores o empleados que proporcionaron un servicio al banco y de los cuales se tiene que pagar el costo correspondiente.

Los estatus que puede tener un documento son:

- Capturado (C): Como su nombre lo indica, el documento sólo ha sido capturado.
- Error (E): El documento contiene errores en los importes registrados.
- Validado (V): Los importes registrados en la carátula de documentos coinciden con su detalle.
- En Firma (F): El documento esta en proceso de firma por el funcionario responsable.
- Autorizado (A): El documento ha sido autorizado para su pago.
- En Trámite (T): El documento ha sido enviado al Área de Central de Pagos para tramitar su pago.
- Respaldado (R): El documento ya fue respaldado en algún medio magnético.

#### DETALLE DE DOCUMENTOS A PAGAR

Para activar esta opción deberá estar seleccionado el icono de  **Detallar Documentos** en

la carátula de captura de documentos y se deberá dar clic en el botón  , con lo cual se desplegará la siguiente pantalla: (figura 5.6)

TESIS CON  
FALLA DE ORIGEN

Beneficiario		Serie	
3 ACCOR SERVICIOS EMPPEGARIALES, S.A. DE C.V.		100746	
Depto.Gasto	7500400000	Buscar	
Documento	230	Fecha Docto.	20 04 2001
Ciudad	0	Buscar	
Concepto	235	Buscar	
Importe		1.500.00	
Descuento		0.00 %	0.00
Iva		15.00 %	225.00
Retención IVA		0.00 %	0.00
Retención ISR		0.00 %	0.00
Total		1.725.00	

Figura 5.6 Captura del detalle de documentos

Mediante esta pantalla se procede a detallar los movimientos que amparan un documento, desglosando los importes por los conceptos fiscales más importantes, IVA, Retención de ISR e IVA.

## REGISTRO DE VIÁTICOS

Tipo de Gasto  
 Admvo.  Anticipo  
 Es Viático

Para activar esta opción deberá estar seleccionado el icono de dentro de la carátula de captura de documentos, con lo cual se desplegará la siguiente pantalla: (figura 5.7)

TESIS CON  
FALLA DE ORIGEN

Datos del Viaje			Serie 100284		
Beneficiario GONZALEZ GOMEZ BEATRIZ					
SALIDA			DESTINO		
7	<input type="text"/>	<input type="button" value="Buscar"/>	99	<input type="text"/>	<input type="button" value="Buscar"/>
<input type="checkbox"/> Fin del viaje			<input type="checkbox"/> Boleto de Avión		
Fecha	Duración en días	Hora	Fecha		
05-06-2003	3	15:00	07-06-2003		
Presupuesto			5,030.00		
<b>COMPROBACION DEL ANTICIPO</b>					
Días Reales			Al		
0			Del		
<input type="button" value="Previa"/> <input type="button" value="Buscar"/> <input type="button" value="Próximo"/> <input type="button" value="Antes"/> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> <input type="button" value="Cancelar"/> <input type="button" value="Salir"/>					

Figura 5.7 Captura de los datos del viaje

El presupuesto por cada ciudad, se calculará automáticamente de acuerdo a la ciudad y al número de días del viaje.

Si se trata de una comprobación, sólo se habilitarán los campos del apartado "Comprobación del Anticipo".

### SOLICITUD DE ANTICIPOS Y BOLETO DE AVIÓN

Para activar esta opción, se deberá seleccionar el icono  ubicado en la carátula de captura de documentos, con lo cual se desplegará la siguiente pantalla: (figura 5.8)

Reportes Anticipo de Viajes	
<input checked="" type="checkbox"/>	Transporte Aéreo
<input checked="" type="checkbox"/>	Anticipo de gastos
  	

TESIS CON  
 FALTA DE ORIGEN

Figura 5.8 Solicitud de anticipos y boleto de avión

Mediante esta pantalla se realizará la impresión de los reportes establecidos para la solicitud de los anticipos y boletos de avión relacionados con gastos de viáticos.



Al momento de seleccionar el botón de pantalla se podrá verificar el reporte previo de los datos capturados en lo que se refiere a la solicitud de boleto de avión y anticipos de viáticos.

El sistema mostrará la siguiente pantalla: (figura 5.9)



CEPM 19274  
-80 1992

**SOLICITUD DE TRANSPORTE AEREO**

FECHA  
16-SEP-2002

SOLICITANTE																	
NOMBRE SOCIETAT ECOMERCIAL SA																	
PUESTO SECRETARIA TITULAR		DEPARTAMENTO Y SERVICIO															
		111204															
*PERSONAL DE SERVICIO TITULAR DESARROLLO SISTEMAS CONTABLES		CLASIFICACION 744020100															
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2">VIAJES</th> </tr> <tr> <td>1</td> <td>DE: DISTRITO FEDERAL A: CHATEPPE, ILL</td> </tr> <tr> <td>2</td> <td>DE: DISTRITO FEDERAL A: CHATEPPE, ILL</td> </tr> </table>		VIAJES		1	DE: DISTRITO FEDERAL A: CHATEPPE, ILL	2	DE: DISTRITO FEDERAL A: CHATEPPE, ILL	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>FECHA:</td> <td>HORA:</td> </tr> <tr> <td>16-09-2002</td> <td>18:00</td> </tr> <tr> <td>FECHA:</td> <td>HORA:</td> </tr> <tr> <td>16-09-2002</td> <td>18:00</td> </tr> </table>	FECHA:	HORA:	16-09-2002	18:00	FECHA:	HORA:	16-09-2002	18:00	<p>QUEDE ENTERADO/A QUE EL PRESENCIA DE ESTE VIAJE SERA PRESENTAR EL COMPROBANTE QUE SERA PARA EL USO DEL TRANSPORTE AEREO. DE LO CONTRARIO SERA CARGADO A LA CUENTA DE CHEQUES EL REPORTE DEL DOCTORE A VUELTA</p>
VIAJES																	
1	DE: DISTRITO FEDERAL A: CHATEPPE, ILL																
2	DE: DISTRITO FEDERAL A: CHATEPPE, ILL																
FECHA:	HORA:																
16-09-2002	18:00																
FECHA:	HORA:																
16-09-2002	18:00																
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2">DEPARTAMENTO Y SERVICIO</th> </tr> <tr> <td colspan="2">NOMBRE SOCIETAT ECOMERCIAL SA</td> </tr> <tr> <td colspan="2">PUESTO SECRETARIA TITULAR</td> </tr> <tr> <td colspan="2">SERVICIO SECRETARIA TITULAR</td> </tr> </table>		DEPARTAMENTO Y SERVICIO		NOMBRE SOCIETAT ECOMERCIAL SA		PUESTO SECRETARIA TITULAR		SERVICIO SECRETARIA TITULAR									
DEPARTAMENTO Y SERVICIO																	
NOMBRE SOCIETAT ECOMERCIAL SA																	
PUESTO SECRETARIA TITULAR																	
SERVICIO SECRETARIA TITULAR																	

AUTORIZACION DIRECTOR CORPORATIVO

Figura 5.9 Vista previa de la solicitud de transporte aéreo

Si se selecciona el botón de impresora  se habilitará la pantalla de impresión estándar de Windows, permitiendo seleccionar el número de copias o el rango de hojas a imprimir: (figura 5.10)

TESIS CON  
FALLA DE ORIGEN

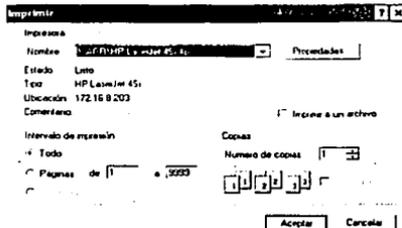


Figura 5.10 Ventana para enviar a impresora el reporte

## REGISTRO DE COMPROBACIONES

Para el registro de las comprobaciones se tendrá que seleccionar durante el alta de un documento los iconos de **Tipo de Gasto**  Admvo.,  Anticipo,  **Comprobar**,  B. Avión,  **Es Viático**,  **Detallar Documentos** ubicados en la carátula de captura de documentos.

Posterior a la captura del detalle de los documentos y a la captura de los datos complementarios dentro de la pantalla de registro de viáticos, se deberán generar los formatos de comprobación de gastos, los cuales se habilitan dando clic en el botón  que se encuentra ubicado en la carátula de captura de documentos, desplegando la siguiente pantalla: (figura 5.11)

TESIS CON  
 FALLA DE ORIGEN

Reportes Comprobación de

- Carátula Comprobación
- Detalle Comprobación



Figura 5.11 Generación de los reportes de comprobación de viáticos

Mediante esta pantalla se realizará la impresión de los reportes establecidos para la comprobación de los anticipos y boletos de avión relacionados con gastos de viáticos.

Al momento de seleccionar el botón de pantalla  se podrá verificar el reporte previo de los datos capturados.

El sistema mostrará la siguiente pantalla: (figura 5.12)



**BANCA CREMI**

**COMPROBACION DE GASTOS DE VIAJE Y VIATICOS**

<b>ANIO</b>	2001	<b>SERIE</b>	100203	<b>FECHA</b>	20-SEP-2001
<b>PERSONA</b>	PRIDGO FOMEPOR NAPTHA			<b>ROL</b>	R01000000000000000000
<b>PROYECTO</b>				<b>VALOR</b>	42969
<b>ANALISTA</b>				<b>VALOR</b>	616 2203 2249
<b>PROGRAMA</b>	PROGRAMA DE GASTOS DE VIAJE Y VIATICOS			<b>CLAVE</b>	7000 00 1000
<b>USUARIO</b>	ALTA A CONTABILIDAD POR PROGRAMA DE GASTOS DE VIAJE Y VIATICOS				

<b>EMPRESA</b>	<b>FECHA</b>	<b>PROGRAMA</b>	<b>USUARIO</b>
PRIDGO FOMEPOR NAPTHA		PROGRAMA DE GASTOS DE VIAJE Y VIATICOS	
PRIDGO FOMEPOR NAPTHA		PROGRAMA DE GASTOS DE VIAJE Y VIATICOS	

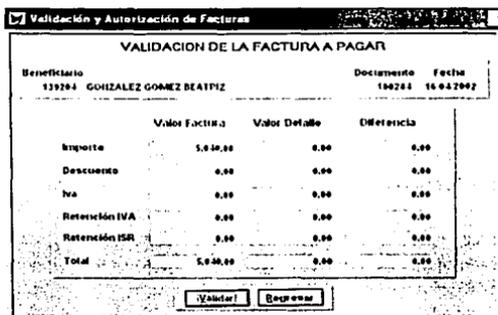
CLAVE	CONCEPTO	IMPORTE	IVA	TOTAL
184	VIATICOS SIN COMPROBANTE FUNCIONARIO	1,703.50	0.00	1,703.50
188	VIATICOS CON COMPROBANTE FUNCIONARIO	6,852.08	1,010.44	7,872.02
<b>TOTALES</b>		<b>8,505.08</b>	<b>1,010.44</b>	<b>9,505.52</b>

Figura 5.12 Vista previa de la comprobación de gastos de viáticos

De forma similar a los anticipos, si se selecciona el botón de impresora  se habilitará la pantalla de impresión estándar de Windows.

## VALIDACIÓN DE DOCUMENTOS

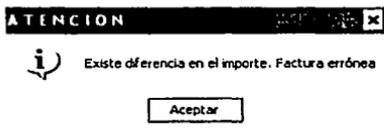
Para activar esta opción, se deberá seleccionar el icono  ubicado en la carátula de captura de documentos, con lo cual se desplegará la siguiente pantalla: (figura 5.13)



	Valor Factura	Valor Detalle	Diferencia
Importe	5.046.00	0.00	0.00
Descuento	0.00	0.00	0.00
Iva	0.00	0.00	0.00
Retención IVA	0.00	0.00	0.00
Retención ISR	0.00	0.00	0.00
Total	5.046.00	0.00	0.00

Figura 5.13 Validación de facturas

Esta pantalla permite validar los importes de la carátula contra los registrados en el detalle, si los importes no coinciden, el sistema genera una pantalla de alerta: (figura 5.14)



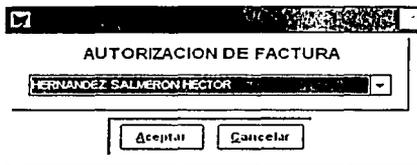
**ATENCIÓN**

Existe diferencia en el importe. Factura errónea

Aceptar

Figura 5.14 Mensaje que indica diferencia en los importes

Si los importes son correctos, se habilita la pantalla para la asignación del funcionario que deberá autorizar el documento: (figura 5.15)

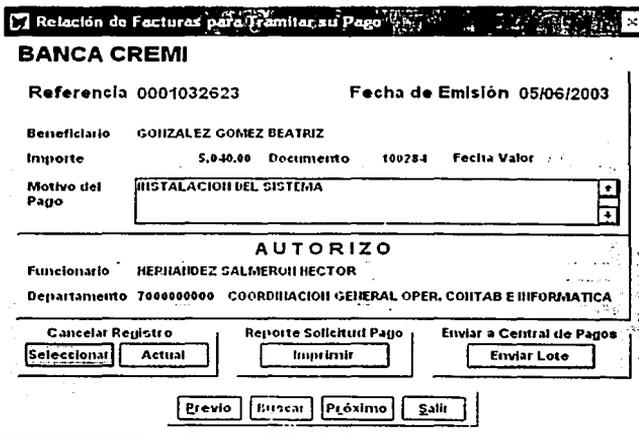


AUTORIZACION DE FACTURA  
HERNANDEZ SALMERON HECTOR  
Aceptar Cancelar

Figura 5.15 Autorización de facturas

## AUTORIZACION Y ENVIO DE DOCUMENTOS

Para activar esta opción se deberá seleccionar el botón  Envío de Facturas ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.16)



Relación de Facturas para Tramitar su Pago

**BANCA CREMI**

Referencia 0001032623 Fecha de Emisión 05/06/2003

Beneficiario GONZALEZ GOMEZ BEATRIZ

Importe 5,940.00 Documento 100284 Fecha Valor

Motivo del Pago INSTALACION DEL SISTEMA

**AUTORIZO**

Funcionario HERNANDEZ SALMERON HECTOR

Departamento 700000000 COORDINACION GENERAL OPER. CONTAB E INFORMATICA

Cancelar Registro Reporte Solicitud Pago Enviar a Central de Pagos

Seleccionar Actual Imprimir Enviar Lote

Previo Buscar Próximo Salir

Figura 5.16 Envío de Facturas



Reporte Solicitud Pago

Imprimir

Reporte solicitud de pago:

Esta función permite agrupar las facturas que están con estatus de "FIRMA" y asignar un número de lote que servirá como referencia para el trámite ante el Área de Central de Pagos, adicionalmente, permite la emisión de un reporte con las facturas contenidas en el lote.

Previo a la impresión del reporte de solicitud de pago, se deberán seleccionar los documentos que han sido autorizados por el funcionario y se le indicará al sistema que realice la asignación del número de lote, esto se hace a través de la siguiente pantalla: (figura 5.18)

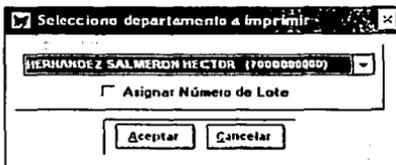


Figura 5.18 Selección del reporte para la solicitud de pago

Enviar a Central de Pagos

Enviar Lote

Enviar al Área de Central de Pagos:

Esta función permite realizar la transferencia electrónica de la información registrada al Área de Central de Pagos para su trámite.

Para tal efecto, se deberá seleccionar por departamento la información que será enviada al Área de Central de Pagos, indicándole al sistema la ruta en donde se desea almacenar el archivo generado, la pantalla que permite realizar esta situación es la siguiente: (figura 5.19)

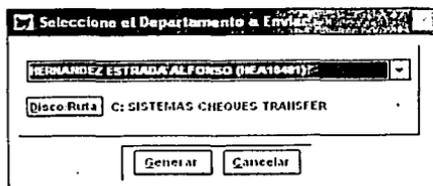


Figura 5.19 Envío de información al Área de Central de Pagos

## CONSULTA DE LOTES

Para activar esta opción se deberá seleccionar el botón  **Consulta de Lotes** ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.20)

Figura 5.20 Consulta de Lotes

A continuación se describen las principales funciones de esta pantalla:

- Ver Facturas:** Muestran la relación de facturas que integran el lote seleccionado.
- Reenviar Lote:** Permite reenviar el lote seleccionado nuevamente al Área de Central de Pagos, cuando por cualquier motivo no se haya recibido la información.
- Reporte:** Permite generar el reporte de Solicitud de Trámite de Pago, en donde se detallan las facturas que integran el lote.
- Recuperar Errores:** Permite recuperar los lotes que hayan sido rechazados por el Área de Central de Pagos.
- Salir:** Cierra la pantalla de Consulta de Lotes Enviados.

En la pantalla mostrada en la figura 5.20, el sistema desplegará una leyenda con el estatus del lote, de acuerdo a lo siguiente:

- **Lote con facturas rechazadas:** Indica que dentro de las facturas que integran el lote, el Área de Central de Pagos rechazó algunos documentos.
- **Lote enviado a central de pagos:** Indica que la información del lote ya fue enviada al Área de Central de Pagos.
- **Lote con recuperación de rechazos:** Indica que el lote contenía errores en algunos movimientos, pero éstos ya han sido reactivados para su captura y corrección.
- **Lote en trámite de pago:** Indica que el lote ya ha sido recibido por el Área de Central de Pagos y está en proceso de revisión para su pago.
- **Lote contabilizado:** Indica que el lote ha sido contabilizado y se han realizado los pagos correspondientes de las facturas que lo integran.

## UTILERIAS DEL SISTEMA



Utilerias Sistema

Para activar esta opción se deberá seleccionar el botón  ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.21)



Figura 5.21 Menú de Utilerias del Sistema

A continuación se describen las principales funciones de esta pantalla:

- Reportes:** Permite generar reportes que muestran la información almacenada en el sistema.
- Utilerias de Sistema:** Permite dar mantenimiento a la información registrada en el sistema: generar índices de tablas, respalda y restaurar información.
- Personalizar Sistema:** Permite establecer la ruta de acceso de los archivos que conforman el sistema, así como personalizar los reportes de acuerdo a las necesidades de cada área. Adicionalmente, presenta la información general sobre el Sistema y la opción de cambio de contraseña.
- Salir:** Regresa al menú principal.

Cuando se selecciona la opción de "Reportes", se mostrarán en forma de cascada los informes que se pueden generar a través del sistema, incluyendo los formatos para la solicitud y comprobación de viáticos, así como el formato de solicitud de reembolso por concepto de pasajes: (figura 5.22)



Figura 5.22 Reportes del sistema

Los archivos que almacenan la información capturada, requieren estar ordenados para



una consulta ágil y rápida, la opción Respaldar/Recuperar Información mostrará la ventana de indexar bases, mediante la cual se generarán los índices de las tablas: (figura 5.23)

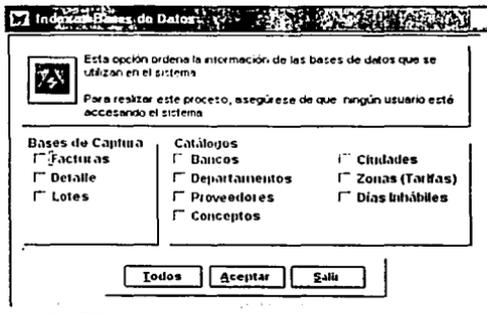


Figura 5.23 Generación de Índices

Como la información que se captura en el sistema se encuentra de manera local, es necesario realizar respaldos de información, para prevenir posibles pérdidas, para tal

Utilidades del Sistema Personalizar Sistema Salir  
Indexar Bases de Datos

efecto se deberá seleccionar la opción **Respaldo/Recuperar Información** con lo cual se habilitará la siguiente pantalla: (figura 5.24)

**Respaldo y Restauración de Información**

### BANCA CREMI

Opción que RESPALDA o RECUPERA la Información de Facturas y Lotes capturados en esta Computadora.

Para realizar este proceso, asegúrese de que ningún usuario esté accediendo el sistema.  
Una vez seleccionada una opción no deberá detener el proceso.

Indicar la ubicación física de los archivos a Respaldo o Recuperar

**Disco:Rt1a**

Respaldo      Información de: //      A: //

Recuperar      Archivo a Recuperar [ ]

**Aceptar    Cancelar    Salir**

Figura 5.24 Respaldo y Restauración de Información

Como el sistema se instala de manera local, cada equipo puede conectarse a la base de datos principal a través de unidades lógicas de red, pero no necesariamente puede ser la

Personalizar Sistema Salir  
Parámetros del Sistema  
Cambiar Contraseña  
Acercar de...

misma de un equipo a otro, por lo que al seleccionar la opción **Acercar de...** se habilitará la pantalla mostrada en la figura 5.25, en donde se definirán las rutas de acceso a la base:

TESIS CON  
FALLA DE ORIGEN

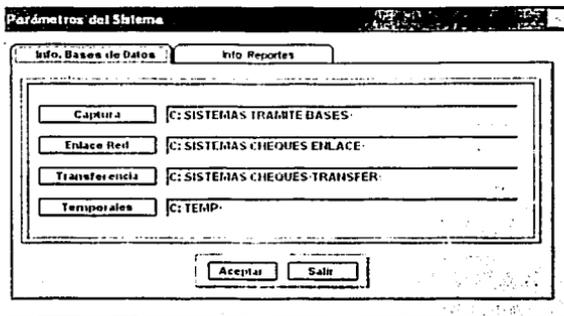


Figura 5.25 Parámetros del sistema

Finalmente, si ocurre algún problema con el sistema, al seleccionar la opción

Personalizar Sistema Salir

Parámetros del Sistema

Cambiar Contraseña

Acerca de... se indica a dónde se pueden dirigir para el soporte a usuarios, la figura 5.26 ilustra esta situación:



Figura 5.26 Pantalla Acerca de...

### 5.3.1.2 MANUAL DE USUARIO DEL SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

El manual de usuario que se describirá a continuación representa un compendio de las funciones y operaciones más importantes del sistema, esto debido a que el manual que se elaboró tuvo 140 hojas, resultando imposible incorporarlo en este trabajo, por lo que sólo se mostrarán las opciones que permitan conocer la operación de la Ventanilla, la elaboración de cheques y la contabilización de las pólizas generadas.

#### ACCESO AL SISTEMA

Para ingresar al sistema, sólo bastará dar doble clic en el icono **Sistema Administrativo-Contable de Pagos**, ubicado en el escritorio de la computadora: (figura 5.27)

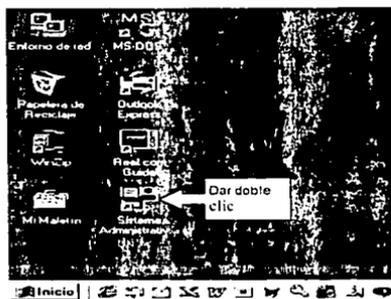


Figura 5.27 Ubicación del Sistema Administrativo-Contable de Pagos

La pantalla inicial que se desplegará es la referente a la seguridad del sistema, a través de la cual se solicitará se ingrese la clave de usuario y contraseña: (figura 5.28)

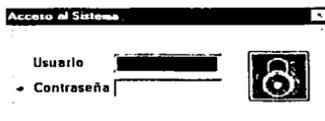


Figura 5.28 Ventana de acceso al sistema

La clave de usuario delimitará la información a procesar por banco, por otra parte, al momento de ingresar, el sistema verificará si existen adecuaciones a los módulos que lo conforman para proceder a su actualización.

### MENÚ PRINCIPAL

El sistema desplegará la pantalla de menú principal (figura 5.29), que cuenta con las siguientes funciones:

- |                    |  |
|--------------------|--|
| Contra-Recibos:    | Registro y validación de los contra-recibos que son entregados a la Ventanilla para su revisión y posterior autorización de pago.  |
| Emisión de Pagos:  | Selección de contra-recibos autorizados para la generación de pago, ya sea través de cheque o abono a cuenta de cheques, así como la solicitud de fondos a Tesorería.            |
| Contabilización:   | Selección de todos los pagos efectuados durante el día para proceder a la asignación de las cuentas contables correspondientes, así como la generación de sus pólizas contables. |
| Catálogos:         | Mantenimiento a todos los catálogos del sistema.   |
| Utilerías:         | Opciones que habilitan el mantenimiento a la bitácora de acceso, a los usuarios del sistema y a la transferencia de información.   |
| Salir del Sistema: | Opción que finaliza la sesión de trabajo.  |

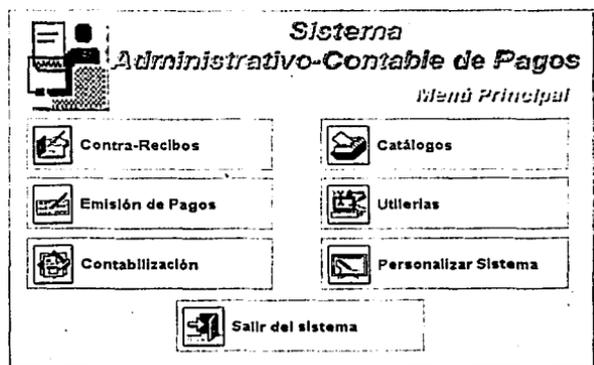


Figura 5.29 Menú Principal del Sistema Administrativo-Contable de Pagos

## CONTRA-RECIBOS

Para activar esta opción se deberá dar clic en el botón  **Contra-Recibos** ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.30)

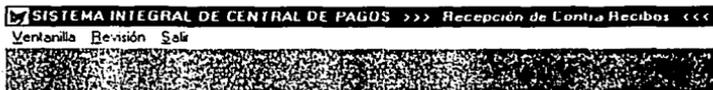


Figura 5.30 Menú de Contra-Recibos

Las opciones de esta sección se muestran en forma de menú de cascada, las cuales permiten el registro y recepción de documentos provenientes de los Departamentos para su autorización o rechazo.

TESIS CON  
 FALLA DE ORIGEN

Al momento de dar clic en la opción de Ventanilla se muestran el siguiente submenú

**Ventanilla** Revisión Salir  
**Captura de Contra-Recibos**  
**Reporte Contra-Recibos en Captura** a través del cual se podrá realizar la recepción de documentos e impresión de contra-recibos registrados en el sistema.

Si se selecciona la opción de **Captura de Contra-Recibos** se mostrará la siguiente pantalla: (figura 5.31)

Captura de Contra-Recibos			
Banco		CONTRA-RECIBO: VALIDADO	
BANCA CREMI	REFERENCIA	CONTRA-RECIBO: 6375	
NOMBRE O RAZÓN SOCIAL	DEPTO/RESPONSABLE	CENTRO COSTOS	
REYRUSO PARRIS, JOS. VICENTE	500200000	116	
RECIBIMOS PARA SU REVISIÓN/PAGO <input type="radio"/> Admón. <input type="radio"/> Anticipo <input type="radio"/> Comprob. <input type="radio"/> B. Avidón			
CONCEPTO	FORMA PAGO	IMPORTE	COMENTARIO
REPORTE	<input type="radio"/> Cheque <input type="radio"/> Abono a Cte. <input type="radio"/> Sin Pago	16.835,00	FAC. 3268, 536, 3269, 537, COPIAS CLERICALDAS, LXI 1/21
	Artículo	0,00	
	Saldo Comprob.	0,00	
FECHA RECEPCIÓN	FECHA VALOR	Prev.	Devon.
10/10/2010	10/10/2010	<input type="checkbox"/>	<input checked="" type="checkbox"/>
FECHA DE PAGO	RESPONSABLE DE REVISIÓN	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	
		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	
<input type="button" value="Previa"/> <input type="button" value="Buscar"/> <input type="button" value="Próximo"/> <input type="button" value="Antes"/> <input type="button" value="Editar"/> <input type="button" value="Revisar"/> <input type="button" value="Cancelar"/> <input type="button" value="Imprimir"/> <input type="button" value="Salir"/>			

Figura 5.31 Captura de Contra-Recibos

Esta pantalla permite registrar los contra-recibos enviados por los diferentes Departamentos que originaron algún gasto.

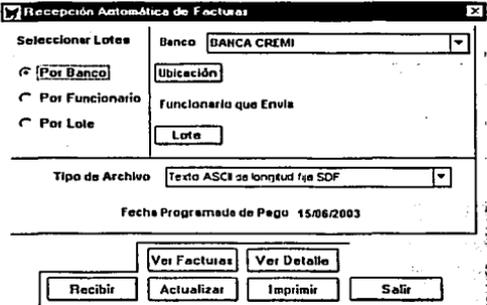
Una función importante de esta pantalla de captura, es permitirle al personal de Ventanilla verificar que la asignación de los Responsables de Revisión haya sido correcta, de lo contrario se deberá asignar otro.

La asignación de Responsables de Revisión se efectúa automáticamente, esto se logra asociando dentro del catálogo de Conceptos de pagos a los Responsables.

Los posibles estatus que puede tener un contra-recibo son:

- Capturado (C): Como su nombre lo indica, el contra-recibo sólo ha sido capturado.  
Error (E): El contra-recibo contiene errores y será reenviado al departamento que lo transmitió.  
Validado (V): El contra-recibo ya fue revisado y autorizado para su pago.  
Cheque (Q): Indica que ya se emitió el cheque del contra-recibo.  
Liberado (L): Indica que los cheques y abonos de los contra-recibos están listos para su entrega al beneficiario del pago.  
Póliza (P): Indica que los pagos correspondientes a los contra-recibos ya han sido contabilizados y enviados al Sistema Central de Contabilidad.

Se puede realizar el registro de los contra-recibos en forma manual mediante el uso del botón  o utilizando la recepción automática de información, que se habilita al momento de dar clic en el botón  , la pantalla que se muestra al seleccionar esta opción es: (figura 5.32)



Recepción Automática de Facturas

Seleccionar Lotes

Por Banco

Por Funcionario

Por Lote

Banco: BANCA CREMI

Ubicación

Funcionario que Envía

Lote

Tipo de Archivo: Texto ASCII de longitud fija SDF

Fecha Programada de Pago: 15/08/2003

Ver Facturas Ver Detalle

Recibir Actualizar Imprimir Salir

Figura 5.32 Recepción automática de facturas

Mediante esta pantalla se selecciona por banco, funcionario o lote, la información que se desea procesar, antes de realizar la actualización de los contra-recibos se puede ver el detalle de los documentos que se están recibiendo.

Una vez que se tienen capturados los contra-recibos, se deberá realizar la impresión correspondiente para entregársela a los proveedores, este reporte amparará los documentos entregados y contra su entrega se darán los cheques.

Para realizar la impresión del contra-recibo se deberá dar clic en el botón **Imprimir** esta acción enviará directamente a la impresora el reporte, una muestra del contra-recibo generado es: (figura 5.33)

BANCA CREMI		CONTRA - RECIBO	
NOMBRE O RAZON SOCIAL		DEPARTAMENTO/RE SPONSABLE	C.C. QUE AUTORIZA
GONZALEZ GOMEZ DE ATRIZ		TITULAR DE DESARROLLO SISTEMAS CONTABLES ACEVEDO ROMERO IRZA YARANI	193
RECEBIMOS PARA SU REVISION Y PAGO:			
<input type="checkbox"/> Adwco. <input checked="" type="checkbox"/> Anticipo <input type="checkbox"/> Comprobación <input type="checkbox"/> B. Antón			
CONCEPTO	FORMA DE PAGO	IMPORTE	COMENTARIO
MAIICOS	Cheque	6,000.00	INSTALACION DEL SISTEMA
TOTAL		6,000.00	
FECHA DE RECIBIDO:	3-Jan-2003 15:19:29	PEC	NOMBRE Y FIRMA DE RECIBIDO
FECHA DE PAGO:	Lun 31-Jan		
NOTAS: 1. RECIBO Y PAGO DE COMPROBANTES SOLO SE HAN DE ACUERDO A LO PROGRAMADO. Recopon de Documentos: Lunes a Jueves de 8:00 a 12:30 Pm. Entrega de Cheques: Lunes a Jueves de 8:00 a 12:30 Pm.			
A SOLICITANTE			

Figura 5.33 Contra-Recibo entregado por el Área de Ventanilla

Al finalizar la recepción de documentos, el personal de Ventanilla realizará la agrupación de los contra-recibos de acuerdo a los Responsables de Revisión, para que sean éstos los que validen los documentos recibidos, para tal efecto, se deberá seleccionar la opción **Reporte Contra-Recibos en Captura** ubicada dentro del submenú de Ventanilla, habilitando la siguiente pantalla: (figura 5.34)

**TESIS CON  
FALLA DE ORIGEN**

Figura 5.34 Reporte de Contra-Recibos en captura

Mediante esta pantalla se realizará la impresión de los contra-recibos asignados por Responsable de Revisión.

Para realizar la revisión de los documentos recibidos e impresión de contra-recibos registrados en el sistema, se deberá seleccionar la opción de "Revisión", con lo cual se

desplegará el siguiente submenú

Al momento de dar clic en la opción **Revisión de Facturas** se mostrará una ventana que solicitará se ingrese el banco a procesar: (figura 5.35)

Figura 5.35 Ventana para seleccionar el Banco a procesar

Internamente el sistema seleccionará los documentos asociados a los contra-recibos que se encuentran con estatus de "Captura", permitiéndole a los Responsables de Revisión validar contra la documentación original los importes, fechas y conceptos de pago, para apoyar esta actividad se utilizará la siguiente pantalla: (figura 5.36)

**Revisión de Facturas Emisadas** 16:37:21

**BANCA CREMI** CONTRA RECIBO REFLENCIA  
10358 8672911289

Beneficiario: 984  PRECIADO HONDON JUAN PEDROE

Tipo de Gasto:  Cheque  Abono en Cta.  Sin Pago

Admno.  Anticipo  Comprobar  B. Avión  Mnt. Documento: 33

In. Viajes  Detalles Documentos  Fecha Documento: 02/03/2001

Concepto: 246  GASTOS Y VIJES DE INFORMACIÓN  Solos: 0

Depto. Gasto: 8500000100  GONZALEZ CHELLAP ALI JAHIRA

FORMA PAGO	Importe	1,586.80	DESCRIPCION DEL CONCEPTO
<input type="checkbox"/> Cheque	Descuento	8.80	PAGO DE LA DIFERENCIA EN EL SERVICIO DE LIMPIEZA Y
<input type="checkbox"/> Abono en Cta.	Ive	225.80	MANTEIMIENTO EN LOS
<input type="checkbox"/> Sin Pago	Retención Iva	8.80	PARABLES ADJUDICADOS DE LA PLAZA DE HERMOSILLO SONORA
<input type="checkbox"/> Pago Fecha Valor	Retención ISR	8.80	CORRESPONDIENTE A LOS MESES DE ENERO Y FEBRERO DEL
///	Total	1,775.00	

Anticipo x Comprobar: 0.00

Figura 5.36 Revisión de Facturas

Para verificar que la afectación contable se realizará correctamente, se podrá emitir un reporte previo de afectación contable tomando como base el concepto de pago, para realizar esta actividad se deberá dar clic en la opción **Reporte Contable de Contra-Recibos**, la pantalla que se muestra es la siguiente: (figura 5.37)

**Reporte Contable de Contra-Recibos Recibido**

Banco:

Persona Responsable de Revisión:

Fecha de Recepción de Contra-Recibos De:  A:

Título del Reporte: **REPORTE CONTABLE DE CONTRA-RECIBOS EN CAPTURA**

Figura 5.37 Reporte previo contable de los contra-recibos

**TESIS CON  
FALLA DE ORIGEN**

Mediante esta pantalla se podrá seleccionar por fecha y Responsable de Revisión los contra-recibos que se desean revisar.

## EMISIÓN DE PAGOS

Para activar esta opción se deberá dar clic en el botón  Emisión de Pagos ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.38)



Figura 5.38 Menú Emisión de Pagos

Las opciones de esta sección se muestran en forma de menú de cascada, las cuales permitirán seleccionar los contra-recibos autorizados para realizar su pago, la emisión de los cheques o abonos, la generación previa de los reportes contables, la liberación de los pagos o en su defecto su cancelación, así como la entrega de los cheques emitidos.

A continuación se muestran todas las posibles opciones que se desplegarán cuando se seleccione el submenú Emisión de Pagos : (figura 5.39)

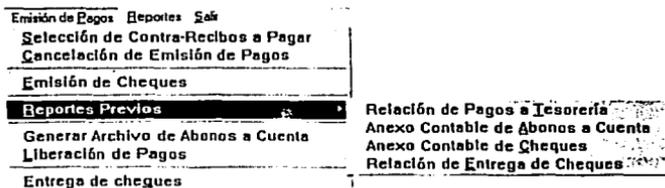


Figura 5.39 Opciones del submenú Emisión de Pagos

Al seleccionar la opción de **Selección de Contra-Recibos a Pagar** se desplegará la siguiente pantalla: (figura 5.40)

**Selección de Contra-Recibos a Pagar**

**Por Banco**      Banco  
BANCA CREM

**Por Región**      Región

**Por Plaza**      Plaza

Responsable de Revisión  
---TCOOS---

Fecha Programada de pago    De: / /      A: / /

Forma de Pago     Cheques     Abonos     Sin Pago

Fecha de Liberación    07/08/2003

Figura 5.40 Selección de contra-recibos a pagar

A través de esta pantalla se podrán seleccionar los contra-recibos a los que se les generará su pago, al concluir con la selección el sistema desplegará una ventana que indicará la fecha y número de emisión asignados: (figura 5.41)

**ATENCIÓN**

Selección de Pagos a Efectuar el día 07/05/2003, con Número de Emisión 001. Concluido.

Figura 5.41 Asignación de fecha y número de emisión de pago

La fecha indica cuando serán liberados los cheques o abonos y se asigna un número de emisión, debido a que puede haber más de una emisión de cheques por día.

Como medida de seguridad, se incluye un proceso de cancelación de emisiones de pagos, la cual se habilitará al seleccionar la opción de **Cancelación de Emisión de Pagos**, desplegando la siguiente pantalla: (figura 5.42)

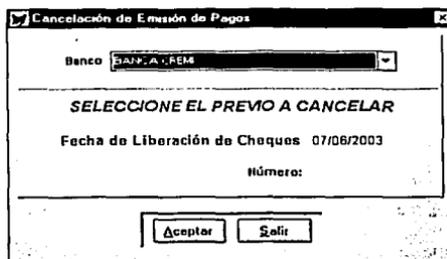


Figura 5.42 Cancelación de Emisión de Pago

Para realizar la cancelación de una emisión se deberá indicar la fecha y número de emisión.

Al seleccionar la opción de **Emisión de Cheques** se desplegará la siguiente pantalla: (figura 5.43)

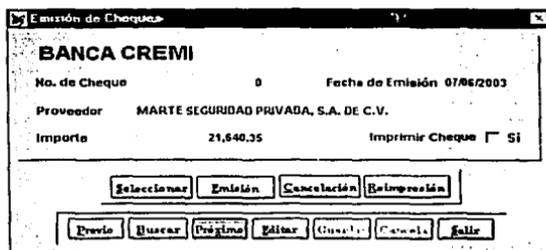


Figura 5.43 Emisión de cheques

La cual permitirá realizar la impresión física de los cheques, ya sea seleccionándolos manualmente mediante el botón **Imprimir Cheque**  **Si** o indicándole un grupo de cheques mediante el botón **Seleccionar**, si se opta por esta última opción se desplegará la siguiente ventana: (figura 5.44)

Selección de Información para Realizar la Emisión de Cheques

Todo
  Seleccionar por Fecha
  Eliminar Selección

Fecha Emisión: / /  
 Número:

Figura 5.44 Selección de información para Emisión de Cheques

Mediante esta ventana se podrá seleccionar la información a imprimir, ya sea por fecha de emisión o toda, así mismo, se incluye una opción para eliminar la selección previa de información.

Si se desea ver que información se ha seleccionado para generar sus cheques, sólo bastará dar clic en el botón  , para que se muestren éstos: (figura 5.45)

Busqueda de Cheques

Fecha Emisión	Numero	Proveedor	Importe	Cheque	Imprimir
07/06/2003	001	MARTE SEGURIDAD PRIVADA, S.A. DE C	21640.35		
07/06/2003	001	CORP. DE SERVS. JUR. CRED. Y COB. S.A	4819.00		
07/06/2003	001	REYNOSO PABLOS JOSE VICTOR	16485.00		
07/06/2003	001	CORP. ASESORES JURD. Y NOTARIALES	1437500.00		
07/06/2003	001	SALINAS DURAN RAFAEL	410.00		
07/06/2003	001	BETANZOS CARRASCO PEDRO	500.00		

Figura 5.45 Relación de movimientos para generar sus cheques

Para realizar la impresión de los cheques bastará dar clic en el botón  para que se habilite la siguiente pantalla: (figura 5.46)

TESIS CON  
 FALLA DE ORIGEN

**Impresión de Cheques**

Ciudad

Departamento: CENTRAL DE PAGOS

No. Cheque Inicial: 1

Aceptar Cancelar

Figura 5.46 Impresión de Cheques

Mediante esta pantalla se indican las leyendas adicionales que se incluirán en el cheque, así como el número inicial, para su posteriormente envío a la impresora.

Al seleccionar la opción de **Generar Archivo de Abonos a Cuenta** se desplegará la siguiente pantalla: (figura 5.47)

**Generación del Archivo de Abonos a Cuenta**

Banco: BANCA CFEM

Selección Fecha de Liberación

Mes: Enero Año: 0 Día: [ ]

Nombre del Archivo a Enviar

Disco/Ruta

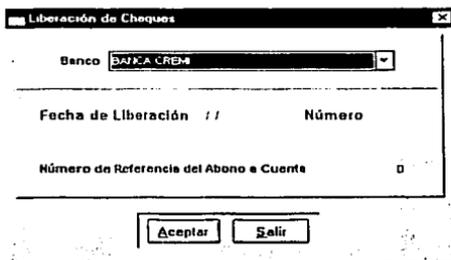
Tipo de Archivo: Texto ASCII de longitud fija SDF

Generar Verificar Enviar Salir

Figura 5.47 Generación del archivo de Abonos a Cuentas de Cheques

Mediante esta pantalla se generará el archivo de transferencia que se utilizará para la carga de datos en la Banca Electrónica, se tendrá que indicar la fecha y emisión que se desea seleccionar, así como la ruta en donde se ubicará el archivo resultante.

Al seleccionar la opción de **Liberación de Pagos** se desplegará la siguiente pantalla: (figura 5.48)



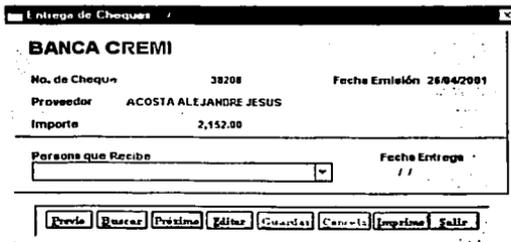
Banco	
BANCA CREMI	
Fecha de Liberación	///
Número	
Número de Referencia del Abono a Cuenta	0

Aceptar Salir

Figura 5.48 Liberación de Cheques

Mediante esta pantalla se realizará la liberación de los cheques emitidos, que consiste en preparar los contra-recibos relacionados con los cheques, para su posterior contabilización.

Al seleccionar la opción de **Entrega de cheques** se desplegará la siguiente pantalla: (figura 5.49)



<b>BANCA CREMI</b>			
No. de Cheque	38208	Fecha Emisión	26/04/2001
Proveedor	ACOSTA ALEJANDRE JESUS		
Importe	2,152.00		
Personas que Recibe		Fecha Entrega	///

Previa Buscar Próximo Editar Guardar Cancelar Imprimir Salir

Figura 5.49 Entrega de Cheques

A través de esta pantalla se realizará la entrega de los cheques a los beneficiarios del pago, permitiendo registrar la persona que recibe, así como la fecha de entrega.

Para realizar la selección y búsqueda de cheques que se deben entregar, bastará dar clic en el botón **Buscar** con lo que se desplegará la siguiente ventana: (figura 5.50)

Búsqueda de Entrega de Cheques			
Proveedor	Fec.Emisión	Cheque	Importe
ACCOM, S.A. DE C.V.	25/04/2001	30186	920.00
ACOSTA ALEJANDRE JESUS	26/04/2001	30208	2,152.00
ACOSTA ALEJANDRE JESUS	18/04/2001	38110	735.20
AEROPARTES Y SERVICIOS DEL CENTRO, S.A. DE C.V.	25/04/2001	38190	5,566.00
AGUILAR COLIO GUILLERMO	24/04/2001	30168	1,052.00
ALVAREZ VARGAS AGUSTIN	26/04/2001	30224	6,060.50
ARCE ULLOA HECTOR	03/04/2001	37865	12.89
ASOC. DE COLONOS DE RINCON DE LOS PASTORES	25/04/2001	30188	5,493.95
B Y B DISTRIBUIDORES, S.A. DE C.V.	26/04/2001	30206	2,273.00
BABZ GUERRERO JESUS	22/03/2001	37655	4,750.00

Figura 5.50 Relación de Cheques para su entrega

Referente a la opción de reportes previos del submenú de "Emisión de Pagos"

<b>Reportes Previos</b>	Relación de Pagos a Tesorería Anexo Contable de Abonos a Cuenta Anexo Contable de Cheques Relación de Entrega de Cheques
-------------------------	---

cabe mencionar que es similar al mostrado en el submenú de "Reportes", la diferencia radica en que en este punto aún no se ha realizado la liberación de cheques.

A continuación se muestran todas las posibles opciones que se desplegarán cuando se seleccione el submenú **Reportes** ubicado dentro del menú de "Emisión de Pagos": (figura 5.51)

Reportes Salir
Relación de Pagos a Tesorería
Anexo Contable de Abonos a Cuenta
Anexo Contable de Cheques
Relación de Entrega de Cheques
Relación de Contra-Recibos Recibidos
Relación de Cheques Emitidos
Cheques Cancelados

Figura 5.51 Reportes Contables

Mediante este submenú se podrán emitir todos los reportes de soporte contable de una emisión de pago, la operación y funcionamiento de éstos es similar, no así su contenido,

pero para efectos de este trabajo sólo se mostrará la operación del reporte "Relación de Pagos a Tesorería".

Al momento de dar clic en la opción **Relación de Pagos a Tesorería**, se desplegará la siguiente pantalla: (figura 5.52)

The screenshot shows a window titled "Relación de Pagos a Tesorería". It contains a dropdown menu for "Banco" with "BANCA FES" selected. Below this is a section titled "Seleccionar Fecha de Liberación" with three input fields: "Mes" (set to "Enero"), "Año", and "Día". Underneath is the text "Título del Reporte" followed by "RELACION DE PAGOS A TESORERIA". At the bottom, there are four buttons: "Buscar", "Reporte", "Cancelar", and "Salir".

Figura 5.52 Reporte Relación de Pagos a Tesorería

En esta pantalla se indica el banco y la fecha de liberación de los pagos para obtener su reporte, para iniciar con la búsqueda de información se deberá dar clic en el botón



Quando se obtiene la información deseada se deberá dar clic en el botón **Reporte**, con lo cual se habilitará la siguiente ventana: (figura 5.53)

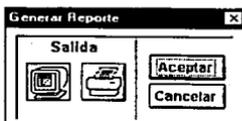


Figura 5.53 Ventana para generar los reportes

Esta pantalla será el estándar que utilizará en todo el sistema cuando se genere un reporte.



Al seleccionar el botón se podrá visualizar el reporte en pantalla, por lo que el sistema mostrará la siguiente ventana: (figura 5.54)

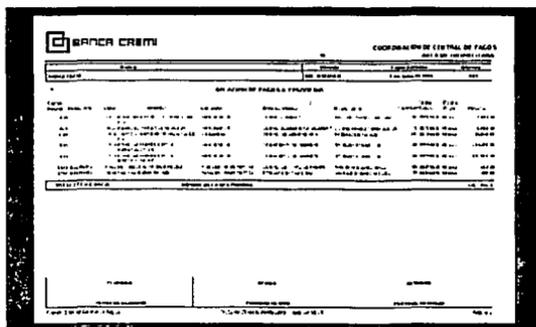


Figura 5.54 Vista previa del reporte



Si se selecciona el botón se habilitará la pantalla de impresión estándar de Windows, permitiendo seleccionar el número de copias o el rango de hojas a imprimir: (figura 5.55)

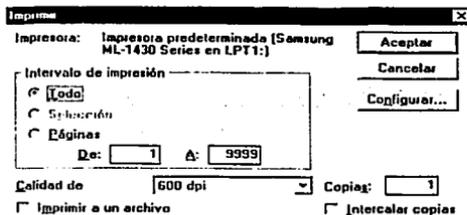


Figura 5.55 Ventana de diálogo para enviar a impresora un reporte

## CONTABILIZACIÓN

Para activar esta opción se deberá dar clic en el botón  Contabilización ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.56)



Figura 5.56 Menú de Contabilización

Las opciones de esta sección se muestran en forma de menú de cascada, las cuales permitirán seleccionar las emisiones de pagos para realizar su afectación contable y sus respectivas pólizas, así como la generación de los reportes contables que servirán como soporte de los movimientos efectuados.

A continuación se muestran todas las posibles opciones que se desplegarán cuando se seleccione el submenú Contabilización : (figura 5.57)



Figura 5.57 Opciones submenú de Contabilización

Al seleccionar la opción Contabilización se desplegará la siguiente pantalla: (figura 5.58)

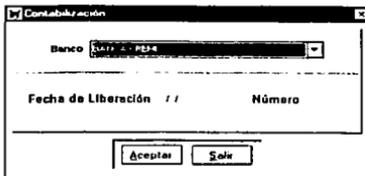


Figura 5.58 Ventana para seleccionar los movimientos a contabilizar

A través de esta pantalla se realizará el primer paso de la contabilización, que consiste en seleccionar los cheques o abonos emitidos, indicando para tal efecto la fecha y número de emisión de pago, antes de proceder con la afectación contable.

Antes de iniciar el proceso, el sistema mandará una alerta solicitándole al usuario ratifique o cancele la acción de contabilización: (figura 5.59)

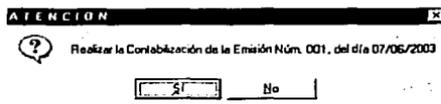


Figura 5.59 Notificación de inicio de Contabilización

El sistema internamente realizará una serie de procesos que conllevarán a la asignación de las cuentas contables de acuerdo a los conceptos de pago capturados, al momento de finalizar dichos procesos, se desplegará la siguiente pantalla indicándole al usuario que el proceso ha terminado satisfactoriamente: (figura 5.60)

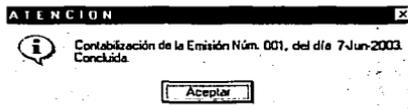


Figura 5.60 Mensaje de finalización de Contabilización

Si la asignación de cuentas contables no se realizó correctamente, entonces se desplegará el siguiente mensaje de error: (figura 5.61)



Figura 5.61 Mensaje de errores en la Contabilización

TESIS CON  
FALLA DE ORIGEN

Para revisar los errores detectados, se deberá seleccionar la opción **Reporte de Errores Contables** con lo cual se desplegará la siguiente pantalla: (figura 5.62)

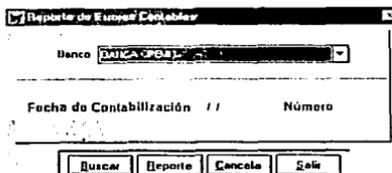


Figura 5.62 Emisión del reporte de errores contables

Mediante esta pantalla se podrá emitir un reporte con las causas de error detectadas durante el proceso de contabilización.

Al seleccionar la opción **Cancelar Contabilización** se desplegará la siguiente pantalla: (figura 5.63)

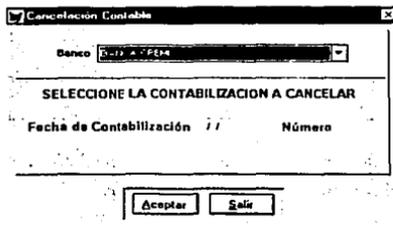


Figura 5.63 Cancelación Contable

La utilización de esta pantalla aplicará cuando durante el proceso de contabilización se hayan detectado errores, o simplemente si no se desea continuar con el proceso de generación de pólizas, se deberá especificar la fecha y número de contabilización que se quiere cancelar.

Antes de proceder con la cancelación contable, el sistema mandará una alerta solicitándole al usuario ratifique o cancele el proceso: (figura 5.64)

**ATENCIÓN**



REALIZAR LA CANCELACIÓN CONTABLE DE LA EMISIÓN NÚM. 001, DEL DÍA 07/06/2003

Sí  No

Figura 5.64 Notificación para la cancelación contable

Si la contabilización se ha realizado correctamente, se deberá realizar la impresión de las pólizas contables, esto se logra seleccionando la opción **Pólizas Contables** con lo cual se desplegará la siguiente pantalla: (figura 5.65)

**Diálogo Pólizas Contables**

Banco:

Fecha de Contabilización // Núm.:

Figura 5.65 Generación de Pólizas Contables

Como previamente ya se realizó la contabilización, en esta pantalla automáticamente se desplegarán todas las contabilizaciones efectuadas en la fecha solicitada, el formato de las pólizas contables generado se ilustra en la siguiente figura: (figura 5.66)

										<b>POLIZA CONTABLE</b>	
										Moneda: <input type="text" value="998 NACIONAL"/>	
<b>Código Cúpo</b>	<b>Código Cta.</b>	<b>S Cta.</b>	<b>SS Cta.</b>	<b>SSS Cta.</b>	<b>Sector</b>	<b>Ram</b>	<b>Serie</b>	<b>Cargos</b>	<b>Abonos</b>		
175 152	5111	12	00	00	00	168110304001	1	410.00		410.00	
<b>CONCEPTO</b>										<b>TOTALES:</b>	
BENEFICIARIO: SALINAS DURAN RAFAEL										410.00	
COMPROBACION DE GASTOS DE TRANSPORTE EN TAXI A DIVERSOS LUGARES SEGUN RELACION ANEXA ILLABORAR CHEQUE A NOMBRE DE RAFAEL SALINAS DURAN (8113)										10757 - 8878012482 - 88809098094	
<b>Departamento</b>	<b>Pieza</b>	<b>Fecha</b>	<b>Elaboró</b>	<b>Revisión Contable</b>	<b>Autorización Contable</b>						
CENTRAL DE PAGOS	800 METROPOLITANA	7-Jun-2003 Jun. 2001	PENDONA ELAORO	PERDO BETANZOS CARABACO	PERDO BETANZOS CARABACO						

Figura 5.66 Formato de Póliza Contable

TESIS CON  
FALLA DE ORIGEN

El último proceso de la contabilización se lleva a cabo al momento de generar la exportación contable de movimientos al Sistema Central de Contabilidad, para realizar esta acción se deberá seleccionar la opción **Exportación ICBS** con lo cual se desplegará la siguiente pantalla: (figura 5.67)

**Exportación Contable**

Banco: **BANCA CREDITA**

Ubicación del Servidor Mesa de Control

Ruta Servidor: C:\CHEQUES\TRANSFER

Fecha de Liberación: 11/11/03      Identificador de Archivo: CR

Figura 5.67 Exportación Contable

Mediante esta pantalla se indicará la ubicación del archivo plano (TXT) generado, así como la información que deberá contener.

Antes de iniciar el proceso de exportación, el sistema mandará una alerta solicitándole al usuario ratifique o cancele el proceso: (figura 5.68)

**ATENCIÓN**

Realizar la Exportación Contable Correspondiente al día 07/06/2003

Figura 5.68 Notificación para la Exportación Contable

Al concluir la exportación contable, el sistema mostrará una ventana indicando el nombre y ruta donde fue depositada la información: (figura 5.69)

TESIS CON  
FALLA DE ORIGEN

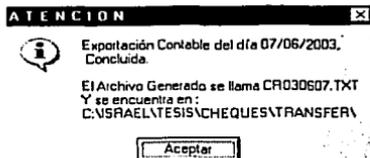


Figura 5.69 Notificación de Archivo generado

A continuación se muestran todas las posibles opciones que se desplegarán cuando se seleccione el submenú Reportes Contables : (figura 5.70)



Figura 5.70 Reportes Contables

Mediante este submenú se podrán imprimir nuevamente las pólizas generadas, indicando la fecha contable, la pantalla que se muestra es la siguiente: (figura 5.71)

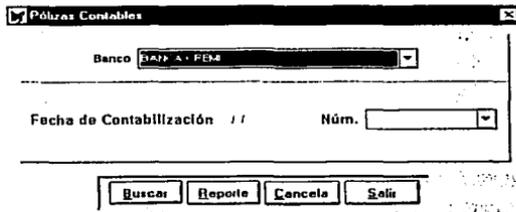


Figura 5.71 Generación de Pólizas Contables

## CATÁLOGOS

Para activar esta opción se deberá dar clic en el botón  **Catálogos** ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.72)



Figura 5.72 Menú de Catálogos del Sistema

A través de estos submenús se podrá dar mantenimiento a todos los catálogos que se utilizan tanto en el **Sistema Administrativo-Contable de Pagos**, como en el **Sistema para Trámite de Pagos**, para facilitar su mantenimiento se agrupan dependiendo el tipo de información.

Catálogos de Estructura: Hacen referencia a la estructura organizacional del banco, los cuales de muestran en el siguiente submenú: (figura 5.73)

Estructuras  
Bancos  
Regiones  
Plazas/Ubicaciones  
Departamentos  
Ciudades  
Zonas (Tarifas Viáticos)

Figura 5.73 Catálogos de Estructura

Catálogos Contables: Hacen referencia a todos los conceptos que se utilizarán para realizar las afectaciones y pólizas contables, los cuales de muestran en el siguiente submenú: (figura 5.74)

TESIS CON  
FALLA DE ORIGEN

Contables  
Cuentas Contables  
Item's  
Centros de Costos  
Conceptos

Figura 5.74 Catálogos Contables

Catálogos de Directorio: Hacen referencia a los Responsables de Revisión y a los beneficiarios de pagos, los cuales se muestran en el siguiente submenú: (figura 5.75)

Directorios  
Proveedores  
Responsables de Revisión  
Días Inhábiles

Figura 5.75 Catálogos de Directorio

## UTILERIAS

Para activar esta opción se deberá dar clic en el botón  **Utilerias** ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.76)



Figura 5.76 Menú de Utilerias del Sistema

A través de estos submenús se podrá dar mantenimiento a:

- La bitácora de accesos al sistema.
- Los usuarios del sistema.
- Los archivos índices de las tablas.
- Se podrán actualizar los catálogos que son utilizados por el Sistema para el Trámite de Pagos.

## PERSONALIZAR SISTEMA

Para activar esta opción se deberá dar clic en el botón  **Personalizar Sistema** ubicado en el menú principal, con lo cual se desplegará la siguiente pantalla: (figura 5.77)



Figura 5.77 Menú Personalizar Sistema

A través de este submenú se indicarán las rutas de acceso a todas las tablas que utiliza el sistema y se permitirá cambiar la contraseña a los usuarios.

Para solucionar cualquier tipo de problema que se presente durante la operación del

sistema, dando clic en la opción Personalizar Sistema **Acerca de...** se presentará una ventana en donde se le indicará al usuario a quien puede acudir para reportar esta situación: (figura 5.78)



Figura 5.78 Pantalla Acerca de..

### 5.3.2. MANUAL TÉCNICO

Explica la lógica de cómo interactúan todos los programas que conforman el sistema e incluye descripciones de estructuras de datos, diagramas de flujo, listados de programas, localización de los programas fuentes, la forma de compilar el proyecto y otros documentos.

Los objetivos de este manual son:

- Proporciona una descripción clara y precisa del sistema desarrollado.
- Ofrecer tras la puesta en marcha del sistema, un conjunto de documentos que permitan el mantenimiento y evaluación del mismo.
- Permite mostrar los datos de manera lógica y sus relaciones.
- Permite Ver el software sin tener que interactuar con él.

La importancia del Manual Técnico, radica en:

- Acortar la cantidad de horas requeridas por las nuevas gentes que se agreguen al equipo de trabajo para que aprendan el sistema antes de realizar el mantenimiento.
- Nos ayuda cuando surgen problemas.
- No se depende de una persona o un grupo reducido de personas para realizar modificaciones al software.
- Se logra un alto nivel de calidad.

Debido a que en capítulos anteriores se han descrito las tablas, la forma en que se elaboran los formularios, menús y demás elementos que conforman el sistema, solamente en los siguientes apartados se realizará una breve explicación de los elementos que intervienen en el sistema.

### 5.3.2.1 MANUAL TÉCNICO DEL SISTEMA PARA EL TRÁMITE DE PAGOS

El proyecto que contiene el menú principal y los programas fundamentales para su ejecución se llama ST\_MENU.PJX, a través del cual se genera el archivo ejecutable ST\_MENU.EXE, en la figura 5.79 se muestran todos los elementos que lo conforman:

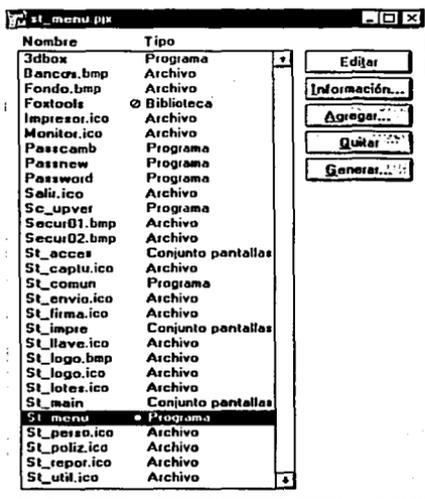


Figura 5.79 Programas y Archivos del Proyecto ST\_MENU

Dentro de este conjunto de elementos el archivo principal es: ST\_MENU.PRG, mediante el cual se realiza la llamada a las rutinas de validación de usuarios y se ejecuta el menú principal del sistema.

Como ya se ha mencionado anteriormente, el sistema se desarrollo utilizando proyectos individuales para facilitar su mantenimiento, los cuales son llamados desde el menú principal, siendo éstos:

TESIS CON  
FALLA DE ORIGEN

- ST\_CAPTU : Contiene todos los elementos para la captura de facturas.
- ST\_FIRMA : Contiene todos los elementos para la autorización de facturas.
- ST\_LOTES : Contiene todos los elementos para la generación de lotes de envío.
- ST\_UTIL : Contiene todos los elementos para el mantenimiento del sistema.

El proyecto ST\_CAPTU.APP está integrado por los siguientes elementos: (figura 5.80)

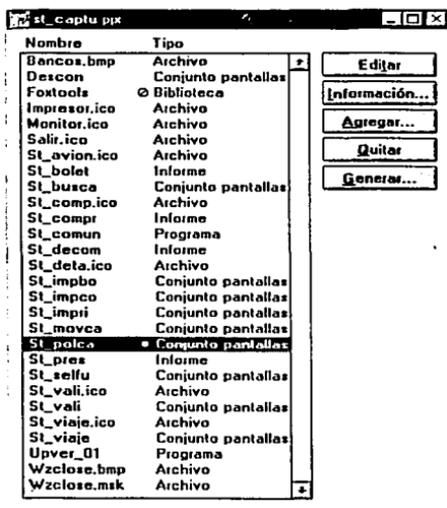


Figura 5.80 Programas y Archivos del Proyecto ST\_CAPTU

El formulario ST\_POLCA.SCX es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que permiten la captura y validación de facturas, incluyendo los que permiten la captura de gastos de viáticos, además de los informes de anticipos y comprobaciones de viáticos.

El proyecto **ST\_FIRMA.APP** está integrado por los siguientes elementos: (figura 5.81)

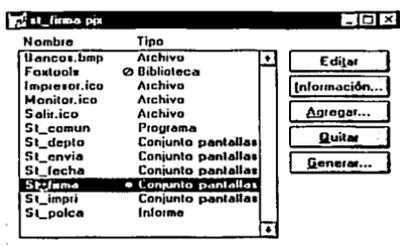


Figura 5.81 Programas y Archivos del Proyecto ST\_FIRMA

El formulario **ST\_FIRMA.SCX** es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan la asignación de los funcionarios que autorizan las facturas para su pago.

El proyecto **ST\_LOTES.APP** está integrado por los siguientes elementos: (figura 5.82)

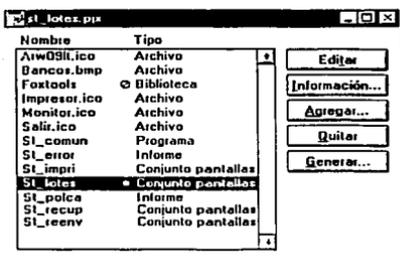


Figura 5.82 Programas y Archivos del Proyecto ST\_LOTES

El formulario ST\_LOTES.SCX es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan la asignación y generación de los lotes de facturas que serán enviados al Área de Central de Pagos, así como de los informes soporte de la información enviada.

El proyecto ST\_UTIL.APP está integrado por los siguientes elementos: (figura 5.83)



Figura 5.83 Programas y Archivos del Proyecto ST\_UTIL

El programa ST\_UTIL.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

TESIS CON  
FALLA DE CUBIEN

Dentro de estos archivos se encuentran los formularios que realizan el mantenimiento y respaldo de las tablas del sistema, así como de los informes de las facturas y anticipos de viáticos que se tienen registrados.

La ubicación física de todos los proyectos, incluyendo formularios, informes, menús, programas y archivos de imágenes BMP e ICO se ilustra en la figura 5.84:

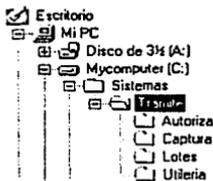


Figura 5.84 Ubicación de los Archivos del Sistema

Las tablas del sistema para el ambiente de pruebas se ubican en la siguiente ruta: (figura 5.85)

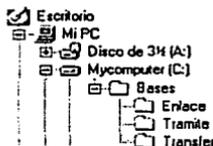


Figura 5.85 Ubicación de las Tablas del Sistema

Para las tablas en producción se recomienda respetar la misma estructura de árbol en el servidor, para facilitar su ubicación y mantenimiento.

TESIS CON  
FALLA DE ORIGEN

### 5.3.2.2 MANUAL TÉCNICO DEL SISTEMA ADMINISTRATIVO-CONTABLE DE PAGOS

El proyecto que contiene el menú principal y los programas fundamentales para su ejecución se llama SC\_MENU.PJX, a través del cual se genera el archivo ejecutable SC\_MENU.EXE, en la figura 5.86 se muestran todos los elementos que lo conforman:

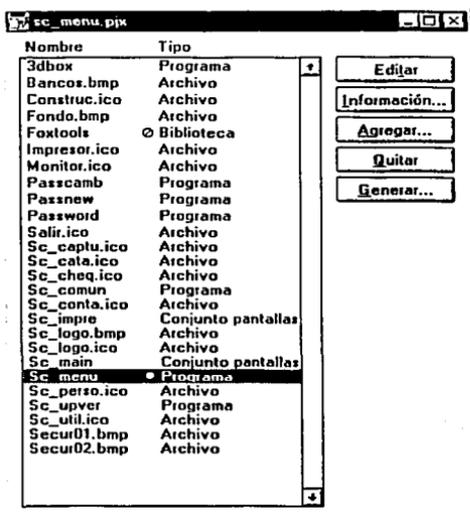


Figura 5.86 Programas y Archivos del Proyecto SG\_MENU

Como ya se ha mencionado anteriormente, el sistema se desarrollo utilizando proyectos individuales para facilitar su mantenimiento, los cuales son llamados desde el menú principal, siendo éstos:

TESIS CON  
FALLA DE ORIGEN

- SC\_CAPTU : Contiene todos los elementos para la recepción de facturas y elaboración de contra-recibos.
- SC\_CONTA : Contiene todos los elementos para elaboración cheques y abonos a cuentas de cheques.
- SC\_ICBS : Contiene todos los elementos para la elaboración y afectación de las pólizas contables.
- SC\_CATA : Contiene todos los elementos para el mantenimiento de los catálogos del sistema.
- SC\_UTIL : Contiene todos los elementos necesarios para el mantenimiento del sistema.
- SC\_PERSO : Contiene todos los elementos para la personalización del sistema.

A continuación se describen de manera general los elementos que conforman cada uno de los proyectos antes citados.

Dentro del proyecto **SC\_CAPTU.APP** se encuentra el formulario **SC\_CAPTU.SCX**, desde el cual se hace la llamada a todos los demás elementos.

En este proyecto se encuentran todos los formularios utilizados para la recepción automática de los lotes, así como para la captura e impresión de los contra-recibos, la asignación de los Responsables de Revisión y todas las funciones inherentes a la Ventanilla de pagos.

La figura 5.87 ilustra los elementos que conforman el proyecto **SC\_CAPTU.APP**:



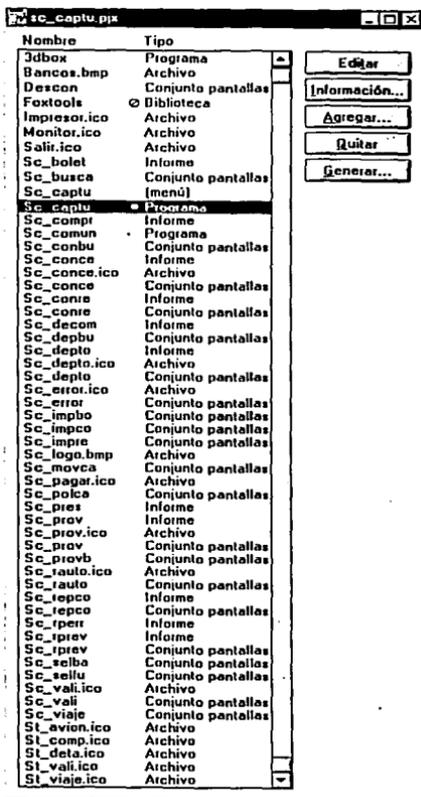


Figura 5.87 Programas y Archivos del Proyecto SC\_CAPTU

El proyecto SC\_CONTA.APP está integrado por los siguientes elementos: (figura 5.88)

**TESIS CON  
FALLA DE ORIGEN**

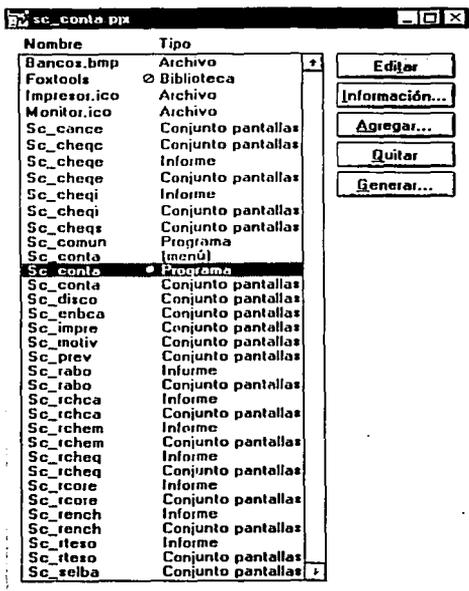


Figura 5.88 Programas y Archivos del Proyecto SC\_CONTA

El programa SC\_CONTA.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan la selección de los contra-recibos a pagar, los informes de la solicitud de recursos a Tesorería, la elaboración previa de la afectación contable, así como la elaboración de los cheques y los abonos a cuentas de cheques.

**TESIS CON  
FALLA DE ORIGEN**

El proyecto SC\_ICBS.APP está integrado por los siguientes elementos: (figura 5.89)

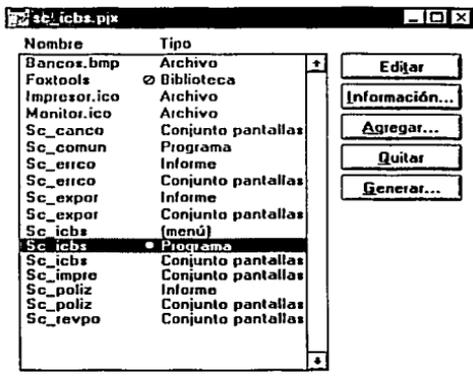


Figura 5.89 Programas y Archivos del Proyecto SC\_ICBS

El programa SC\_ICBS.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan la afectación contable de los pagos efectuados, los informes que realizan las pólizas contables y los programas que generan el archivo que será enviado al Sistema Central de Contabilidad.

Este proyecto es la parte medular del **Sistema Administrativo-Contable de Pagos**, ya que con la utilización de los conceptos de pago, se logró generar las guías contables de todos los movimientos de gasto, eliminando así los errores que se tenían durante el llenado de las pólizas contables, contribuyendo además a la reorientación de actividades del personal que sólo se dedicaba a la captura de los movimientos contables.

TESIS  
FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN

El proyecto SC\_CATA.APP está integrado por los siguientes elementos: (figura 5.90)

Nombre	Tipo
Adios	Programa
Adios	Conjunto pantallas
Banco.bmp	Archivo
Construc.ico	Archivo
Fonttools	Biblioteca
Impreso.ico	Archivo
Monitor.ico	Archivo
Salir.ico	Archivo
Sc_banco	Informe
Sc_banco.ico	Archivo
Sc_banco	Conjunto pantallas
Sc_cata	(menu)
Sc_cata	Programa
Sc_cenco	Informe
Sc_cenco.ico	Archivo
Sc_cenco	Conjunto pantallas
Sc_ciuda	Informe
Sc_ciuda	Conjunto pantallas
Sc_comun	Programa
Sc_conce	Informe
Sc_conce.ico	Archivo
Sc_conce	Conjunto pantallas
Sc_ctas	Informe
Sc_ctas.ico	Archivo
Sc_ctas	Conjunto pantallas
Sc_depbu	Conjunto pantallas
Sc_depto	Informe
Sc_depto.ico	Archivo
Sc_depto	Conjunto pantallas
Sc_dian	Informe
Sc_dian.ico	Archivo
Sc_dian	Conjunto pantallas
Sc_impie	Conjunto pantallas
Sc_items	Informe
Sc_items.ico	Archivo
Sc_items	Conjunto pantallas
Sc_logg.bmp	Archivo
Sc_plaza	Informe
Sc_plaza.ico	Archivo
Sc_plaza	Conjunto pantallas
Sc_prov	Informe
Sc_prov.ico	Archivo
Sc_prov	Conjunto pantallas
Sc_prov	Conjunto pantallas
Sc_reg	Informe
Sc_reg.ico	Archivo
Sc_reg	Conjunto pantallas
Sc_turno	Informe
Sc_turno.ico	Archivo
Sc_turno	Conjunto pantallas
Sc_zonas	Informe
Sc_zonas	Conjunto pantallas

Figura 5.90 Programas y Archivos del Proyecto SC\_CATA

El programa SC\_CATA.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan el mantenimiento a todos los catálogos que integran el sistema, así como sus respectivos informes.

El proyecto SC\_UTIL.APP está integrado por los siguientes elementos: (figura 5.91)

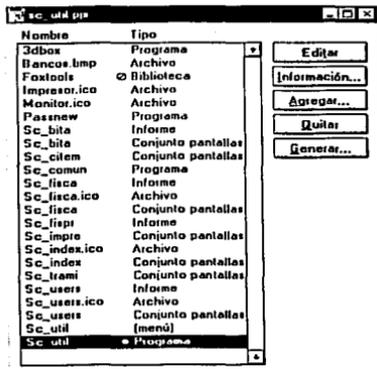


Figura 5.91 Programas y Archivos del Proyecto SC\_UTIL

El programa SC\_UTIL.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que realizan el mantenimiento y respaldo de las tablas del sistema, así como el mantenimiento a los usuarios.

El proyecto SC\_PERSO.APP está integrado por los siguientes elementos: (figura 5.92)

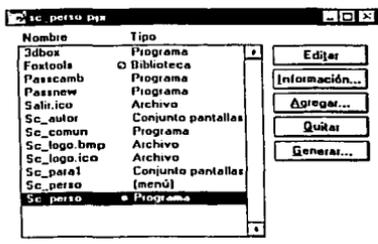


Figura 5.92 Programas y Archivos del Proyecto SC\_PERSO

El programa SC\_PERSONA.PRG es el archivo principal desde el cual se hace la llamada a todos los demás elementos.

Dentro de estos archivos se encuentran los formularios que permiten establecer los parámetros generales del sistema para su correcto funcionamiento.

La ubicación física de todos los proyectos, incluyendo formularios, informes, menús, programas y archivos de imágenes BMP e ICO se ilustra en la figura 5.93:

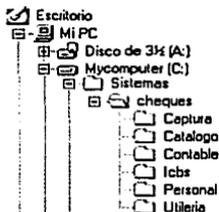
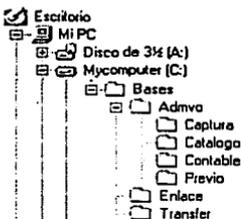


Figura 5.93 Ubicación de los Archivos del Sistema

Las tablas del sistema para el ambiente de pruebas se ubican en la siguiente ruta: (figura 5.94)



**TESIS CON  
FALLA DE ORIGEN**

Figura 5.94 Ubicación de las Tablas del Sistema

Para las tablas en producción se recomienda respetar la misma estructura de árbol en el servidor, para facilitar su ubicación y mantenimiento.

## 5.4 MANTENIMIENTO AL SISTEMA

El término mantenimiento queda definido según la norma ISO 8402 y la norma IRAM 36052 como un conjunto de atributos referidos al esfuerzo necesario para realizar modificaciones específicas, las cuales pueden incluir correcciones, mejoras o adaptaciones del software, originados por cambios en el entorno y cambios en las especificaciones de funciones y requerimientos.

Las actividades de mantenimiento que se llevan a cabo tras poner una aplicación en producción se pueden resumir en:

1. La primer actividad de mantenimiento es debida a que no es razonable asumir que las pruebas del software hayan descubierto todos los errores de una aplicación. Durante el uso de cualquier programa, se encontrarán errores. El proceso que incluye el diagnóstico y la corrección de uno o más errores se denomina mantenimiento correctivo.
2. La segunda actividad de mantenimiento se produce por el rápido cambio tecnológico de la informática. Por lo tanto, el mantenimiento adaptativo, modifica el software para que funcione adecuadamente con su entorno.
3. La tercera actividad de mantenimiento se produce cuando una aplicación tiene éxito. A medida que se usa el software, se recibe de los usuarios recomendaciones sobre nuevas posibilidades, sobre modificaciones de funciones ya existentes y sobre mejoras en general, esto es el mantenimiento adaptativo.
4. La cuarta actividad de mantenimiento se da cuando se cambia el software para mejorar una futura facilidad de mantenimiento, o para proporcionar una base óptima para futuras mejoras, comúnmente llamada mantenimiento perfectivo. Esta actividad está caracterizada por las técnicas de reingeniería.

Es por ello que se debe tener muy en cuenta los nuevos requisitos del usuario, para adaptar o perfeccionar el software.

#### 5.4.1 PERFECTIVO

Una de las principales preocupaciones surgidas durante la elaboración y culminación del sistema por parte del usuario final, consistió en cómo se realizará el mantenimiento del sistema en el sentido de incluir nuevas funciones y procesos.

Dicha situación se solventará durante el proceso de mejoramiento del sistema (mantenimiento) que se realizará durante todo el tiempo de vida de éste.

Para facilitar esta labor, el sistema se diseñó de tal manera que cualquier nuevo requerimiento del tipo:

- Adecuaciones a formularios e informes ya existentes.
- Diseño de nuevos informes estadísticos.
- Diseño de nuevos formularios de captura y procesos.
- Inclusión de nuevos menús.

Se podrán realizar de manera rápida, debido a que el sistema consta de archivos tipo APP, que son proyectos compilados de forma individual y ejecutados desde un solo archivo EXE, los cuales se modificarán y probarán desde un ambiente de pruebas especialmente diseñado para este propósito, ubicado dentro del Área de Desarrollo de Sistemas, antes de ser liberados a producción.

#### 5.4.2 ADAPTATIVO

Un aspecto importante que no se debe perder de vista es el avance vertiginoso de la tecnología, principalmente el referente al software y a la forma de acceder y mostrar información.

Por lo que se vuelve de vital importancia que las aplicaciones se puedan integrar a la tecnología de Internet y que sus bases de datos se puedan conectar a sitios WEB.

Al respecto podemos decir que FoxPro en su más reciente versión Visual FoxPro 7.0, cuenta con herramientas que proporcionan velocidad y flexibilidad permitiéndole actuar como base de datos del lado del servidor e integrar la funcionalidad de Internet en las aplicaciones existentes.

La implementación del Sistema para el Trámite de Pagos y el Sistema Administrativo-Contable de Pagos se realizó en la versión 2.6 de FoxPro debido a que su desarrollo se orientó hacia las Instituciones Bancarias con falta de presupuesto e infraestructura tecnológica, pero puede perfeccionarse y adaptarse para aquellas Instituciones que si cuentan con una mejor infraestructura, dentro de las principales adecuaciones que se tendrían que realizar, tenemos:

- Capacidad de poder enviar por correo electrónico los lotes de las facturas, para solicitar su pago.
- Posibilidad de poder actualizar el sistema mediante transferencias FTP.
- Accesar a un sitio WEB (Intranet) para poder consultar el estatus de pago de las facturas enviadas al Área de Central de Pagos.

Finalmente, en lo que respecta a la posibilidad de poder perfeccionar el sistema, cabe hacer mención que Visual FoxPro es una excelente plataforma para la construcción de aplicaciones, debido a su velocidad extremadamente rápida para la recuperación de datos locales y su flexibilidad cuando se acceden datos remotos, así como a su poderoso lenguaje orientado a bases de datos y a objetos, el cual es ideal para la programación compleja de acceso a datos y a la lógica de las reglas de negocios.



**CONCLUSIONES**

## CONCLUSIONES

El desarrollo e implementación del **Sistema para el Trámite de Pagos** y el **Sistema Administrativo-Contable de Pagos**, me permitió obtener una serie de experiencias que puedo clasificar en tres grandes grupos:

- Laboral
- Académica
- Personal

### Laboral

La puesta en marcha del sistema mejoró significativamente el tiempo que se requería para la solicitud del pago de algún servicio, además, se logró unificar y normalizar los criterios para la solicitud de pagos, así como la estandarización de la emisión de cheques y la elaboración de las pólizas contables.

Se creó un sistema único, ágil y sencillo, tanto para las Áreas solicitantes de pagos, como para el Área Central de Pagos, logrando al respecto:

- Un sistema acorde a las necesidades del usuario.
- Reducción de costos.
- Eliminación de duplicidad de funciones.
- Sistema gráfico amigable al usuario.
- El diseño a través de APP, permite la incorporación gradual de nuevos procesos y funciones.

Por lo anterior, se puede decir que el Sistema cumple con los objetivos planteados al inicio del desarrollo, cubriendo completamente las necesidades de procesamiento de información y proporcionando un alto grado de confiabilidad en ésta.

## **Académica**

Este proyecto me permitió poner en práctica lo aprendido durante mi educación profesional, debido a que el desarrollo del sistema se hizo desde identificar y aterrizar las necesidades del usuario, poder entrevistar y realizar el levantamiento de información, analizar y diseñar posibles soluciones hasta llegar a la óptima, implementar y capacitar al usuario.

Las materias que principalmente tuve oportunidad de aplicar en este proyecto, fueron:

**Ingeniería de Programación:** A través de sus métodos y herramientas se realizó el análisis, diseño y planificación del proyecto.

**Bases de Datos:** Mediante sus modelos de datos se definió el óptimo de acuerdo a las especificaciones y necesidades de las Instituciones Bancarias en cuestión.

**Administración de Centros de Cómputo:** Mediante sus conceptos definidos para la organización y administración, se diseñaron los controles de accesos a la información, así como la distribución de los clientes y el servidor.

**Sistemas Operativos:** Conociendo el funcionamiento de los diferentes tipos de sistemas operativos, fue posible realizar la transferencia de información entre el sistema Windows 98/2000 y UNIX.

**Redes de Computadoras:** Se utilizó para conocer el protocolo de comunicación a través del cual se realizó la conexión de los equipos de cómputo, su configuración y conectividad.

La conjunción de estas materias, junto con toda la trayectoria académica que tuve, se traducen en la capacidad de razonar y entender cualquier problema para su posterior solución, cuyo objetivo radica en facilitar el trabajo y actividades de otras personas.

### Personal

La elaboración de este proyecto me permitió realizar un trabajo escrito, el cual por diversas razones no había efectuado, dentro de estas causas, la principal era que no encontraba algún tema de mi interés, pero éste, me dio la oportunidad de poder crear algo desde sus inicios y culminarlo hasta satisfacer las necesidades de toda un Área.

Durante la elaboración del proyecto, pude interactuar con abogados, contadores y economistas, que me dieron posibilidad de adquirir otro tipo de conocimientos, permitiéndome expresar mis ideas, así como poder captar y plasmar en unas líneas de código sus requerimientos.

Finalmente, me da la satisfacción de poder decir que pese a todos los contratiempos que tuve, he terminado mi educación académica como Ingeniero en Computación, que desde mi punto de vista es una de las carreras más versátiles, debido a que la tecnología avanza día a día, lo cual me permite seguir adquiriendo nuevos conocimientos.



## Bibliografía

- Roger, S. Pressman, 1993. *Ingeniería del software*, 3ª. Edición, McGraw Hill.
- Korth, H., 1993. *Fundamentos de bases de datos*, 2ª Edición, McGraw-Hill.
- Date, C., 1993. *Introducción a los sistemas de bases de datos, Volumen 1*, 5ª Edición, McGraw-Hill.
- David, M. Kroenke, 1996. *Procesamiento de Bases de Datos, Fundamentos, diseño e instrumentación*, 5ª Edición, Prentice Hall.
- Alice Y. H., Tsai, 1990. *Sistemas de Bases de Datos, Administración y uso*, Prentice Hall.
- Kendall & Kendall, 1997, *Análisis y diseño de sistemas*, 3ª edición, ed. Pearson educación.
- Charles Siegel, 1994, *Cómo usar FoxPro 2.5*, 1ª edición, ed. Megabyte, Noriega Editores.