

64  
3j



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

FACULTAD DE INGENIERIA

**SISTEMA DE CONTROL Y AUTOMATIZACION  
DE DOCUMENTOS A GRAN ESCALA  
(SICADE)**

**T E S I S**  
QUE PARA OBTENER EL TITULO DE:  
**INGENIERO EN COMPUTACION**  
P R E S E N T A N :  
**MARTHA LAURA LOPEZ COQUIS  
JUAN VLADIMIR PADILLA CERON  
RAUL IGNACIO RESENDIZ GARCES  
GUILLERMO SOTO TINOCO**

**DIRECTORA DE TESIS  
ING. GLORIA MATA HERNANDEZ**



MEXICO, D. F.

FEBRERO 1996

**TESIS CON  
FALLA DE ORIGEN**

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

## **DEDICATORIAS Y AGRADECIMIENTOS**

---

Dedico este trabajo:

A la pareja que me formó como persona,  
profesionista, amiga, hermana  
y principalmente como hija.

A mis padres por sus desvelos,  
dedicación, cuidados, consejos,  
pero principalmente por su gran ejemplo  
y entrega total

A el Profr. Francisco López Galicia y

A la Profra. Ma. Hilda D. Coquis Rosas

A mis hermanos:

Armando, Gustavo y Héctor Francisco  
por este gran amor, amistad,  
cariño y apoyo que ha existido  
entre nosotros.

A mis sobrinos:

Armando Tonatiuh y Francisco Miztli,  
por esa alegría, carisma, espontaneidad y  
felicidad que le dieron a nuestro hogar  
con su presencia.

A la persona que en estos últimos años  
me ha impulsado como pareja,  
persona, mujer, pero sobre todo como profesionista.

A Juan Vladimir Padilla Cerón,  
por que llegó a mi vida en el momento preciso,  
primero como compañero,  
posteriormente fue un gran amigo,  
pero ahora es mi presente.

Durante todo este tiempo,  
me he encontrado con distintas personas  
que forman parte importante de mi vida,  
debido al apoyo, orientación, consejos,  
guía, confianza, pero sobre todo por estar junto a mi  
en los momentos más relevantes de mi vida,  
por eso quiero agradecerles a cada uno ellos  
por brindarme su amistad.

En especial a Paty y Mely por sus consejos y su amistad.  
Y a mis compañeros Raúl y Guillermo por el trabajo que  
desarrollamos.

Martha Laura López Coquis.

## **A DIOS:**

Por la salud física y mental de cada día.

## **A Delia Cerón Vergara:**

Mujer de espíritu indomable, que me enseñó a tener la mirada siempre hacia adelante y la frente muy en alto.

Gracias mamita querida por todo cariño, tu ternura y valor, por luchar y darme la vida e impulsarme a ser un hombre de bien.

Gracias mamita linda por tu apoyo, consejos, energías, lágrimas, regaños, palabras y risas pero sobre todo por la ciega confianza en mí, que me dio la seguridad de lograr siempre lo yo quisiera. Esta tesis es la cosecha de tu siembra.

## **A Gadiel García Cerón y Jimena García Cerón:**

Por existir, por ser una razón de vivir, de superarse, de triunfar.

Gracias mis nenes queridos por esperarme cada día, confiar en mí y compartir sus vivencias conmigo. Muchas gracias por su ingenio, su alegría, sus sentimientos y su comprensión, pero sobre todo por escucharme y dejarse querer como los quiero.

## **A Juan Cerón Guerrero:**

Por brindarme tanto cariño, consentimiento, tiempo y atención. Por el refresco y las golosinas. Por enseñarme a amar a Juchi. Por enseñarme a trabajar la madera y tantas cosas más. Gracias abuelito querido, donde quiera que estés mil gracias por dejar que estuviera a tu lado siempre, ya que conocerte fue lo más maravilloso que me sucedió. Canolito lindo, esta tesis sólo es el principio de tu gran obra, gracias "papá abuelito"...

## **A Amparo Vergara Suárez:**

Por adorarme como sólo una gran abuela puede hacerlo. Por ese ejemplo de tenacidad, de no dejarse vencer. Gracias abuelita linda por tus cuidados, guisos, atenciones y preferencias. Para tí todo mi corazón y gratitud hasta donde estés.

## **A Martha Laura López Coquis:**

Por darme la oportunidad de sentir lo que es la amistad, amor, ternura, cariño y apoyo. Por dejarme ocupar un lugarcito en tu corazón y por impulsarme en cada momento a ser mejor, por darme tanta ilusión y seguridad. Gracias mi nenita bonita, mi consentida, mi niña...

## **A Javier García Olivares:**

Por asumir un difícil papel, por dejar que yo tuviera papá. Por no intervenir en mis decisiones y respetar mis deseos. Por enseñarme los secretos del fútbol y por sus grandes sentimientos, gracias papá Javier.

A mis tíos y tías:

"Malena" Cerón, Pablo Reynoso, Ana Cerón V., Porfirio Cerón y Elba Cerón, por esos años setentas maravillosos, por enseñarme la música de los beatles, por enseñarme a manejar el coche y a nadar, por el cariño, cuidados y confianza. Por hacerme sentir el hermano menor de la gran familia.

A mis tíos Ana Cerón G. y Rodolfo Calvo:

Por brindarme su "terruño" en Juchi, por soportarme y quererme cada vez que los visito. Ustedes fueron también un gran motivo para superarme, gracias mis queridos "segundos abuelos".

A Antonio García:

Por su amistad, ejemplo, impulso y cariño. Gracias tío Toño.

A mis primos:

Juan Pablo, Marlen, Israel, Paco, Juan Manuel, Daniel, Adriana, Lizbeth, Yenisey, Gerardo y Mayra, por tanta diversión, enojos, risas y cariño.

A mis queridos amigos:

Enrique Morales Flores, por el gran ejemplo de coraje y empuje ante la vida. Por inducirme al buen camino, por dejar ser un hermano mayor para mi, por su gran alegría, por hacer inolvidable el C.C.H. Sur y la Facultad. Gracias "Ron Enrique".

Samuel Mejía Bautista, por su gran nobleza y corazón. Por el fino humor. Por compartir sus conocimientos. Por tantos y buenos momentos, gracias "Samy".

Juan Galeana, Alejandra Pérez, Lucía Pineda, Arturo Beltrán y Octavio Amador.

A mis maestros:

Ing. *Juan Aguilar*, por demostrarme que no hay obstáculo en la vida que no se pueda superar. Por su sabiduría y paciencia, gracias querido maestro.

Ing. *Torres H.*, por su sabiduría y certeros consejos.

Ing. *Ian García Tsao*, por el orden, la metodología, la seriedad, el cumplimiento y el profesionalismo. Gracias por todos los conocimientos y la formación otorgada.

Ing. *Gloria Mata*, por cumplir cabalmente con el principio básico de un buen profesor: "dirigir el aprendizaje". Así como compartir su valioso tiempo y visión.

**A la FACULTAD DE INGENIERÍA de la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO:**

Por ayudar en mi formación, crecimiento y desarrollo. Por los cinco años de alojamiento. Por las instalaciones y las porras. Por el orgullo de ser "escorpión rojo" y "puma". Por introducirme en la sangre el "Por mi raza hablará el espíritu".

Juan Vladimir Padilla Cerón.

**A mis padres:**

Este es parte del triunfo que han obtenido por haberme apoyado durante el transcurso de mi carrera. Los días largos y las veladas interminables han terminado. Gracias por haberme brindado su cariño y comprensión, por sus consejos, ya que sin todo esto nunca hubiese llegado al final de este camino por el cual me han guiado, nunca dejaré de agradecer a Dios por haberme dado unos padres tan maravillosos como ustedes. Gracias a Pedro y Tomasa.

**A mis Hermanos:**

Recuerdo aquellos momentos cuando fuimos pequeños, donde soñábamos y reíamos, hoy se ha cumplido unos de esos sueños, el cual nuevamente vuelvo a compartir con ustedes; los juegos acabaron, pero aún puedo reír y acordarme de lo grato que es cuando un hermano triunfa, ya que el triunfo es de todos. Gracias Eréndira, Cuauhtémoc y Ricardo.

**Grupo ITS:**

Hace un año aproximadamente conocí a un grupo de personas entusiastas y aguerridas, las cuales me apoyaron en los momentos mas difíciles del desarrollo de este documento, el cual hoy culminó y agradezco todas las atenciones recibidas. Gracias a: Patty, Jaime, ABG, Griss, Edith, a todo Soporte Técnico, Administración y Ventas.

Raúl Ignacio Reséndiz Garcés.

**Dedico este trabajo:**

**A mis padres, por todo el apoyo y cariño que nos han brindado a mi y a mis hermanos, por la confianza depositada en cada uno de nosotros y principalmente por alentarnos a ser cada día mejores.**

**Guillermo Soto Tinoco.**



# PRÓLOGO

---

El manejo de documentos en una institución o empresa implica el recibirla, dar respuesta y enviarla a su destino. Si la institución es pequeña, esto se realiza manualmente sin ningún problema. Sin embargo, hay instituciones en las que se manejan grandes volúmenes de documentos, como es el caso de Secretarías de Estado, Bancos, Aseguradoras, etc., en las que el flujo de información es continuo y tiene que pasar a través de diversas instancias para completar la información y secuencia, y dar por concluido el asunto.

Esto puede realizarse el mismo día o bien puede llevarse meses para su gestión total. Los documentos pueden ser de una hoja o pueden ser libros históricos que abarquen varios tomos. Bajo esta circunstancia el dar seguimiento a un documento se vuelve totalmente complicado, no confiable e involucra mayor personal para su control y más aun cuando es información confidencial.

La mayoría de las empresas dan seguimiento a los documentos de una manera poco eficiente, por lo que surge la necesidad de buscar una solución que permita agilizar el proceso. Pueden existir muchas soluciones para mejorar la problemática como, utilizar más personal para repartirse la carga de trabajo, lo cual no sería una solución muy óptima; auxiliarse de herramientas como una computadora o impresora para lograr cierto grado de automatización y simplicidad en las tareas; o desarrollar todo un sistema computarizado que efectúe todo el proceso.

De las posibles soluciones la que plantea mejores beneficios es la de automatizar todo el proceso ya que permitiría tiempos de respuesta mas cortos y facilitaría el trabajo.

El manejo de documentos en forma manual en una institución que recibe mucha documentación, tiene muchos inconvenientes ya que el proceso es muy lento y deficiente. Esta problemática se resuelve con un sistema computarizado a nivel empresarial que permita el control, automatización y administración de documentos, el cual de seguimiento del tramite desde su inicio hasta el final, reportando todos los parámetros involucrados en la secuencia e implementado mediante digitalización de los documentos y en una plataforma capaz de manejar grandes volúmenes de información.

Para llevar a cabo la automatización se requiere de una metodología de desarrollo que permita cumplir con los objetivos y las expectativas, para ello la metodología que se emplea es la de análisis y diseño estructurado.

El sistema desarrollado esta planeado para soportar un volumen considerable de información, pero claro esta que al pasar el tiempo y dependiendo de que tan intensamente se utilice el sistema, la capacidad del sistema se agotara, para ello se eligió una plataforma que permita futuros crecimientos ya sea que se le aumenten más recursos de almacenamiento y procesamiento o se emigre a un sistema mas grande.

El código también puede sufrir cambios y adecuaciones a futuros requerimientos y necesidades, para ello se prevé un buen diseño del sistema y documentación detallada de los procesos, que permitan facilitar el mantenimiento del código, ya sea que se le agreguen nuevos procesos o se modifiquen los ya existentes.

Por otro lado, el trabajo está estructurado en 8 capítulos y 3 apéndices.

En el capítulo 1, se muestra el manejo y el seguimiento de la documentación en forma manual, la problemática que implica el dar continuidad a un asunto de acuerdo al número de turnados, los cuales se deben controlar en forma manual, además que es la coordinación administrativa, lo que requiere y como funciona.

En el capítulo 2 se exponen las ventajas y desventajas que se tienen con la automatización del proceso, analizando cómo es que se lleva a cabo el proceso manual y como la automatización simplifica estas tareas haciéndolas más eficientes y rápidas.

Menciona los requerimientos y costos de Software y Hardware para su implementación. Se muestra una configuración monousuario y una configuración en red, indicando las ventajas y desventajas de cada una. Se incluyen también las características generales del sistema

Al final del capítulo se muestran los objetivos del sistema y las etapas del proceso manual que se van a automatizar y sus características.

En el capítulo 3, se aborda la metodología del análisis y diseño estructurado que involucra el desarrollo de diversos tipos de modelos tales como el "diagrama de flujo de datos" y "modelo entidad-relación", los cuales quedarán concentrados en lo que se conoce como "especificación funcional" que a su vez, es el material indispensable para diseñar mediante el "diagrama de transición" la arquitectura general del sistema y finalmente para escribir y probar los programas generados.

En el capítulo 4, se llevan a la práctica los aspectos teóricos que se trataron en el capítulo anterior, ya que se esquematiza todo el sistema y se logra identificar la frontera máquina-hombre dando paso al diseño tanto del sistema como de la base de datos que este utiliza.

En el capítulo 5, se provee de un manual de capacitación para que toda aquella persona que desee aprender a operar el sistema tenga material adecuado, que lo familiarice con el SICADE.

No obstante, cuando el usuario ya tiene cierta relación con los procedimientos que rigen el control de la documentación, es conveniente ver el capítulo 6, ya que en este se ofrece una explicación detallada de las pantallas y todas las herramientas con que cuenta SICADE.

También es de mucha utilidad un manual de instalación, ya que en este se concentran los aspectos más importantes de qué se necesita para que el sistema sea instalado adecuadamente y lograr así un buen desempeño del mismo.

En el capítulo 7 se enuncian los pasos necesarios para instalar e integrar la aplicación, en el ambiente operativo que se haya seleccionado.

La instalación del sistema puede ser en red o en modo monousuario. Para fines prácticos se considera la instalación en un ambiente monousuario, haciendo referencias, aclaraciones y cambios para los ambientes de sistemas abiertos.

En el capítulo 8, se muestran únicamente unos ejemplos de prueba con el fin de no extender el presente material mas allá de lo necesario.

En las conclusiones se manifiestan los problemas y obstáculos que se tuvieron que superar para poder conformar el sistema y la presente documentación, así como las satisfacciones obtenidas al observar cómo los usuarios adoptan el sistema como suyo.

Evidentemente los apéndices no pueden faltar en un trabajo serio y formal, ya que siempre se debe dar la posibilidad al lector de profundizar en el detalle de la información. Tales apéndices contemplan un listado del código de los productos para desarrollo de ORACLE, así como un listado de las figuras y otro de las tablas.

Finalmente, se incluye un glosario de términos que de acuerdo a la experiencia, contempla el significado de palabras o frases técnicas que desde el punto de vista de cómputo son comunes y no así para un lenguaje cotidiano.

	Página
<b>Prólogo</b>	<b>1</b>
<b>Capítulo 1 Control y administración de la documentación</b>	<b>1</b>
1.1 Antecedentes . . . . .	2
1.2 Problemática actual . . . . .	4
1.3 La coordinación administrativa . . . . .	5
<b>Capítulo 2 Automatización del manejo de documentos</b>	<b>11</b>
2.1 El proceso de la automatización . . . . .	12
2.2 Requerimientos del ambiente de implementación . . . . .	15
2.3 Costos . . . . .	19
2.4 Características generales del nuevo sistema . . . . .	20
<b>Capítulo 3 Teoría del análisis y diseño estructurado moderno de sistemas</b>	<b>21</b>
3.1 Utilización de modelos . . . . .	22
3.2 El diagrama de flujo de datos (DFD) . . . . .	22
3.3 Guía para la construcción de un DFD . . . . .	27
3.4 Modelo conceptual de datos . . . . .	32
3.5 Representación del diagrama Entidad Relación (E-R) . . . . .	36
3.6 Normalizar el modelo de datos . . . . .	40
3.7 Conceptos de base de datos relacional . . . . .	43
3.8 Normalización de tablas . . . . .	45
3.9 Diagrama de transición de estados . . . . .	47
3.10 El diccionario de datos . . . . .	55
3.11 Diseño estructurado moderno . . . . .	58
3.12 Diseño de la base de datos . . . . .	60

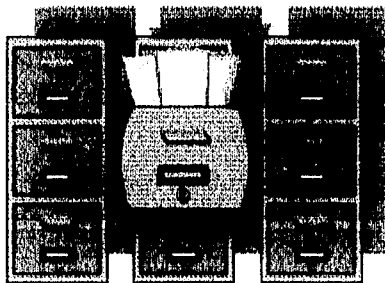
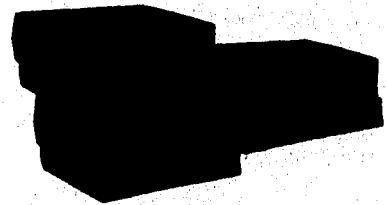
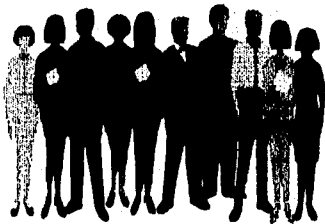
<b>Capítulo 4</b>	<b>Análisis y diseño del sistema de control y automatización de documentos a gran escala (SICADE)</b>	<b>67</b>
4.1	DFD-0 Sistema de control y automatización de documentos a gran escala (SICADE)	68
4.2	Recibir los documentos	70
4.3	Controlar (Digitalizar) los documentos	71
4.4	Generar los reportes	73
4.5	Mantener los catálogos	75
4.6	Mantener el sistema	76
4.7	Diseño de la base de datos	77
4.8	Modelo entidad relación	79
4.9	El diccionario de datos del SICADE	82
4.10	El diccionario de la base de datos del SICADE	85
<b>Capítulo 5</b>	<b>Manual de capacitación del sistema de control y automatización de documentos a gran escala (SICADE)</b>	<b>89</b>
5.1	Descripción de los estados o etapas de la información	91
5.2	Descripción de la funcionalidad del sistema	91
5.3	Descripción de los parámetros requeridos por pantalla del SICADE	97
<b>Capítulo 6</b>	<b>Manual del usuario del sistema de control y automatización de documentos a gran escala (SICADE)</b>	<b>105</b>
6.1	Inicio del SICADE	106
6.2	Altas, bajas y actualización de catálogos	108
6.3	Consultas de catálogos	119
6.4	Documentos	128
6.5	Reportes	142
6.6	Utilerías	149
6.7	Salir	154

	<b>Página</b>
<b>Capítulo 7</b>	
<b>Manual de instalación del sistema de control y automatización de documentos a gran escala (SICADE)</b>	<b>155</b>
7.1 Componentes y arquitectura de hardware . . . . .	156
7.2 Arquitectura de software . . . . .	158
7.3 Instalación de software . . . . .	188
7.4 Mantenimiento de hardware y software . . . . .	193
<b>Capítulo 8</b>	
<b>Plan de pruebas</b>	<b>199</b>
8.1 Tipos de prueba . . . . .	200
8.2 Prueba de menús principales . . . . .	202
8.3 Prueba de pantallas . . . . .	205
8.4 Pruebas funcionales . . . . .	218
<b>Capítulo 9</b>	
<b>Conclusiones</b>	<b>225</b>
<b>Apéndice A</b>	
<b>Listado parcial del sistema</b>	<b>229</b>
A.1 Listado de formas . . . . .	231
A.2 Listado de user exit . . . . .	232
A.3 Listado de reportes . . . . .	233
A.4 Ejemplo de una forma . . . . .	234
A.5 Listado de 16 programas user exit . . . . .	241
<b>Apéndice B</b>	
<b>Listado de figuras</b>	<b>281</b>
<b>Apéndice C</b>	
<b>Listado de tablas</b>	<b>289</b>
<b>Glosario</b>	<b>295</b>
<b>Bibliografía</b>	<b>303</b>

# CAPÍTULO 1

## CONTROL Y ADMINISTRACIÓN DE LA DOCUMENTACIÓN

---





## 1.1 Antecedentes

Por lo general, las grandes empresas tienen una organización bien definida, la cual depende de la función que realizan dentro de la economía y sociedad. Aunque cada una de ellas tiene una constitución muy particular, su organización se basa en un organigrama como el de la figura 1.1

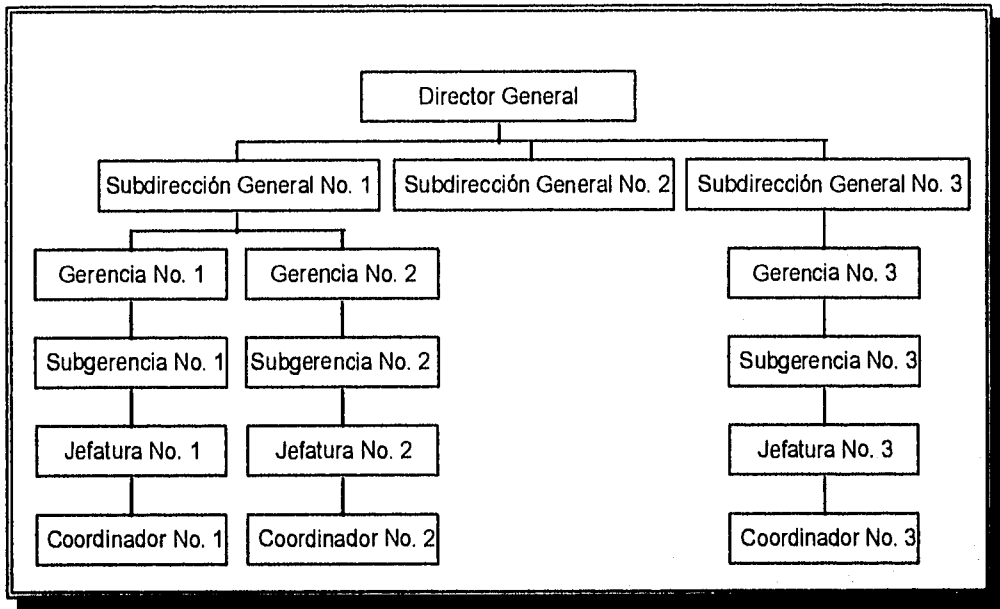


Figura 1.1 Organización general de una empresa.

En cualquier empresa existe la recepción de documentos enviados por otra institución dirigidos al Director General, o bien, a cualquier persona que ocupa otro nivel jerárquico dentro de la misma, cada uno de estos oficios deberán ser notificados a la máxima autoridad por medio de una copia de la gestión.

Algunos asuntos contenidos en estos oficios no necesariamente requieren respuesta y/o solución del Director General, sin embargo, debe estar enterado del asunto y son turnados a aquellas subdirecciones generales, gerencias, subgerencias, coordinaciones, departamentos, etc., que puedan dar respuesta o solución al documento correspondiente.

De cualquier forma, la resolución final deberá ser conocida por la persona que ocupa el máximo nivel jerárquico por medio de una copia del oficio enviado a la institución solicitante.

Según datos recabados en algunas constructoras, aseguradoras, Bancos, Grupos Financieros, Automotrices, Secretarías Gubernamentales tales como la Secretaría de Educación Pública, Secretaría de Gobernación, Secretaría de Comunicaciones y Transportes, Secretaría de Hacienda y Crédito Público,

Secretaría de Comercio y Fomento Industrial, etc., se reciben alrededor de 100 documentos diariamente.

Para llevar a cabo el control de la documentación, es necesario que exista un grupo de personas que realicen las siguientes actividades:

- Recibir los documentos que llegan a la institución; al escrito, se le asigna un número de documento y se sella.
- Clasificarlos de acuerdo a su importancia, tipo de gestión y si son o no confidenciales.
- Turnar el asunto mediante un oficio al personal correspondiente de la empresa. Al tener clasificados los documentos, se advierte quien puede atender el asunto. Posteriormente, de ser necesario, se elabora un nuevo oficio en donde se turna la gestión a la o las personas correspondientes que pueden atenderlo.
- Controlar y recopilar la información generada del seguimiento de la documentación. Se registran los siguiente datos de un documento: tipo, número de documento, clasificación, referencia del oficio, fecha de elaboración, fecha de recepción, fecha de despacho, fecha de descargo, fecha del evento, hora, fecha de archivo, dependencia, remitente, referencia alfabética, tema, expediente, asunto, resolución, turnado e instrucción. Se hace una lista de las personas a quienes se les ha enviado la gestión y la instrucción que deben de realizar, asimismo, de los documentos que el Director General no ha firmado, las invitaciones, audiencias, viajes y cartas enviadas a él, o bien, las que se turnaron y descargaron. Cuando el oficio es turnado a una o varias personas, se debe saber quien no lo ha atendido, quien si y cuantos se han archivado.
- Una vez que los turnados han regresado el oficio contestado, se crea una respuesta y se envía a la institución que originalmente realizó la petición. Cuando se le turna a una sola persona, esta se encarga de contestar directamente a la empresa solicitante y mediante una copia de la gestión le informan al grupo de personas que controla la documentación.
- Archivar los oficios generados. Se asigna un folder en el que se guarda el oficio original y los que se generaron a partir de este. Obviamente cuando al asunto a tratar se le ha dado una respuesta satisfactoria se archiva definitivamente el folder citado.
- Informar y clasificar los documentos de acuerdo al tipo de gestión, cuántos y cuáles se han recibido, turnado, contestado, archivado y enviado. Con base en la información recopilada se elaboran reportes ejecutivos en los que se informa de manera general o detallada de todo lo concerniente a uno o varios documentos.

De acuerdo a lo anterior, la técnica que se utiliza para el seguimiento de la documentación no es complicada, debido a que se realizan las siguientes funciones principales: recibir, analizar, asignar y contestar. Básicamente el gran volumen de gestiones y el control de éstas es lo complejo, ocasionando retraso, descontrol y algunas veces fuga de información. Por lo tanto, se considera necesario tener una herramienta auxiliar, que permita coordinar y controlar el trabajo, información y documentación, para los altos funcionarios de un empresa.

## **1.2 Problemática Actual**

Actualmente en un buen número de empresas todos los procedimientos para el control del flujo de la documentación son totalmente manuales, es decir, no se cuenta con alguna computadora o herramienta que auxilie en la elaboración de oficios en los que se turna el asunto a la o las personas que pueden dar respuesta a la gestión, o sea, se elabora el oficio correspondiente con máquina de escribir y en ocasiones se tienen que repetir por errores humanos.

Aunque existen empresas en donde se auxilian con computadoras y procesadores de texto para establecer un determinado control en algunos procedimientos, la mayoría de ellos son en forma manual. Los datos de los documentos se asientan en formatos, esto es tardado ya que por lo general se necesita mucho espacio para su registro, y además la cantidad de formatos llenados provoca que se pierda control en el seguimiento del oficio.

Adicionalmente se elabora una lista en donde se anota el nombre de la persona a la que se dirige uno o más documentos, el número de documento y la fecha estimada para recibir su respuesta, lo cual requiere también de un gran trabajo porque a veces se envían varios oficios diferentes a una sola persona o bien, un documento a varias personas. En este caso también es fácil perder el control del seguimiento del documento.

Cuando se desea determinar cuales son los oficios que ha atendido el Director General y los que están pendientes, se revisa el registro de los documentos y se obtiene una lista en donde se detallan los datos de éste. Este proceso es sumamente tedioso y requiere de una gran concentración y dedicación puesto que no se debe omitir un solo dato, además de la gran cantidad de tiempo invertido y del personal que se dedica a este trabajo.

Otro caso similar es cuando se quiere saber en forma rápida y oportuna cuando y a quienes se turnaron los documentos recibidos en determinada fecha, cuantos y cuales oficios se deben resolver de acuerdo a un tiempo estimado, así como aquellos documentos que ya se resolvieron y están archivados en forma definitiva.

También con esta información se constituye la agenda de trabajo del Director General, de acuerdo a los viajes programados, conferencias o invitaciones, las cuales también se hacen por escrito. Es difícil poder generar estos reportes ya que este tipo de gestiones no llegan consecutivamente a una empresa.

Otra información requerida y que no se obtiene por falta de tiempo son reportes de cuántos y cuáles oficios tratan un cierto tema o estadísticas de capacidad de respuesta, de volumen de información, etc. que pueden ayudar en la planeación de recursos como espacio en archivo y personal asignado por cantidad de carga de trabajo. Además, la localización de los documentos es muy tardada y deficiente porque no se puede clasificar rápidamente la documentación de acuerdo a la fecha del documento, recepción, despacho, descargo, archivo, dependencia, remitente, turnado, tema, referencia alfabética e instrucción.

Como puede distinguirse, el volumen de información que se maneja diariamente es enorme, por lo que se requiere un gran número de personas para realizar eficientemente estas actividades.

Trabajar en forma manual la mayoría de los procedimientos provoca que los documentos se turnen con cierto retraso o se llegue a perder la continuidad de la documentación y como consecuencia la respuesta no se obtenga con la oportunidad deseada. Esto da mala imagen a la empresa y tomando en cuenta que el tiempo determina el éxito o el fracaso de algunos proyectos se trata de que la mayoría de los asuntos sean resueltos con la mayor rapidez posible.

Una consecuencia significativa es que se genera una problemática que a largo plazo es casi imposible resolver o que ya no tenga caso proseguir porque simplemente no se atendió a tiempo. O bien que la empresa solicitante opte por buscar otro proveedor que cubra sus necesidades, lo cual implica pérdida de clientela, prestigio, credibilidad y por consiguiente ganancias.

### **1.3 La Coordinación Administrativa**

La Coordinación Administrativa es el departamento que pertenece a la Secretaría Particular del C. Secretario de Hacienda y Crédito Público, este departamento se encarga de administrar, controlar y dar seguimiento a la documentación de la máxima autoridad de esta institución. Por lo tanto, la Coordinación Administrativa necesita un sistema que le permita facilitar la administración y el control de los documentos que llegan a esta institución.

Los procedimientos que se requieren son de consulta, captura, generación de reportes, creación de nuevos oficios y control de fechas de despacho, descargo y archivo, las cuales facilitan el seguimiento y administración de la documentación de una empresa.

Los procesos que se desean implementar para llevar a cabo con facilidad el seguimiento de la documentación, son los siguientes:

- **No es factible automatizar la recepción de documentos debido a que es necesario que se le asigne un número consecutivo y una fecha de recibido, además, este se controla conforme van llegando los oficios a la empresa.**
- **Todos los reportes se generan de acuerdo a los datos de la gestión, los cuales proporcionan información que se puede automatizar y así tener un cierto control y administración de la documentación que se genera a través de éstos.**
- **El personal de la Coordinación Administrativa tendrá que capturar todos los datos contenidos en los documentos (fecha del documento, recepción, dependencia, remitente, tema, referencia alfabética, asunto, tipo de gestión, número de documento y clasificación), en caso de ser documento confidencial no se capturarán el asunto sino se digitalizará por razones de seguridad.**
- **Se genera el reporte de los documentos que faltan por turnar, es decir, la impresión de turnos. Este nos ayudará a determinar quién o quienes darán solución a la gestión, creando así, los oficios pertinentes para el seguimiento de la gestión, controlando la fecha de despacho y la instrucción de cada turno.**
- **La o las personas a las cuales se les turnaron los documentos, tendrán la obligación de informar a la Coordinación Administrativa la resolución del asunto por medio de un oficio o la copia de éste. Por lo tanto, se deberá capturar la resolución, la fecha de descargo y se corregirá la instrucción (Descargado).**
- **Es necesario poder validar el tipo de documento, clasificación, dependencia, turnados e instrucciones, esto nos ayudará a que la información no se repita.**
- **Cuando un documento no se ha contestado, la empresa externa envía por segunda ocasión otro oficio como recordatorio de la información o petición que se solicitó anteriormente. El primer documento es antecedente del segundo, este a su vez es subsecuente del primero, es decir, ambos documentos tienen una relación entre sí por el asunto a tratar.**

En estos casos se requiere un generador de oficios subsecuentes, en donde no se transcriba toda la información y la Coordinación Administrativa tome en cuenta si existen oficios antecedentes y subsecuentes. Además se requiere facilitar la consulta de los mismos documentos, para lo cual, es necesario un módulo que permita buscar un documento con su respectivo antecedente y subsecuente, en donde se requieren los siguientes datos: identificador, tipo, número, clasificación, referencia, remitente, dependencia, referencia alfabética y dependencia, tanto del documento consultado como del antecedente y subsecuente. Este tipo de documento ayudará al departamento a controlar los oficios que tengan algún retraso.

La Coordinación Administrativa requiere módulos de consulta, los cuales permitan localizar uno o varios documentos y/o gestiones de acuerdo a diferentes condiciones de búsqueda, tales como: tipo, referencia, hora, dependencia, remitente, referencia alfabética, tema, expediente, turnado, un rango de número de documento, fecha de documento, recepción, despacho, descargo, evento, archivo y por diferentes clasificaciones y turnados; o bien, realizar la consulta únicamente por referencia alfabética, remitente y dependencia proporcionándole una misma condición.

En ambos casos, se requiere observar todos los datos por pantalla, además facilitar la impresión de los siguientes reportes, después de haber realizado la consulta por pantalla.

- **Normal.** Presenta todos los datos de un documento al ser recibido, tales como: identificador, número de documento, clasificación, fecha de documento, recepción, despacho, descargo, evento, archivo, hora, dependencia, remitente, referencia alfabética, tema, expediente, los diferentes turnados con sus respectivas instrucciones, asunto y resolución.
- **Resumido.** Este reporte incluye los siguientes datos: tipo, número de documento, clasificación, fecha de documento, recepción, despacho, dependencia, remitente, los diferentes turnados con sus respectivas instrucciones y el asunto.
- **Completo.** Las impresiones de este reporte deben tener los siguientes datos: Identificador, número de documento, clasificación, fecha de documento, recepción, despacho, dependencia, remitente, los diferentes turnados con sus respectivas instrucciones, el asunto y la resolución.
- **De seguimiento.** Reporte que incluye número de documento, clasificación, fecha de documento, recepción, despacho, dependencia, remitente, los diferentes turnados con sus respectivas instrucciones, el asunto y el encabezado de resolución dejando el espacio en blanco.

Es necesario tener otro módulo de consultas especiales, donde los usuarios proporcionen datos como: tipo de documento, diferentes tipos de clasificaciones, un rango de identificador, diferentes turnados e instrucción, para obtener posteriormente la siguiente información: identificador, número de documento, clasificación, referencia, fecha de documento, recepción, despacho, descargo, evento, archivo, hora, asunto, antecedente, subsecuente y si está o no digitalizado.

Estos módulos permitirán presentar los datos por pantalla o impresos, ya sea por impresora láser o de matriz, debido a que los primeros serán considerados como reportes formales y los segundos como informales o de seguimiento de la documentación.

Todos los datos que sean capturados deben estar presentes en cada uno de los reportes, debido a que estos ayudarán al seguimiento y a la administración de la documentación.

También se necesita un reporte llamado *Impresión de Turnos*, el cual se obtendrá proporcionando un rango de números de documentos y/o distintos turnados, de acuerdo a estas condiciones generales, el reporte tendrá los siguientes datos: número de documento, fecha del actual, referencia, fecha del documento, asunto y el o los otros turnados si existen. Este no servirá de guía para identificar si una gestión será atendida por uno o mas turnados o bien, si ya se turnó el documento.

El reporte de los *Documentos turnados sin resolución*, requiere el o los números de documentos, con lo cual se imprimirán el número y fecha de documento, turnado, instrucción y asunto. Únicamente se considerarán gestiones y el listado estará ordenado por el número del documento.

El reporte de los *Documentos turnados con resolución*, tendrá casi las mismas características del reporte anterior, la diferencia radica en que éste tendrá la dependencia, fecha del documento, referencia, el número de identificador, la resolución y como título *Reporte Verificador*.

En el reporte de invitaciones se desplegará el número de documento, fecha, turnado, instrucción y asunto. Para obtenerlo se proporcionará únicamente el número de documento, ya que el tipo de documento siempre será una gestión y la clasificación será "I+" (mas importante). El título es *Reporte de Invitaciones Recibidas para el Director General*.

El reporte de audiencias tendrá las mismas características que el anterior, en donde se desplegará el título *Reporte de Audiencias Recibidas para el Director*.

El reporte de obsequios presentará los mismos datos que los ya citados, la diferencia estriba en las condiciones para su generación, debido a que traerá los documentos donde el tipo sea igual a R (regalos) y el título será *Reporte de Obsequios (Regalos, Libros, etc.) Recibidos para el C. Director*.

El reporte de discursos necesita los siguientes datos para su generación: un rango de número de identificador, fecha de recepción, evento, diferentes tipos de documentos, turnados, clasificación e instrucción. También debe proporcionar la siguiente información: fecha de discurso, dependencia, remitente, asunto, la clave del turnado, número de identificador y expediente.

Para el programa de viajes es necesario un reporte, el cual requiere un rango de número de identificador, fecha de recepción, evento, distintos tipos de documentos y clasificación. De acuerdo a estos datos deberá presentar la siguiente información: fecha del evento, número de identificador, tipo, número de documento, turnado, expediente, tema, asunto y resolución.

Asimismo, se requiere un reporte de oficios despachados, el cual presentará como información primordial el turnado, instrucción, asunto, número de oficio, identificador, fecha del documento y despacho. Para la generación de este reporte es necesario proporcionar el número de documento, diferentes turnados, clasificación e instrucción.

Los documentos que no fueron turnados, obviamente les dará resolución el Director General, para esto es necesario tener un reporte que maneje la relación de oficios, cartas y diversos documentos para firma. Para generarlo se proporcionarán el rango de números de documentos, recepción, turnados, clasificación e instrucción. El listado contemplará el número de identificador, fecha de recepción, envío, documento, dependencia, asunto y resolución.

En el caso de la relación de los proyectos de decretos para firma del director general, se deberán proporcionar los mismos datos del reporte anterior para que se presenten identificador, fecha de documento, recepción, envío, evento, referencia, turnado, dependencia, asunto y resolución.

El catálogo alfabético es un reporte que permite clasificar la documentación de acuerdo a la dependencia, tema, referencia alfabética, turnado, remitente y expediente. El reporte deberá presentar todos los datos del documento y para poder generarlo se requiere el rango de los números de documentos.

Para tener un control de las invitaciones del Director General, es necesario proporcionar el rango de los números de identificador, fecha de recepción, evento, tipo de documento, turnado, clasificación e instrucción, obteniendo así un listado que presente la fecha de invitación, recepción, expediente, dependencia y asunto.

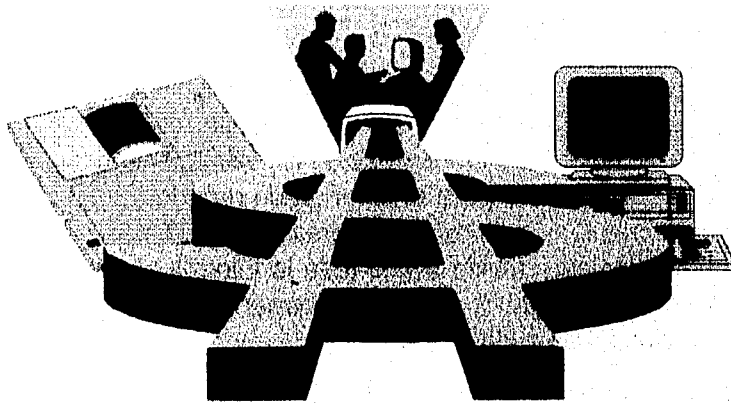
También será útil un reporte estadístico de los documentos pendientes y el número de documentos por área, éste requerirá un rango de los números de identificador, tipo de documento y clasificación.



# CAPÍTULO 2

## AUTOMATIZACIÓN DEL MANEJO DE DOCUMENTOS

---



## **2.1 El Proceso de la Automatización**

El recurso más intensamente utilizado por las grandes compañías es la información. De ahí la necesidad de agilizar el flujo de información de una forma eficiente y rápida.

Actualmente es bastante considerable el volumen de documentos que se reciben diariamente en las grandes empresas o instituciones, los cuales hay que canalizarlos de una manera óptima para que sean asignados a su destinatario final quien dará una respuesta a dicho documento.

Este proceso cuyo objetivo es recibir documentos y asignarlos a la persona adecuada para que atienda y de respuesta al documento, involucra muchos pasos intermedios que por lo general se llevan a cabo manualmente por varias personas o con un cierto grado de automatización, cada persona que participa en el proceso tiene una función específica como: clasificar, ordenar, asignar, distribuir, archivar, etc.

La situación actual de la mayoría de las empresas grandes en cuanto al manejo y administración de los documentos que reciben es la siguiente: se tiene que las empresas reciben un volumen de información de aproximadamente 100 documentos diarios, de los cuales cada uno tiene ciertas prioridades. Parte de los documentos no se encuentran totalmente ordenada ni clasificada, el manejo de todo este volumen de información dentro de las empresas es totalmente manual, el tiempo que se emplea para el manejo de las operaciones es extenso y tedioso.

### **VENTAJAS DE LA AUTOMATIZACIÓN**

En el proceso manual las operaciones se hacen a mano, lo que representa una desventaja y limitaciones, ya que para emitir reportes se debe emplear bastante tiempo, además que la información no siempre se tiene disponible y es de difícil acceso, por ejemplo para obtener algún reporte se necesita que esté el encargado respectivo lo cual implica un retraso. Con la automatización este tipo de problemas se eliminan debido a que toda la información se encuentra dentro del sistema logrando un mayor grado de independencia y no es necesario que estén los responsables de cada etapa.

El proceso manual es poco versátil, ya que es laborioso hacer cambios y corregir errores. Con la automatización los cambios son rápidos y sencillos permitiendo tener la información actualizada. Hay que agregar que la emisión de reportes tiene una presentación moderada, siendo que con la automatización se tiene la posibilidad de emitir reportes en varios formatos o presentaciones y con una mayor calidad.

El manejo de la documentación en forma manual, tiene algunos inconvenientes como las siguientes:

- Es poco ágil.
- Es difícil determinar en que etapa del proceso se encuentra cada documento.
- Los resúmenes se hacen con máquina de escribir.
- La obtención de reportes del estatus de cada documento requiere demasiado tiempo.
- Se depende demasiado del personal que participa en las diferentes etapas del proceso.

La automatización da la posibilidad de encontrar, extraer y utilizar la información existente y determinar en qué parte del proceso se encuentra el documento. Los documentos son almacenados de forma ordenada y clasificada, permitiendo en cada modulo dar altas, bajas, cambios, consultas y reportes en una forma inmediata.

La automatización permite trabajar bajo un ambiente amigable y de fácil manejo, las operaciones se llevan a cabo a través de menús e íconos.

Las personas que intervienen están demasiado ocupadas y no tienen tiempo de realizar cuadros estadísticos, emitir reportes extras o efectuar actividades de retroalimentación como planear el personal asignado por cantidad de carga de trabajo, estimar los tiempos de respuesta, determinar la eficiencia del personal, etc.

La automatización permite llevar a cabo estadísticas, emitir una gran variedad de reportes, liberando al personal de trabajo y permitiéndoles realizar otro tipo de actividades.

Una de las grandes desventajas del manejo manual es que muchas de las personas que intervienen en el proceso, llevan el registro de los documentos mediante una gran cantidad de utensilios de papel y tinta como calendarios, agendas, boletines, notas, recados, etc., y cada uno de ellos lleva un cierto control personal sobre los documentos, por ello estas personas son indispensables, por que si faltara alguna de ellas, la persona que ocupe su lugar no conocería los pasos necesarios para continuar con el proceso del documento. La automatización elimina esta dependencia por que se convierte en un proceso simplificado y sencillo de manejar, además se requiere de pocas personas para llevar acabo todo el proceso.

Como en el proceso manual intervienen las personas, es muy factible que se cometan errores humanos algunos de ellos son:

- Errores al escribir a máquina los oficios necesarios.
- Se olvidan o se pierden documentos.
- Descuido al traspapelar documentos.

Con la automatización se reducen los errores humanos, ya que las personas sólo le dan seguimiento a los documentos a través del sistema y casi no intervienen en el proceso.

Para el manejo del sistema no se requiere de mucho personal ni que este tenga grandes conocimientos en computación para su manejo, debido a que los menús e íconos son fáciles de usar por lo que cualquier persona puede trabajarlo.

Con un sistema automatizado se pueden simplificar las tareas de una forma muy significativa y a corto plazo se reciben grandes beneficios principalmente eficiencia, rapidez, veracidad, confiabilidad, consistencia y presentación.

## **DESVENTAJAS DE LA AUTOMATIZACIÓN**

Así como la automatización tiene grandes ventajas, también hay que mencionar sus desventajas:

Hay que considerar que la implementación de la automatización requiere de tiempo. Hay que entender primero como se lleva a cabo el proceso manual, definir todas sus etapas, características y variantes, así mismo hay que interactuar con el personal para que expliquen a detalle la función que desempeñan, para posteriormente continuar con el desarrollo de la automatización, todo este proceso de conocer el sistema requiere de tiempo por lo que la automatización no se da de manera inmediata.

A pesar de que el proceso se efectúa automáticamente, pueden existir aun errores humanos como los siguientes:

- Se pueden cometer errores al capturar los documentos.
- El sistema no esta exento de errores de programación, los cuales se pueden presentar en cualquier momento, a pesar de que ya existió una etapa de prueba antes de liberar el sistema.
- Se puede perder información por un mal manejo del sistema.
- El personal que lo opera puede borrar accidentalmente información o programas propios del sistema.
- Los respaldos en disco o en cinta pueden dañarse si no se les da un manejo apropiado.

También hay que considerar que se requiere de equipo de cómputo adecuado así como del respectivo software, lo que implica un gasto adicional.

Se requiere personal especializado para darle mantenimiento a la aplicación. La mayoría de las empresas tienen su área especializada de soporte técnico, pero se requerirá periódicamente la asistencia y asesoría de los diseñadores del sistema para que hagan las modificaciones y adecuaciones necesarias.

En resumen las ventajas de la automatización superan considerablemente a las desventajas, es cierto que el costo inicial es elevado, pero a largo plazo se justifica.

Además, en el mundo de los negocios se requiere cada vez mas de disponibilidad y rapidez en la consulta de la información, siendo que las empresas que no automatizan sus procesos se quedan rezagadas, quedando en desventaja por que sus tiempos de respuesta son muy largos y deficientes, lo cual implica que se pierdan clientes o proveedores.

Y tarde o temprano, aunque no lo quieran, estas empresas tendrán que automatizar sus procesos, de lo contrario, la competencia les ganará mercado.

## **2.2 Requerimientos del Ambiente de Implementación**

Los elementos que se necesitan para poder implementar la aplicación son hardware, software y personal que se encargue de operar la aplicación.

La implementación inicial de un sistema es crucial pues la calidad de la instalación afecta su performance (desempeño), mantenimiento, y aún su estabilidad.

Una instalación de alta calidad comprende hardware de una capacidad y velocidad considerable, y software bien diseñado que permita explotar al máximo el hardware.

El software y el hardware apropiado traen como consecuencia óptimo desempeño, velocidad en las transacciones, mantenimiento fácil y máxima flexibilidad para futuros crecimientos.

Una instalación ineficiente puede obstaculizar el aprovechamiento de la inversión en software y originar problemas de desempeño, ocasionando que la aplicación pierda sus beneficios, como alentar el sistema por falta de hardware de alta velocidad, o problemas por espacio en memoria o disco, lo que implicaría que se le de demasiado mantenimiento al sistema o que la aplicación no cumpla con los objetivos que se plantearon, es decir que en lugar de facilitar el trabajo, la automatización se convierta en una carga para el personal que lo opera.

La figura 2.1 muestra como se integran los componentes para conformar todo el sistema:

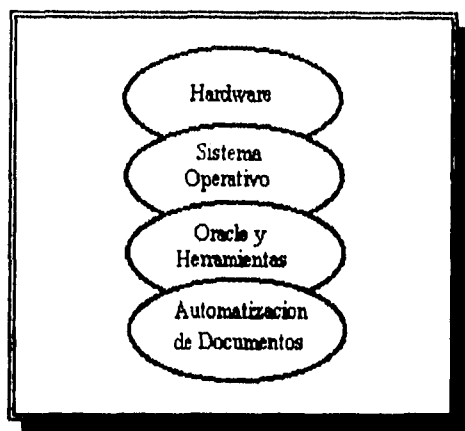


Figura 2.1 Componentes Generales de la Implementación.

La siguiente sección especifica el hardware y software que se necesita para poder utilizar la aplicación de automatización del manejo de documentos. El sistema puede correr en una configuración monousuario o en red.

## **REQUERIMIENTOS PARA LA CONFIGURACIÓN MONOUSUARIO**

Para la configuración monousuario se requiere como mínimo de una computadora 486 con 500 MB en disco duro y 16 MB en RAM.

Los 16MB (o más) son indispensables para que el sistema soporte la aplicación. Además se requiere un monitor de alta resolución, para que los usuarios puedan trabajar en un ambiente gráfico.

Es conveniente tener dos impresoras, una láser y otra matricial. La láser se ocupa para la presentación final de los reportes y la matricial se ocupa para borradores, o bien para emitir reportes que no requieran mucha calidad en la impresión.

El software requerido es el Sistema Operativo (MS-DOS y Windows), las herramientas y el administrador de base de datos (Developer 2000 y Oracle 7) y la aplicación SICADE. La configuración monousuario se muestra en la figura 2.2.

## **HARDWARE**

- Impresora láser o matricial (opcional)
- Computadora Personal 486
- Con drive de 3.5 pulgadas
- 500 MB en disco duro
- Monitor super VGA
- 16 MB en RAM
- A 66 MHz

## SOFTWARE

- MS-DOS versión 6.0 o mayor
- Microsoft Windows 3.1
- Developer 2000 (herramientas Oracle)
- ProC
- Personal Oracle 7
- Sistema de Automatización de Documentos.

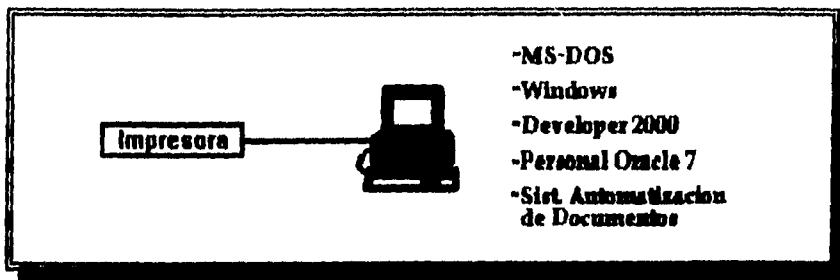


Figura 2.2 Configuración Monousuario

## REQUERIMIENTOS PARA LA CONFIGURACIÓN EN RED

Para la configuración en red, se considera una Red de Área Local (LAN). En los "Clientes" se requiere hardware y software que permita un ambiente gráfico, además una tarjeta para red y software para red.

En el Servidor se requiere el protocolo de comunicaciones Oracle Net, Oracle 7 Enterprise (que permite crear y correr aplicaciones en una arquitectura Cliente-Servidor) y el SICADE.

También se requieren dos impresoras, matricial para borradores y láser para reportes de alta calidad. En la figura 2.3 se muestra la configuración en red.

## HARDWARE

- Se requiere una arquitectura cliente-servidor
- Tarjetas de red.

## SOFTWARE

- MS-DOS versión 6.0 o mayor para cada computadora
- Microsoft Windows 3.1
- Developer 2000 herramientas Oracle
- ProC
- Personal Oracle 7
- Personal Oracle 7 Enterprise (especial para Red)
- Protocolo de comunicaciones SqlNet para Windows

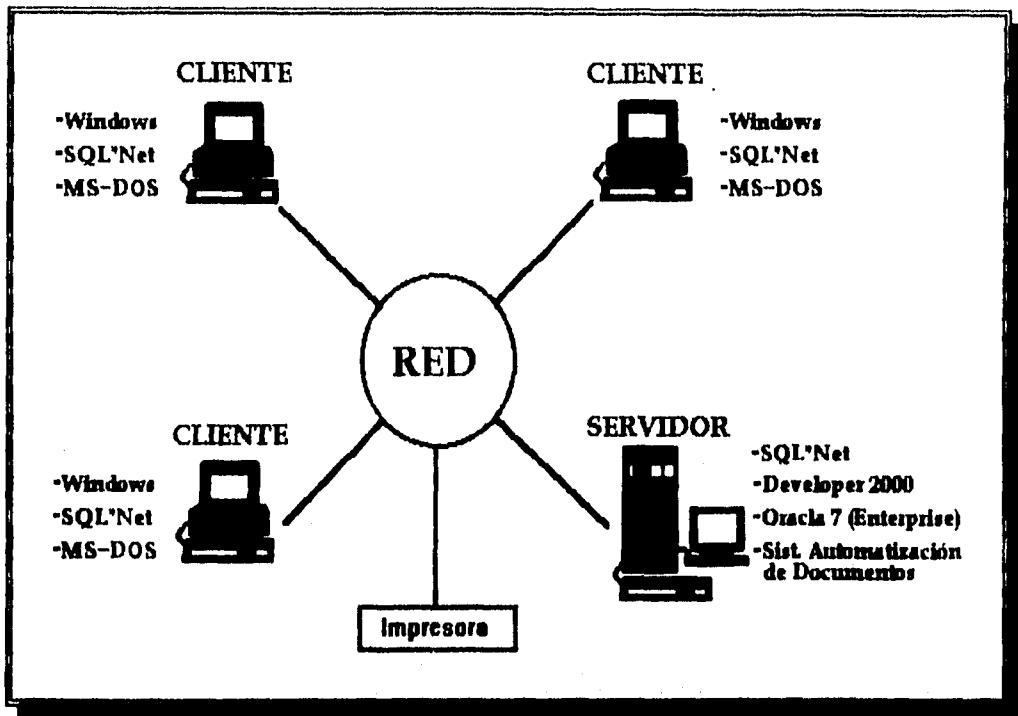


Figura 2.3 Configuración en Red

### **VENTAJAS Y DESVENTAJAS DE LA APLICACIÓN FUNCIONANDO EN MODO MONOUSUARIO**

#### **Desventajas:**

- No se pueden compartir recursos.
- El potencial de la aplicación se reduce porque queda centralizada.
- Alto costo.

#### **Ventajas:**

- El mantenimiento al sistema se reduce.
- No se requiere demasiado personal.

### **VENTAJAS Y DESVENTAJAS DE LA APLICACIÓN FUNCIONANDO EN RED**

#### **Ventajas:**

- Múltiples usuarios pueden utilizar la aplicación
- Se comparten los recursos: impresora, datos, procesamiento, etc.



**Desventajas:**

- Se requiere mayor inversión en equipo.
- Como la aplicación es en modo gráfico las terminales deben tener el software y el hardware necesario para soportar la aplicación.

**2.3 Costos**

Los costos de hardware y software varían de precio de un proveedor a otro, pero en general la diferencia de precio no es muy considerable. El precio del SICADE no se ha estimado porque no se a pensado en su comercialización, lo cual requeriría un estudio detallado de mercado.

**SOFTWARE**

• Developer 2000	5995 DII
• ProC	600 DII
• Personal Oracle 7	495 DII
• Per. Oracle Enterprise Red	745 DII
• SqlPlus	600 DII
• Runtime	60 DII
Sistema de Automatización de Documentos	-----

**HARDWARE**

	<b>Costo TOTAL</b>
1) PC con configuración mínima (incluye software):	1500 DII
• Computadora Personal 486	
• Con drive de 3.5 pulgadas	
• 500 MB en disco duro	
• Monitor super VGA	
• 16 MB en RAM	
• A 66 MHz	
• MS-DOS versión 6.0 o mayor	
• Microsoft Windows 3.1	
2) Impresoras:	
• Impresora Matricial	500 DII
• Impresora Láser	2500 DII
3) Scanner (Para documentos confidenciales)	2000 DII

## **2.4 Características Generales del Nuevo Sistema**

En los negocios, organizaciones y en las actividades de todos los días se automatizan más funciones críticas y se confía más en la automatización. Esta realidad es una carga creciente para los analistas de sistemas al asegurar que los sistemas que desarrollen sean de una calidad adecuada y que cumplan con el propósito para el cual fueron creados.

La importancia de un sistema depende de los alcances y del grado de automatización que se logre, cierto es que la automatización es una gran ventaja, pero no se logra al cien por ciento, ya que siempre será necesario que intervenga la mano del hombre.

Por ello es conveniente definir las etapas que se van a automatizar y en que grado van a intervenir las personas encargadas.

De los inconvenientes del proceso manual surge la necesidad de automatizar el manejo de documentos, que permita entre otras cosas:

- Reducir el tiempo de todo el proceso.
- Obtener reportes rápidos y de buena calidad.
- Reducir el número de personas que intervienen asignándolas a otras áreas o facilitando su trabajo.
- Tener un sistema confiable.
- Tener un historial de todos los documentos.
- Tener niveles de seguridad para los documentos importantes o personales.
- Darle mayor seguridad a toda la información.
- Permitir realizar estadísticas.

# CAPÍTULO 3

TEORÍA DEL ANÁLISIS Y DISEÑO ESTRUCTURADO  
MODERNO DE SISTEMAS

---



### **3.1 Utilización de modelos**

El objetivo principal del *análisis estructurado* es transformar sus dos entradas principales, las políticas del usuario y el esquema del proyecto, en una especificación estructurada. Esto implica modelar el ambiente del usuario con diagramas de flujo de datos, diagrama de entidad-relación y diagramas de transición de estado.

El proceso del análisis, en sus etapas primarias, consiste en desarrollar un modelo ambiental y un modelo de comportamiento para conformar el modelo esencial que representa una descripción formal de lo que el nuevo sistema debe hacer, independientemente de la naturaleza de la tecnología que se use para cubrir los requerimientos.

Dichos modelos se construyen por tres razones:

- Para resaltar las características importantes del sistema y minimizar aquellas menos importantes.
- Para discutir cambios y correcciones a los requerimientos del usuario, a bajo costo y con riesgo mínimo.
- Para verificar que se entiende el ambiente del usuario, y que se ha documentado de tal manera que los diseñadores y programadores puedan construir el sistema.

Lograremos esto al utilizar diagramas gráficos, ya que con ellos se identifican fácilmente los componentes de un sistema y su interfaz. Todos los demás detalles se presentan en documentos textuales de apoyo tales como lo que se conoce como "*especificación del proceso*" y el "*diccionario de datos*".

Asimismo los diagramas gráficos ofrecen y cumplen con una característica importantísima: La capacidad de mostrar un sistema por partes en forma descendente. Esto quizás no sea importante para sistemas pequeños, pues de ellos se puede decir todo lo necesario en unas pocas páginas, y cualquiera que necesite conocer bien algún aspecto del sistema puede conocerlo en su totalidad.

Sin embargo, algunos sistemas como el que nos ocupa, deben subdividirse para un profundo análisis y seguimiento; es como presentar primero el mapa de la República Mexicana, después el de un Estado o más, posteriormente el de un Municipio, etc.

### **3.2 El diagrama de flujo de datos (DFD)**

La herramienta gráfica en la que nos vamos a apoyar es el *Diagrama de Flujo de Datos*, el cuál permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por "conductos" y "tanques de almacenamiento" de datos.

Otras formas de nombrar al diagrama de flujo de datos son las siguientes:

- DFD (Abreviatura que se usará en todo este libro).
- Carta de burbujas.
- Diagrama de burbujas.
- Modelo de proceso.
- Diagrama de flujo de trabajo.
- Modelo de función.

El DFD es una de las herramientas más comúnmente usadas, sobre todo por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja.

Los DFD se utilizaron por primera vez en la ingeniería de software como notación para el estudio del diseño de sistemas. A su vez, la notación se tomó prestada de artículos anteriores sobre teoría de gráficas, y continúa siendo utilizada por los ingenieros de software que trabajan en la implantación directa de modelos de los requerimientos del usuario.

La gran ventaja que presenta el DFD es que prácticamente no requiere explicación; se puede simplemente mirar el diagrama y entenderlo. La notación es sencilla y clara y, en cierto sentido, intuitivamente obvia. Esto es particularmente importante cuando recordamos que quien lo observará no es el analista, sino el usuario.

Los componentes de un DFD son el *proceso*, el *flujo*, el *archivo* y la *entidad*.

El *proceso* es el primer componente del DFD. Los sinónimos comunes son *burbuja*, *función* o *transformación*. Se presenta gráficamente como un círculo o un rectángulo con las esquinas redondeadas, como se muestra en la figura 3.1.

Tiene como objetivo mostrar una parte del sistema que transforma entradas en salidas; es decir, permite observar cómo es que una o más entradas se transforman en salidas. Se debe destacar que el proceso se nombra o describe con una sola palabra, frase u oración sencilla. En casi todos los DFD el nombre del proceso denota *lo que hace*.



Figura 3.1 Representación de un PROCESO.

El siguiente componente de un DFD es el *flujo*. Se usa para describir el movimiento de bloques o paquetes de información o sea, un conjunto de datos, de una parte del sistema a otra.

Un *flujo* se representa gráficamente por medio de una flecha que entra o sale de un proceso, tal como se ve en la figura 3.2.

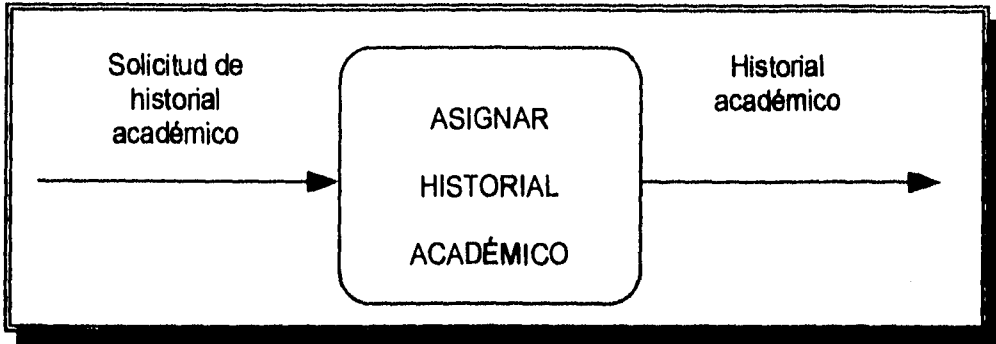


Figura 3.2 Flujo de entrada y flujo de salida en un proceso.

Por ello, los flujos representan datos en movimiento, mientras que los archivos representan datos en reposo.

El flujo de la figura 3.2 tiene un nombre, el cual representa el significado del paquete, o sea, un conjunto de datos, que se mueve a lo largo del flujo. Esto quiere decir que el flujo sólo lleva un tipo de paquete, como lo indica su nombre.

Es importante considerar que a veces es útil consolidar varios flujos elementales en uno solo, ya que es mejor utilizar un flujo llamado *MUEBLES* en lugar de diversos flujos llamados *SILLA*, *SILLÓN* y *MESA*. Aunque parezca obvio, hay que tener en mente que el mismo contenido pudiera tener distintos significados en distintas partes del sistema.

Por ejemplo el número 8724626-3 tiene distinto significado cuando viaja a lo largo de un flujo llamado *NUMERO DE CUENTA* que cuando viaja a lo largo de uno llamado *NUMERO DE CUENTA VALIDO*.

En el primer caso, significa un número de cuenta que pudiera ser o no válido; en el segundo caso, significa un número de cuenta que, dentro del contexto del sistema en cuestión, se sabe que es válido.

También hay que notar que las flechas muestran la dirección del flujo, una cabeza de flecha en cualquier extremo (o posiblemente ambos) del flujo indica si los datos se están moviendo hacia adentro o hacia afuera de un proceso (o ambas cosas).

Los flujos de datos pueden divergir o converger en un DFD; conceptualmente esto es algo así como un río principal que se divide en varios más pequeños, o varios pequeños que se unen. Sin embargo, esto tiene un significado especial en un DFD en el cual hay paquetes de datos que se mueven a través del sistema.

En el caso de un flujo divergente, esto significa que se están mandando copias por duplicado de un paquete de datos a diferentes partes del sistema, o bien que un paquete complejo de datos se está dividiendo en varios paquetes individuales más, cada uno de los cuales se está mandando a diferentes partes del sistema. De manera inversa, en el caso de un flujo convergente, significa que varios paquetes elementales de datos se están uniendo para formar agregados más complejos de paquetes de datos.

Otro componente de un DFD es el *archivo*, el cuál se utiliza para modelar una colección de paquetes de datos en reposo.

En la figura 3.3 se presenta la notación gráfica del archivo.

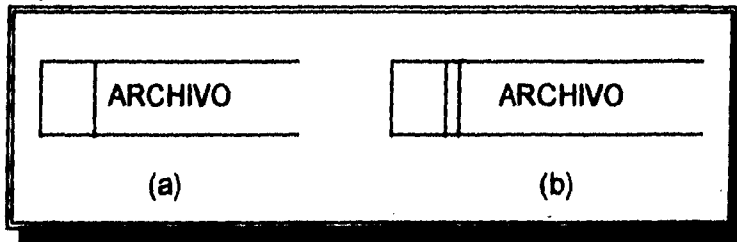


Figura 3.3 Representación gráfica del archivo. (a) Notación para dibujarse una sola vez en una hoja. (b) Notación cuando se dibujan dos o más veces en una sola hoja.

De modo característico, el nombre que se utiliza para identificar al archivo es el plural del que se utiliza para los paquetes que entran y salen del archivo por medio de flujos.

Los archivos se conectan por flujos a los procesos. Así, el contexto en el que se muestra un archivo en un DFD es uno de los siguientes:

- Un flujo desde un archivo.
- Un flujo hacia un archivo.

Normalmente se interpreta un flujo que procede de un sistema como una lectura o un acceso a la información del archivo.

Esto significa específicamente que:

- Se recupera del archivo un solo paquete de datos; esto es, de hecho, el ejemplo más común del flujo desde un archivo.
- Se ha recuperado más de un paquete del archivo.
- Se tiene una porción de un paquete del archivo.
- Se tienen porciones de más de un paquete del archivo.

Cuando se examinan los flujos que entran y salen de un archivo surgirán muchas preguntas de tipo procedimiento: ¿representa el flujo un solo paquete, muchos, porciones de uno o porciones de diversos paquetes?. En algunos casos, podemos darnos cuenta simplemente viendo la etiqueta del flujo.

Si el flujo no está etiquetado, significa que todo el paquete de información se está recuperando; si la etiqueta del flujo es la misma que la del archivo significa que se recupera todo un paquete (o múltiples instancias de uno completo); si la etiqueta del flujo es diferente del nombre del archivo, entonces se están recuperando uno o más componentes de uno o más paquetes.

Existe un detalle de tipo procedimiento del cual podemos estar seguros, el archivo es pasivo, y los datos no viajarán a lo largo del flujo a menos que el proceso lo solicite explícitamente.

Existe otro detalle de tipo procedimiento que consideran, por convenio, los sistemas de proceso de datos, el archivo no cambia cuando un paquete se mueve del archivo a lo largo del flujo.

Un flujo hacia un archivo habitualmente se describe como una escritura, una actualización o posiblemente una eliminación. Específicamente, sólo puede significar que se tiene una de las situaciones siguientes:

- Se están guardando uno o más paquetes nuevos en el archivo. Dependiendo de la naturaleza del sistema, los paquetes nuevos pudieran anexarse (es decir, de alguna manera acomodarse para que estén "después" de los paquetes existentes); o pudieran colocarse en algún lado entre los paquetes existentes.

Esto es a menudo un asunto de la implantación (es decir, controlado por el sistema específico de administración de bases de datos).

- Uno o más paquetes se están borrando o retirando del archivo.
- Uno o más paquetes se están modificando o cambiando. Esto pudiera traer consigo un cambio de todo un paquete, o (más comúnmente), de sólo una porción, o de una porción de múltiples paquetes.

En todos estos casos es evidente que el archivo cambia como resultado del flujo que ingresa. El proceso (o procesos) conectados con el otro extremo del flujo es el responsable de realizar el cambio al archivo.

Finalmente, el último componente del DFD es una *entidad externa* o *terminador*. Gráficamente se representa como un rectángulo, como se muestra en la figura 3.4.

Los terminadores representan entidades externas con las cuales el sistema se comunica.

Comúnmente, un terminador es una persona o un grupo, por ejemplo, una organización externa o una agencia gubernamental, o un grupo o departamento que esté dentro de la misma compañía u organización, pero fuera del control del sistema que se está modelando.

En algunos casos, un terminador puede ser otro sistema.



Suele ser muy fácil identificar los terminadores en el sistema que se está modelando. A veces el terminador es el usuario; es decir, el usuario suministra datos al sistema o bien los recibe ya procesados.

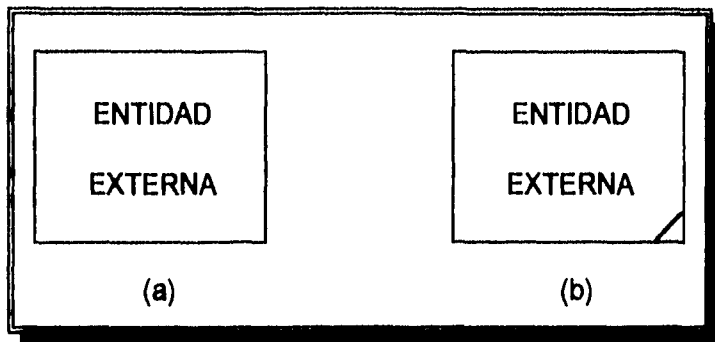


Figura 3.4 Representación gráfica de la entidad externa. (a) Notación para dibujarse una sola vez en una hoja. (b) Notación cuando se dibujan dos o más veces en una hoja.

En otros casos, el usuario se considera parte del sistema y ayudará a identificar los terminadores relevantes, es decir, de los que reciben información o bien a los que se la entrega ya procesada.

Existen tres factores importantes que hay que tomar en cuenta en relación a las entidades:

- Son externos al sistema que se está modelando; los flujos que conectan los terminadores a diversos procesos (o archivos) en el sistema representan la interfaz entre él y el mundo externo.
- Como consecuencia, es evidente que ni el analista ni el diseñador del sistema están en posibilidad de cambiar los contenidos de un terminador o la manera en que trabaja.
- Las relaciones que existan entre los terminadores no se muestran en el modelo de DFD. Si existen relaciones entre los terminadores y si es esencial modelarlos para poder documentar los requerimientos del sistema, entonces, por definición, los terminadores son en realidad parte del sistema y debieran modelarse como procesos.

### **3.3 Guía para la construcción de un DFD**

Existen reglas adicionales que se requieren para poder utilizar los DFD con éxito. Algunas de estas reglas ayudarán para no elaborar DFD erróneos, incompletos o lógicamente inconsistentes.

Tienen como finalidad auxiliar en el dibujo de un DFD grato a la vista, y que, por tanto, tenga más probabilidades de que el usuario lo lea con cuidado.

1. Identificar las entradas y salidas netas.
  - Determinar el contexto del sistema, esto es, las entidades externas e internas al sistema.
  - Buscar los flujos de datos que cruzan el contexto, estos constituirán las entradas y salidas netas.
  - El contexto debe ser suficientemente grande como para incluir todo lo relevante al esfuerzo de desarrollo, pero también lo suficientemente pequeño para no incluir cuestiones irrelevantes.
  
2. Llenar el cuerpo del DFD.
  - Concentrarse en los flujos de datos.
  - Averiguar si dentro de los procesos identificados hay flujos de datos, en tal caso separar ese proceso en dos, tres o en los procesos que sean necesarios.
  - Cuestionarse en cada flujo de datos:
    - ¿Qué se necesita para construirlo?
    - ¿De dónde provienen sus componentes?
    - ¿Qué procesos se requieren para ello?
    - ¿Podemos construir este flujo de datos a partir de los flujos de datos de entrada?
  
3. Introducir en el DFD los archivos que representen los depósitos de información que el usuario ha identificado.
  
4. Revisar si no hay flujos de datos faltantes o sobrantes o redes desconectadas.
  
5. Nombrar los flujos de datos.
  - Asignar nombres a todos los flujos de datos.
  - Los flujos que entran o salen de un archivo no requieren nombre en la mayoría de los casos.
  - Evitar nombres ambiguos.
  - No usar homónimos o sinónimos al nombrar un flujo de datos.
  - El nombre del flujo de datos no únicamente debe representar los datos sino también lo que sabemos acerca de ellos.
  - Tener cuidado de no mezclar datos que nada tienen que ver entre sí.
  - Si no se encuentra un nombre preciso, dividir el flujo de datos.

## 6. Nombrar los procesos.

- Asegurarse de que todos los flujos de datos ya tienen nombre.
- Tratar de nombrar los procesos usando un verbo y un objeto directo.
- Un proceso en un DFD puede representar una función que se está llevando a cabo, o pudiera indicar cómo se está llevando a cabo, identificando a la persona, grupo o mecanismo involucrado.

En el último caso, obviamente es importante etiquetar con precisión el proceso, de modo que quienes leen el DFD, especialmente los usuarios, puedan confirmar que se trata de un modelo preciso. Sin embargo, si el proceso lo hace una sola persona, se recomienda identificar el papel que dicha persona está representando, más que su nombre o identidad.

- Los nombres de los procesos deben hacerse en términos de sus entradas y salidas.
- Evitar ambigüedades o nombres que abarquen parcialmente el proceso.
- Si se obtienen nombres ambiguos reparticionar el proceso.
- No utilizar verbos muy generales como "Procesar", "Hacer"; ni objetos directos muy generales como "Información", "Datos", etc.

## 7. Asignar un número a los procesos.

- Como una forma conveniente de referirse a los procesos en un DFD se asigna un número a cada burbuja. No importa mucho cómo se haga esto, de izquierda a derecha, de arriba a abajo o de cualquier otra manera servirá, mientras haya consistencia en la forma de aplicar los números. La única situación que hay que tener en cuenta es que el sistema de numeración implicará, para algunos lectores casuales de un DFD, una cierta secuencia de ejecución.

Esto no es así en absoluto. El modelo de DFD es una red de procesos asíncronos que se intercomunican, lo cual es, de hecho, una representación precisa de la manera en la que en realidad muchos sistemas operan.

- La utilidad de la numeración se aprecia también cuando un proceso necesita dividirse en otros procesos más elementales -llamados "hijos"-, ya que éstos heredan el número del proceso "padre". Por ejemplo, a un proceso X se le asigna el número 2, otros procesos derivados de éste tendrán los números 2.1, 2.2, 2.3, etc. Los números de los procesos derivados del 2.1, serán 2.1.1, 2.1.2, etc.

8. Omitir detalles de manejos de errores.
  - Cuando se detecten rutas de error se recomienda dejarlas indicadas y trabajar en ellas posteriormente.
  - Si el error no requiere descartar algún proceso ya efectuado, ignorarlo.
  - Si el error requiere deshacer algo que ya se hizo entonces no ignorarlo.
  
9. Mostrar flujo de datos y no de control.
  - Verificar que en todos los flujos de datos se esté pasando información, de otra manera, eliminarlos.
  - Verificar que la información en cada flujo de datos esté usándose en el proceso al que llega, de no ser así, eliminarla.
  
10. Revisión del DFD.
  - Es imposible que el DFD esté correcto la primera vez.
  - Si el primer DFD dibujado resulta ser el final, seguramente tiene errores.
  - Ejecutar todo este proceso de manera iterativa.

Esto pudiera parecer mucho trabajo, pero bien vale la pena el esfuerzo de desarrollar un modelo preciso, congruente y agradable, de los requerimientos del sistema.
  - Estar preparado para sustituir el primer DFD por otras versiones mejoradas.
  - Este proceso iterativo dará grandes beneficios en las fases posteriores del ciclo de vida del sistema.

Ahora bien, existen otras reglas respecto a cómo asegurar que el DFD mismo sea consistente, estas son:

- Evitar procesos infinitos, es decir, aquellos que tienen entradas pero no salidas. Estos procesos también son conocidos como agujeros negros.
- Evitar los procesos de generación espontánea, que tienen salidas sin tener entradas, por que son sumamente sospechosas y generalmente incorrectas.
- Estudiar profundamente los flujos y procesos sin nombre, ya que esto indica que puede ser dividido en datos o procesos más elementales.
- Analizar cuidadosamente los archivos de "sólo lectura" o "sólo escritura".

Un archivo típico debiera tener tanto entradas como salidas. La única excepción a esta regla es algún archivo externo, que sirve de interfaz entre el sistema y alguna entidad externa.

Cuando el sistema a modelar es intrínsecamente complejo y tiene docenas o incluso cientos de procedimientos que modelar, se recurre a la construcción de un DFD global en una serie de niveles de modo que cada uno proporcione sucesivamente más detalles sobre una porción del nivel anterior.

El DFD del primer nivel consta sólo de un proceso, que representa el sistema completo; los flujos de datos muestran las interfaces entre el sistema y las entidades externas junto con los archivos externos que pudiera haber. Este DFD especial se conoce como Diagrama de Contexto.

El DFD que sigue del diagrama de contexto se conoce como el DFD 0. Representa la vista de más alto nivel de las principales funciones del sistema, al igual que sus principales interfaces.

Algunos otros aspectos que deben considerarse adicionalmente son:

- Saber cuántos niveles debe haber en un DFD. Cada DFD debe tener no más de media docena de burbujas y archivos relacionados. Así, si se ha partido un sistema grande en tres niveles, pero sus DFD de nivel más bajo aún contienen 50 procesos cada uno, entonces falta por lo menos un nivel más.

Si no podemos escribir una especificación razonable para un proceso en alrededor de una página, entonces es probable que sea demasiado compleja y debiera partirse en DFD de menor nivel antes de tratar de escribir la especificación.

- No existen reglas acerca del número de niveles en el análisis de un sistema típico. En un sistema simple, probablemente se encontrarán dos o tres niveles; uno mediano tendrá generalmente de tres a seis niveles; uno grande tendrá de cinco a ocho niveles. El número total de procesos se incrementará exponencialmente a medida que se baja de nivel al inmediato inferior.
- No es necesario dividir todas las partes del sistema con el mismo nivel de detalle. Algunas partes del sistema pueden ser más complejas que otras y pueden requerir uno o más niveles de partición.
- Mostrar los niveles al usuario. Muchos usuarios sólo querrán ver un diagrama: un usuario ejecutivo de alto nivel pudiera querer ver tan sólo el diagrama de contexto o tal vez el diagrama de nivel 0; un usuario de nivel operacional pudiera querer ver tan sólo la figura que describe el área en la cual está interesado. Pero si alguien quiere ver una parte más extensa del sistema, o tal vez todo, entonces tiene sentido presentar los diagramas de una manera descendente: comenzar con el diagrama de contexto y continuar hasta niveles más bajos de detalle.

- Asegurar que los niveles del DFD sean consistentes entre sí. Para asegurar que cada figura sea consistente con su figura de más alto nivel se sigue una regla sencilla: los flujos de datos que salen y entran de un proceso en un nivel dado deben corresponder con los que entran y salen de toda la figura en el nivel inmediato inferior que la describe.
- Mostrar los archivos en los diversos niveles. Esta es una área donde la redundancia se introduce deliberadamente en el modelo. La regla es la siguiente: mostrar un archivo en el nivel más alto donde primeramente sirve de interfaz entre dos o más procesos; luego, mostrarlo de nuevo en cada diagrama de nivel inferior que describa más a fondo dichos procesos de interface. Otra consideración importante es que los archivos locales, que utilizan sólo los procesos en una figura de menor nivel, no se mostrarán en niveles superiores, dado que se incluyen de manera implícita en un proceso del nivel inmediato superior.
- Realizar la partición de los DFD en niveles. A pesar de que los DFD deben presentarse al público usuario de una manera descendente, no es necesariamente cierto que se deban desarrollar así.

El enfoque descendente intuitivamente es muy atractivo, sin embargo, un enfoque que tiene más éxito es identificar primero los acontecimientos externos a los cuales debe responder el sistema y utilizarlos para crear un primer borrador del DFD.

### **3.4 Modelo Conceptual de Datos**

El modelo conceptual de datos es el primer paso del proceso TOP-DOWN para el desarrollo de una base de datos, se realiza en las etapas de Análisis y Diseño del Ciclo de Desarrollo de un Sistema.

Representa una forma efectiva para integrar y documentar los requerimientos de información de una organización, así como ser independiente del hardware o del software usados para la implementación

El objetivo del Modelo Conceptual de Datos es desarrollar el Modelo Entidad-Relacion (E-R), el cual puede ser utilizado para una base de datos de red, jerárquica o relacional. Los componentes del modelo (E-R) son las entidades y las relaciones.

Las *entidades* representan a las personas, instituciones, grupos financieros, escuelas, etc, de las cuales se necesita conocer información y a las que se les pueden asignar atributos.

Para dibujar una entidad se utilizan cuadrados o rectángulos de cualquier dimensión con las esquinas redondeadas.

El nombre de la entidad se escribe en mayúsculas y en singular. En ocasiones también se escribe un nombre alternativo o sinónimo, el cual se anota abajo del principal y entre paréntesis.

Finalmente, el nombre de los atributos en minúsculas como se muestra en la figura 3.5.



Figura 3.5. Ejemplo de cómo dibujar una entidad.

Se debe estar seguro de que la entidad posea múltiples "ocurrencias", "instancias" o "registros", considerando que cada instancia de la entidad tiene valores específicos para cada atributo de la entidad.

Cada instancia debe ser identificada como única de otras instancias de la misma entidad. Un atributo o conjunto de atributos que identifican de manera única a una instancia dentro de una entidad son llamados *Identificadores Únicos (UID)*.

Si una entidad no puede tener un identificador único (UID), ésta no puede ser una entidad.

Los atributos que identifican de manera única una entidad y pertenecen a los UID de las entidades son precedidos por un símbolo de número y asterisco ( # \* ).

El otro componente de un modelo (E-R) es la *relación*, la cual se considera como un evento bidireccional que representa la asociación entre dos entidades, o entre una entidad consigo misma.

La forma de escribir una relación debe contemplar el formato que se presenta en la figura 3.6.

Por ejemplo, si tuviéramos las entidades PROFESOR y MATERIA, su relación estaría dada como sigue:

- Cada materia puede ser enseñada por uno y solamente un profesor.
- Cada profesor puede ser asignado a una o más materias.

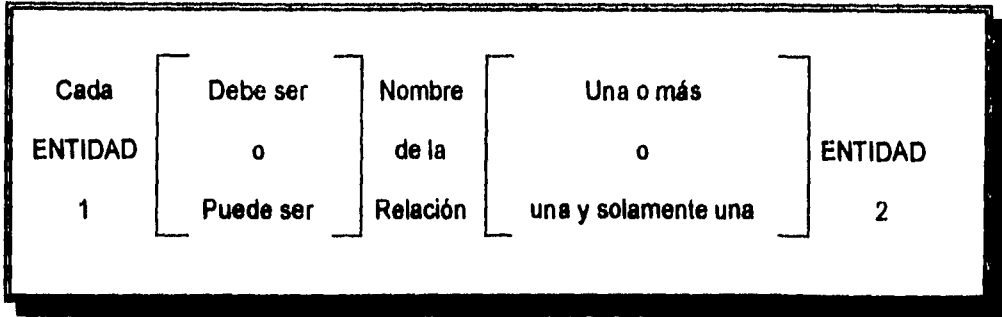


Figura 3.6 Formato para escribir la relación entre dos entidades.

Es importante destacar que cada dirección de una relación tiene:

- Un nombre. Por ejemplo: Enseñado *por* o asignado *a*.
- Una opción. Por ejemplo: *Debe ser* o *puede ser*.
- Un grado o cardinalidad. Por ejemplo: *Uno y solamente uno*, o *uno o más*.

Un grado de 0 es etiquetado como *puede ser*.

Para dibujar la relación se necesita una línea entre dos entidades y los nombres de las relaciones escritas en minúsculas.

La opcionalidad queda esquematizada en la figura 3.7

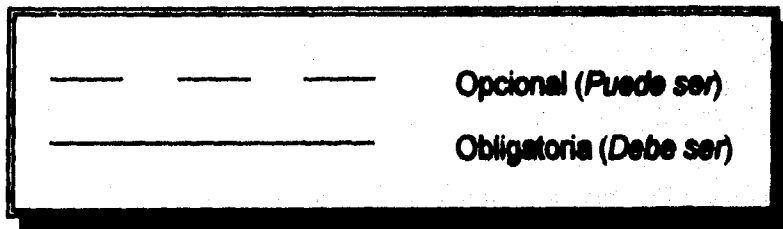


Figura 3.7 Opcionalidad

El Grado queda ilustrado en la figura 3.8

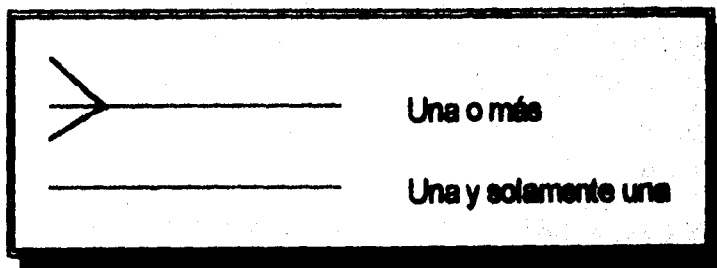


Figura 3.8 El grado representado por una "pata de gallo" y una línea simple.



En resumen, lo anterior se ejemplifica en la figura 3.9:

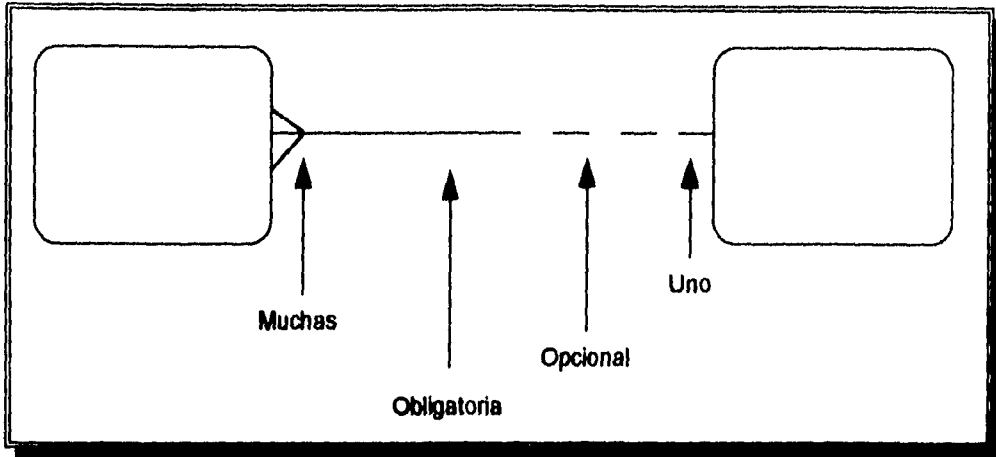


Figura 3.9 Representación gráfica de una relación entre entidades

Existen tres grados de relación:

- Relaciones muchos a uno.
- Relaciones muchos a muchos.
- Relaciones uno a uno.

Una relación uno a uno (1 a 1 ó 1:1) tiene un grado de *uno y solamente uno* en ambas direcciones.

Una relación muchos a uno (M a 1 ó M:1) tiene el grado de *uno o más* en una dirección y el grado de *uno y solamente uno* en la otra dirección.

Una relación muchos a muchos (M a M o M:M) tiene el grado de *uno o más* en ambas direcciones. Usualmente son opcionales en ambas direcciones, aunque pueden ser opcionales también en una sola dirección.

También en este momento se usa una **Matriz de Relaciones** como un apoyo para la recolección inicial de información sobre las relaciones entre una serie de entidades.

Los estándares de una **Matriz de Relaciones** son:

- Esta matriz muestra si están relacionadas y en qué forma cada entidad ( renglón) con cada entidad (columna) mostrada en la matriz.
- Todas las entidades están listadas en el lado izquierdo y en la parte superior de la matriz.
- Si una entidad está relacionada con otra entidad, entonces el nombre de esta relación se muestra en la celda de intersección.
- Si una entidad no está relacionada con otra entidad, entonces se dibuja una línea en la celda de intersección.

- Cada relación por encima de la diagonal es el inverso o imagen espejo de la relación por debajo de la línea diagonal.
- Las relaciones recursivas (o sea, una entidad relacionada consigo misma) son representadas por las celdas en la diagonal.

En la figura 3.10 se representa una matriz de relaciones.

	Entidad 1	Entidad 2	...	Entidad n
Entidad 1	-----	Relación 1-2	...	Relación 1-n
Entidad 2	Relación 2-1	-----	...	-----
...	...	...	...	...
Entidad n	Relación n-1	-----	...	-----

Figura 3.10 Matriz de relaciones.

### **3.5 Representación del Diagrama Entidad Relación (E-R)**

El diagrama (E-R) debe realizarse de tal manera que sea fácil de leer por la gente que necesita trabajar con él. Sus características son:

#### **LIMPIO Y ORDENADO**

- Alinear las cajas de las entidades.
- Dibujar las líneas de relación como rectas horizontales o verticales.
- Usar un ángulo de 30° a 60° el cual facilita seguir las líneas de la relación cuando éstas se cruzan.
- Evitar el uso de muchas líneas paralelas, ya que dificulta el seguimiento.

#### **TEXTO CLARO**

- Escribir el texto con claridad, en forma concisa.
- Evitar abreviaciones y modismos.
- Agregar adjetivos para mejorar el entendimiento.
- Alinear el texto horizontalmente.
- Poner el nombre de la relación al final de la línea y en lados opuestos de la línea.

## FORMAS FÁCILES DE RECORDAR

- Hacer el Diagrama E-R fácil de recordar. Que la gente recuerde las formas y los patrones.
- No se deben dibujar diagramas E-R en una cuadrícula.
- Compactar en la medida de lo posible las cajas de las entidades para ayudar a la visualización del diagrama.

## REGLAS DE FORMATO

- Tratar de posicionar las "patas de gallo" como las de las figuras 3.11 y 3.12, en la parte izquierda para líneas horizontales y en la parte superior para las líneas verticales.
- Posicionar las entidades más volátiles arriba y a la izquierda del diagrama.
- Posicionar las entidades menos volátiles abajo y a la derecha del diagrama.

Los *atributos* representan información específica que se necesita conocer o tener acerca de una entidad. Los atributos describen una entidad para calificar, identificar, clasificar, cuantificar o expresar el estado de una entidad. Dicho de otra manera, representan un tipo de descripción o detalle, mas no una instancia.

Los atributos deben tener nombres claros para el usuario y no codificados para el desarrollador. Un atributo debe estar asignado en una sola entidad y deben ser descompuestos hasta su mínimo componente.

También es importante verificar que un atributo no sea derivado o calculado de los valores existentes de otros atributos.

La *opcionalidad de atributos* puede ser clasificada como sigue:

### *Atributos obligatorios.*

- Un valor *debe ser* conocido por cada ocurrencia de la entidad.
- Marcarlo con un asterisco (\*).

### *Atributos opcionales.*

- Un valor *puede ser* conocido por cada ocurrencia de la entidad.
- Marcarlo con una o minúscula.

Los atributos se pueden *identificar* examinando las notas de entrevistas y realizando preguntas al usuario.

Los atributos pueden aparecer en notas de entrevistas como:

- Frases y palabras descriptivas.
- Sustantivos.
- Frases preposicionales.
- Pronombres y sustantivos posesivos.

Un *identificador único* (UID) es cualquier combinación de atributos y/o relaciones que sirven para identificar en forma única una ocurrencia o instancia de una entidad. Cada ocurrencia de una entidad debe ser identificada de manera única.

En algunas ocasiones el identificador único de una entidad forma parte del identificador único de otra, en estos casos se utiliza la *barra UID*, como en la figura 3.11.

Una relación incluida dentro de un UID debe ser obligatoria y *una y solamente una* en la dirección que participa en el UID.

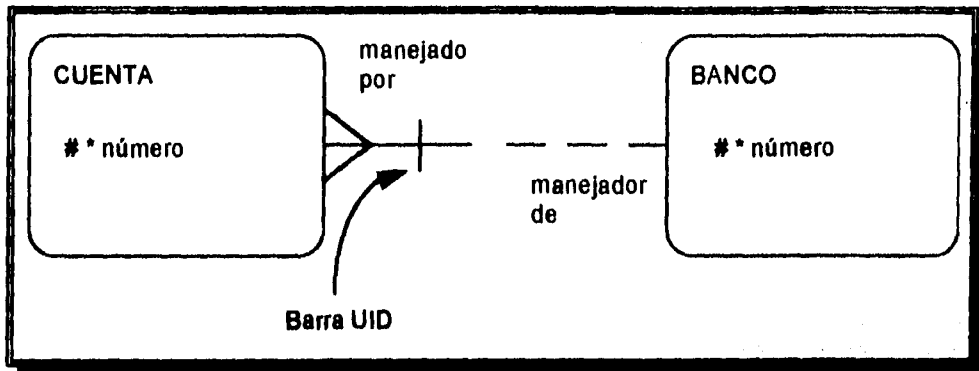


Figura 3.11. La barra UID indica que la relación con el BANCO es parte del UID de CUENTA.

Una entidad puede ser identificada de manera única a través de múltiples relaciones como lo indica la figura 3.12.

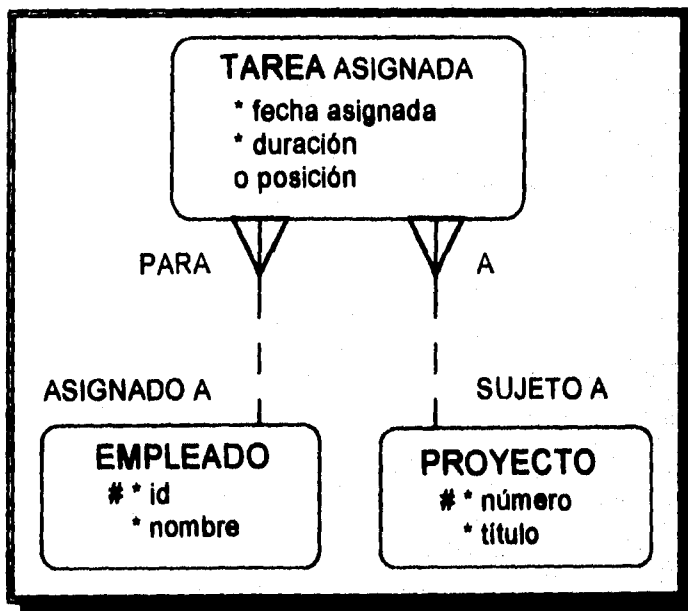


Figura 3.12 a. La TAREA ASIGNADA no tiene UID.

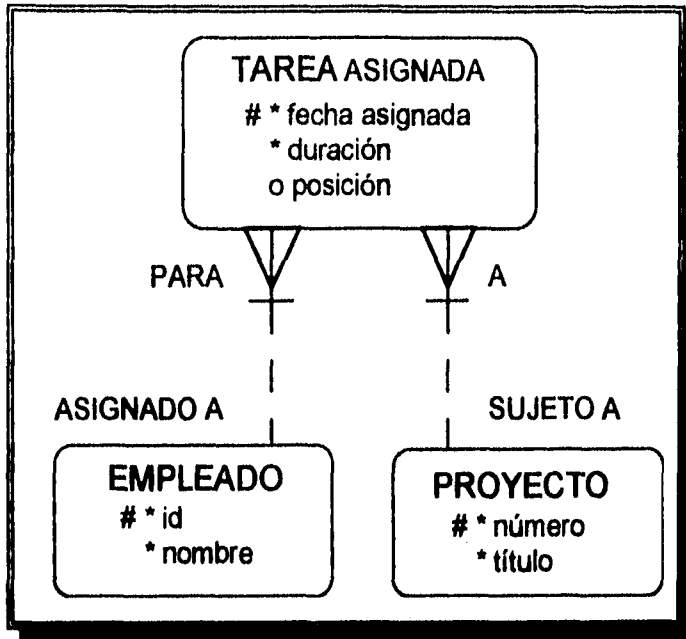


Figura 3.12 b. Una TAREA ASIGNADA es identificada de manera única por el EMPLEADO a quien se le dio la TAREA ASIGNADA, por el PROYECTO que contiene la TAREA ASIGNADA y por la fecha asignada.

Ambas relaciones son obligatorias y una y sólo una, en la dirección incluida en el UID.

Una entidad puede tener más de un UID como se ilustra en la figura 3.13

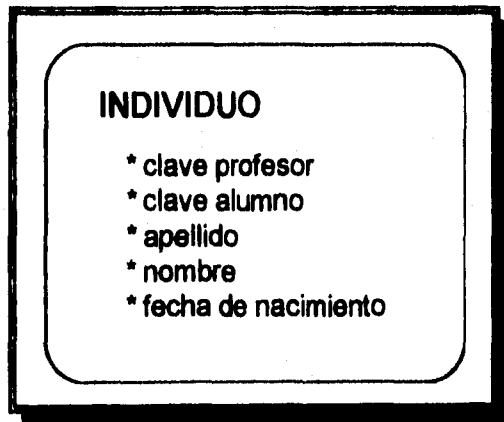


Figura 3.13. Candidatos a UID: "clave profesor", "clave alumno" y "apellido/nombre".

Aquí probablemente la combinación de apellido y nombre no es única.

En casos como el anterior se elige un candidato UID para ser el UID primario, y los otros para ser los UID secundarios tal como lo muestra la figura 3.14.

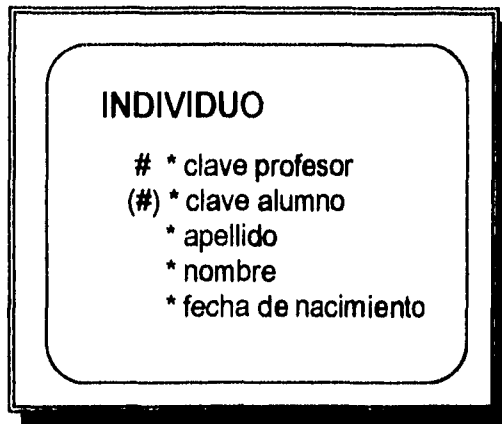


Figura 3.14. Cada UID se marca con (#) o se deja sin ninguna marca.

Cuando una entidad no ofrezca posibles candidatos para ser UID, es válido crear uno artificial que asuma dicho papel.

Esto se ilustra en la figura 3.15.

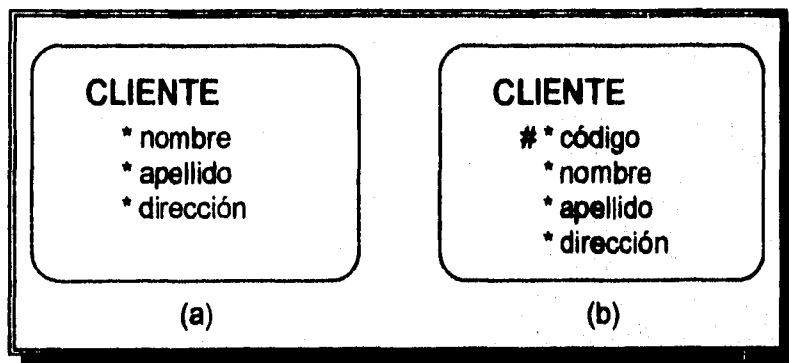


Figura 3.15. En la entidad CLIENTE (a) no se aprecia un posible UID. Por lo que fue necesario crear un atributo artificial "código" en la entidad CLIENTE (b) para habilitarlo como UID.

### **3.6 Normalizar el Modelo de Datos**

Normalizar es un concepto de bases de datos relacional, pero sus principios se aplican al Modelo Conceptual de Datos. Para validar cada atributo se deben utilizar las reglas de normalización a nivel diseño Modelo de Datos, con lo cual se traslada en forma automática a un diseño de base de datos.

La *primera forma normal* dice que todos los atributos deben tener un sólo valor para cada instancia.

Se debe verificar que cada atributo tenga un valor único para cada ocurrencia de la entidad, es decir, ningún atributo deberá tener valores repetidos. Si un atributo tiene múltiples valores, se crea una entidad adicional y lo relaciona con la entidad original mediante una relación M:1

La segunda forma normal recomienda que un atributo debe ser dependiente del identificador único completo.

Se debe validar que cada atributo dependa completamente del UID. Cada instancia específica del UID debe determinar una sola instancia de cada atributo.

Validar que un atributo no dependa de una sola parte del UID de la entidad. Si un atributo no es dependiente del UID completo, está fuera de lugar y deberá ser movido.

Finalmente, la tercera forma normal menciona que ningún atributo no-UID puede ser dependiente de otro atributo no-UID. O sea, se debe validar que cada atributo no-UID no dependa de otro atributo no UID.

Se debe mover cualquier atributo no-UID que dependa de otro atributo no-UID.

Si un atributo depende de otro atributo no-UID, es necesario mover ambos, el atributo dependiente y el atributo del que depende, a una nueva entidad relacionada con la entidad actual.

Cuando algunos atributos se asocian con relaciones M:M (muchos a muchos) se tienen que resolver al sustituir una entidad intersección y dos relaciones M:1 (muchos a uno).

Por ejemplo, si consideramos la relación M:M entre PRODUCTO y VENDEDOR, ilustrada en la figura 3.16 ¿Cuál es el precio actual de un PRODUCTO específico para un VENDEDOR específico?

La tercera forma normal de diseño es un objetivo generalmente aceptado para eliminar redundancia en un diseño de base de datos.

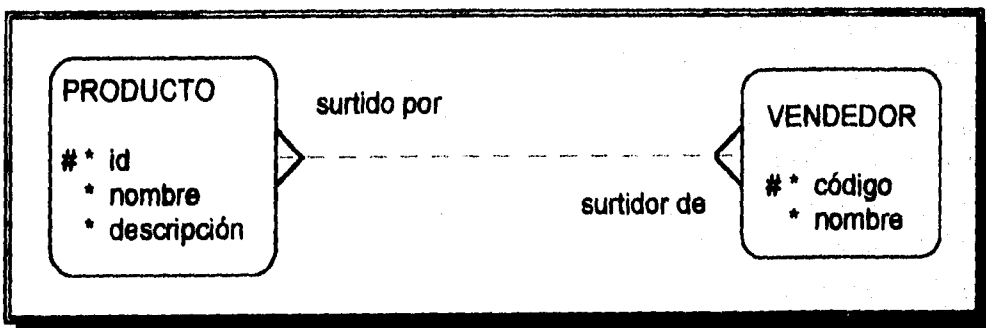


Figura 3.16 Ejemplo de una relación M:M

El precio actual parece ser un atributo de la relación entre PRODUCTO y VENDEDOR.

Los atributos únicamente describen entidades. Si los atributos describen relaciones, las relaciones deberán ser resueltas.

La relación M:M entre PRODUCTO y VENDEDOR puede ser resuelta agregando una entidad intersección llamada CATÁLOGO.

El precio actual es realmente un atributo de la entidad CATÁLOGO, tal como se muestra en la figura 3.17.

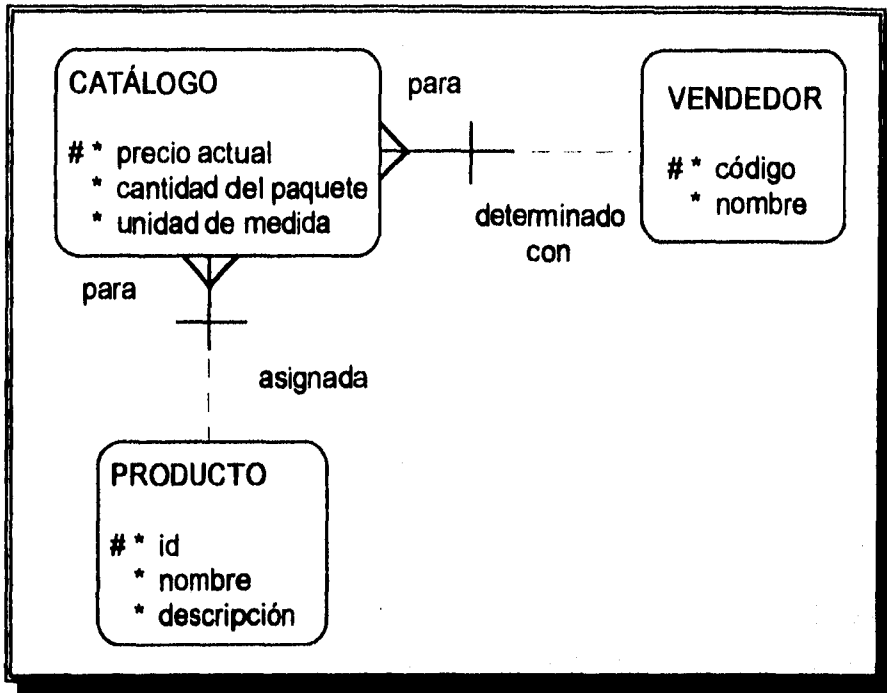


Figura 3.17 Ejemplo de una entidad Intersección.

Una vez que está definida la entidad CATÁLOGO, se analizan los requerimientos del cliente para ver la posibilidad de agregar atributos como: cantidad del paquete y unidad de medida.

El UID del CATÁLOGO esta compuesto por las dos relaciones.

Una entidad Intersección es frecuentemente identificada por las dos relaciones que le dieron origen, como se ve en la figura 3.17.

Las relaciones desde una entidad intersección son siempre obligatorias y son muy comunes para representar situaciones de negocios en el mundo real.

Las entidades intersección normalmente generan requerimientos para atributos adicionales como el uso de cantidades y fechas. Estas tienden a ser entidades volátiles y de volumen alto.

No olvidar que el UID de una entidad intersección está frecuentemente compuesta por las relaciones entre las entidades que le dieron origen.



Si se resuelven las relaciones M:M al final de la fase de Análisis se puede llegar a tener una entidad Intersección sin atributos.

Una entidad intersección sin atributos es justamente una lista de referencia cruzada en doble sentido entre las ocurrencias de las entidades.

Una entidad intersección sin atributos es la excepción a la regla de que una entidad debe tener atributos para ser una entidad. Existen otros tipos de modelos para representar las relaciones entre entidades, como los que a continuación se presentan:

*Modelo jerárquico de datos.* Representación jerárquica de datos como un conjunto de relaciones muchos a uno.

*Modelo Recursivo.* En el modelo recursivo se representan relaciones entre una entidad y ella misma.

*Modelo de roles.* Este modelo se usa cuando una misma persona puede estar considerada en varias entidades.

*Modelo de subtipos.* Usar subtipos para modelar exclusivamente tipos de entidad que tienen atributos o relaciones comunes.

Un supertipo es una entidad que tiene subtipos y puede ser separado en dos o más subtipos mutuamente excluyentes. También puede tener atributos y relaciones compartidas por sus subtipos.

*Modelos de relaciones exclusivos.* Para modelar dos o más relaciones mutuamente excluyentes de la misma entidad se usa un arco.

Las relaciones en arco frecuentemente tienen los mismos nombres de relación y deben ser o todas obligatorias o todas opcionales. Un arco pertenece a una sola entidad y solamente debe incluir relaciones originadas de esta entidad.

Una entidad puede tener múltiples arcos, pero una relación específica solamente puede participar en un sólo arco.

*Modelos de datos de tiempo.* Se utilizan cuando se requiere agregar entidades y relaciones al modelo Entidad-Relación para acomodar datos históricos.

Una entidad intersección se usa frecuentemente para mantener información acerca de las relaciones que cambian con el tiempo.

### **3.7 Conceptos de Base de Datos Relacional**

Una Base de Datos Relacional es aquella que es percibida por el usuario como una colección de relaciones o de tablas de dos dimensiones.

Las tablas de Base de Datos Relacional son sencillas pero disciplinadas, debe tener Integridad de Datos, sus datos deben ser precisos y consistentes.

El Instituto Nacional Americano de Estándares (ANSI) ha establecido a SQL como el lenguaje estándar para operar sobre las Bases de Datos Relacionales.

Una Base de Datos Relacional puede soportar un conjunto completo de operaciones relacionales. Las operaciones relacionales manipulan conjuntos de valores de datos. Las tablas pueden ser utilizadas en la creación de otras tablas. Las operaciones relacionales pueden ser anidadas.

Una *Llave Primaria (PK)* es una columna o grupo de columnas que identifican de manera única a cada renglón en una tabla.

Cada tabla debe tener una llave primaria y ésta debe ser única, no pueden cambiar de valor. El UID de una entidad irá de acuerdo con la Llave Primaria en su tabla correspondiente.

Una Llave Primaria que consta de múltiples columnas se llama *Llave Primaria Compuesta*, cuyas columnas deben ser únicas en combinación. Éstas pueden tener duplicados en forma individual, pero en combinación, no se permiten duplicados. Ninguna parte de la Llave Primaria puede ser NULA.

Una tabla puede tener más de una columna o combinación de columnas que pueden servir como la Llave Primaria de la tabla. Cada una de estas es llamada Llave Candidata o Alterna. Todas las Llaves Alternas deben ser únicas y NO NULAS. Los UID secundarios concuerdan con las Llaves Alternas. Los nombres de personas normalmente no son Llaves Alternas por que no se puede garantizar que sean únicas.

Una *Llave foránea (FK)* -o secundaria-, es una columna o combinación de columnas en una tabla, que se refieren a una llave primaria en la misma o en otra tabla. Las llaves foráneas son utilizadas para hacer "JOIN" entre tablas, se basan en los valores de los datos, son puramente lógicas, pueden ser repetidas, nulas y pueden coincidir con un valor de una Llave Primaria existente. Si una llave foránea es parte de una Llave Primaria, la FK no puede ser NULA.

La *integridad de datos* se refiere a la exactitud y consistencia de los datos. Las *restricciones<sup>1</sup> de integridad* de datos definen el estado relacional correcto de la base de datos, aseguran que los usuarios realicen únicamente operaciones en las cuales dejarán a la base de datos en un estado correcto y consistente.

Todos los constraints de integridad de datos deben ser forzados por el DBMS o el software de aplicación.

Un dato es inconsistente si existen múltiples copias de un registro y no todas las copias han sido actualizadas. Una base de datos inconsistente puede proveer información incorrecta o contradictoria a los usuarios.

---

<sup>1</sup>1. Se refiere al término *constraints*, el cual no es muy común o familiar.

Los requerimientos del usuario también pueden determinar el estado correcto de una base de datos. Estos requerimientos son llamados Constraints de Integridad de Datos definidas por el usuario.

Los constraints de los datos definidos por el usuario pueden ser administrados por políticas o ser requeridos por las leyes gubernamentales.

Frecuentemente esos requerimientos son completamente arbitrarios o al menos parecen ser arbitrarios.

Los constraints de integridad de datos definidos por el usuario pueden incluir múltiples columnas y tablas.

### **3.8 Normalización de Tablas**

La normalización minimiza la redundancia de los datos y ayuda a identificar entidades, relaciones y tablas mal diseñadas.

Un dato sin normalizar es redundante, ocasionando problemas de integridad. Las transacciones de actualización y borrado pueden no ser consistentes en todas las copias de los datos causando inconsistencia en la base de datos.

Cada valor de una llave no primaria DEBE depender únicamente de la llave completa, solo de ella y no de ningún otro campo.

Las reglas de las formas normales se muestran en la tabla 3.1

Regla de la Forma Normal	Descripción
Primera Forma Normal (1FN)	La tabla debe tener un sólo valor para cada renglón. La tabla no puede contener grupos repetitivos.
Segunda Forma Normal (2FN)	La tabla debe estar en 1FN. Cada columna que no es llave debe ser dependiente de la llave primaria completa.
Tercera Forma Normal (3FN)	La tabla debe estar en 2FN. Una columna que no es llave primaria no debe depender de otra columna no llave.

Tabla 3.1 Reglas de las Formas Normales.

La tercera forma normal es un objetivo normalmente aceptado para un diseño de base de datos para eliminar la redundancia.

Las formas normales posteriores ya no son utilizadas.

Un dato sin normalizar no cumple con ninguna regla de normalización.

Para realizar la *conversión a la primera Forma Normal*, se deben remover los grupos de repetición o vectores de la base de datos y crear una nueva tabla con la PK de la tabla base y el grupo de repetición.

Para llevar a cabo la *conversión a la segunda Forma Normal*, se deben determinar cuáles columnas, que no son llave, no dependen de la llave primaria completa de la tabla, también se deben remover esas columnas de la tabla base y crear una segunda tabla con esas columnas y la(s) columna(s) de la PK de la cual dependen.

Si cada columna no depende de la llave primaria completa, la tabla no está en 2FN.

Cualquiera tabla con una llave primaria de una sola columna está automáticamente en la 2FN.

Para hacer la *conversión a la tercera Forma Normal*, se deben determinar que columnas son dependientes de otra columna no llave. Después se remueven esas columnas de la tabla base y al final se crea una segunda tabla con esas columnas no llave de la cual son dependientes.

Una tabla está en Tercera Forma Normal si una columna no llave no es funcionalmente dependiente de otra columna no llave.

Una columna no llave no puede ser funcionalmente dependiente de otra columna no llave.

El *especificar la integridad referencial* implica considerar que el valor de una columna de llave foránea debe coincidir con un valor existente en otra columna que sea llave primaria (o ser NULO). Usar los constraints de la integridad referencial para especificar como se debe mantener la integridad referencial.

Se recomienda especificar el Constraint de borrado para definir qué debe suceder si un renglón que contiene una llave primaria referenciada es borrada, así como especificar un Constraint de actualización para definir qué debería suceder cuando una llave primaria es actualizada.

En lo que se refiere al *diseño de índices*, se debe tener presente que un índice es asociado con una sola tabla física y contiene los valores de una o más columnas de esa tabla. Al usar índices se mejora considerablemente el acceso a datos.

Los índices proveen acceso rápido a los renglones de datos y evitan búsquedas en toda la tabla, facilitan uniones (join) entre tablas, aseguran que no existe un valor duplicado si se define como único y son usados automáticamente cuando se referencian en las cláusulas WHERE de las instrucciones de SQL (siempre y cuando la columna no sea modificada).

Un índice concatenado es un índice creado en un grupo de columnas en una sola tabla, también se usan índices para implementar llaves y soportar los requerimientos de acceso de la aplicación.

Los índices únicos se crean para llaves primarias y generalmente los índices no únicos para llaves foráneas. Un índice único hace referencia a una columna o grupo de columnas que tienen valores únicos en la tabla. Los índices no únicos hacen referencia a columnas o grupos de columnas que no son únicas en la tabla.

La creación de índices también se considera para llaves alternativas (índices únicos), en columnas no llaves críticas usadas en las cláusulas WHERE y para cualquier llave de búsqueda.

Los índices requieren espacio y generan una sobre carga al momento de hacer actualizaciones. Asimismo, se debe tener en cuenta que bajo ciertas condiciones, los índices no son usados por el RDBMS.

### **3.9 Diagrama de Transición de Estados**

El diagrama de transición de estados, también conocido como DTE, enfatiza el comportamiento dependiente del tiempo del sistema.

Hasta hace poco, los modelos del comportamiento dependiente del tiempo del sistema importaban sólo para una categoría especial de sistemas conocidos como sistemas de tiempo-real.

Ejemplos de estos sistemas son: control de procesos, sistemas de conmutación telefónica, sistemas de captura de datos de alta velocidad, etc. Algunos de estos sistemas son pasivos, es decir, no buscan controlar el ambiente que los rodea, sino más bien reaccionan a él o capturan datos que le atañen.

Otros sistemas de tiempo real son más activos, o sea que, pretenden mantener el control sobre algún aspecto del ambiente que los rodea.

Los sistemas de este tipo manejan fuentes externas de datos de alta velocidad, y deben proporcionar alguna respuesta y datos de salida de manera suficientemente rápida como para manejar el ambiente externo. Una parte importante de la especificación de tales sistemas es la descripción de *qué sucede cuando*.

Para los sistemas enfocados a cuestiones administrativas, lo tratado anteriormente normalmente no ha sido tan importante. Las entradas pueden llegar al sistema de diferentes fuentes a velocidades relativamente altas, pero habitualmente se pueden detener si el sistema está ocupado haciendo otra cosa.

Generalmente los únicos asuntos que involucran tiempos en este tipo de sistemas son especificaciones de tiempo de respuesta.

## NOTACIÓN DE LOS DIAGRAMAS DE TRANSICIÓN DE ESTADOS

En la figura 3.18 se muestra un DTE típico. Este diagrama muestra el comportamiento de una máquina contestadora de teléfono normal.

Los componentes del diagrama son los *estados* y los *cambios de estado* representados por rectángulos y por flechas respectivamente.

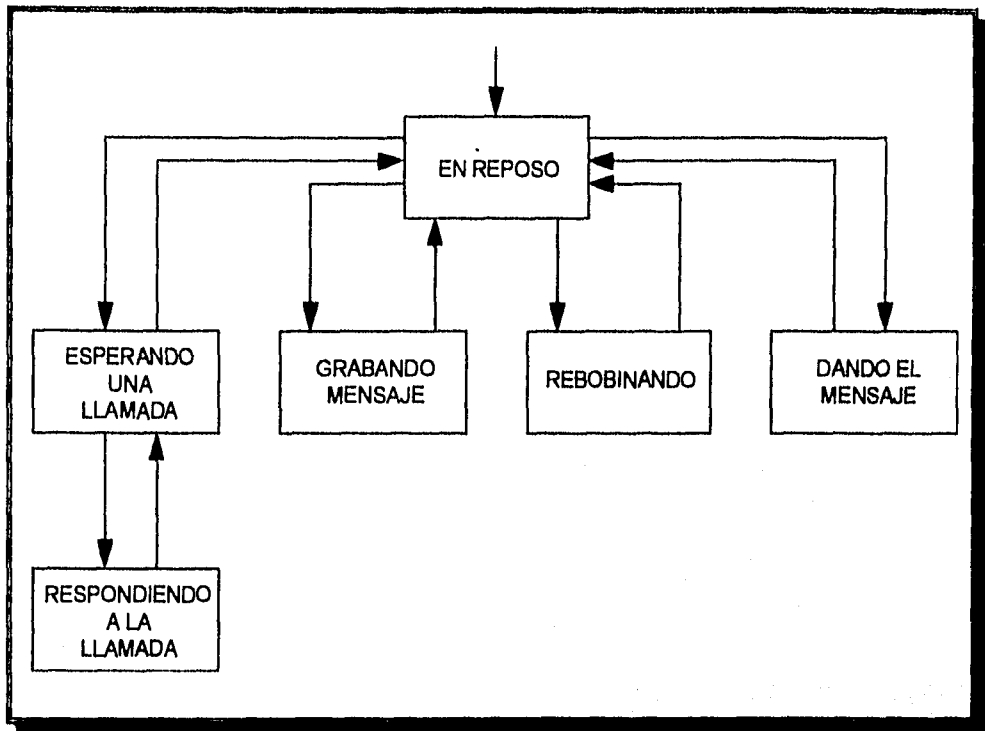


Figura 3.18 Diagrama típico de transición de estados.

Un estado se define como *un conjunto de circunstancias o atributos que caracterizan a una persona o cosa en un tiempo dado: forma de ser; condición económica, etc.* Cada rectángulo representa un estado en el que se puede encontrar el sistema.

Muchos de los estados indican que el sistema está esperando a que algún evento ocurra, y no se expresan en términos de que la computadora esté haciendo algún procedimiento. Esto se debe a que el diagrama de transición de estados se usa para desarrollar un modelo esencial del sistema, es decir, un modelo de cómo se comportaría el sistema si hubiera tecnología perfecta.

Un aspecto de la tecnología perfecta sería que la computadora trabaje de manera infinitamente rápida, de modo que cualquier proceso o cálculo que tenga que hacer, o cualquier acción que deba tomar, se haga en cero momentos.

Así, cualquier estado observable en el que el sistema se pueda encontrar sólo puede corresponder a periodos en los que :

- a) está esperando que algo ocurra en el ambiente externo o,
- b) está esperando a que alguna actividad que se esté dando en ese momento en el ambiente cambie a otra.

Esto no significa que los sistemas sean incapaces de ejecutar acciones o que no se pretendan mostrarlas, sino sólo que las acciones, que ocurren instantáneamente en nuestro modelo de tecnología perfecta, no son lo mismo que los estados, que representan condiciones observables en las que el sistema se puede encontrar. Esto es, un estado representa algún comportamiento del sistema que es observable y que perdura durante algún periodo finito. Es importante recordar que un estado cambia periódicamente.

Un sistema que existió en un solo estado no sería un objeto de estudio muy interesante: sería estático. De hecho, los sistemas de información que se modelan normalmente pueden tener docenas de estados diferentes.

Un sistema cambia de un estado a otro al tener gobernado su comportamiento por reglas ordenadas. Se muestran los cambios de estado válidos en el DTE conectando pares relevantes de estados con una flecha.

Así, la figura 3.19 muestra que el sistema puede ir del estado 1 al estado 2. También muestra que cuando el sistema se encuentre en el estado 2 puede ir al estado 3 o regresar al 1.

Sin embargo, de acuerdo con este DTE, el sistema no puede ir directamente del estado 1 al 3. Por otro lado, el diagrama representa que el sistema sí puede ir directamente del estado 3 de regreso al 1.

Hay que notar también que el estado 2 tiene dos estados sucesores. Esto es muy común en los DTE; de hecho, cualquier estado puede llevar a un número arbitrario de estados sucesores.

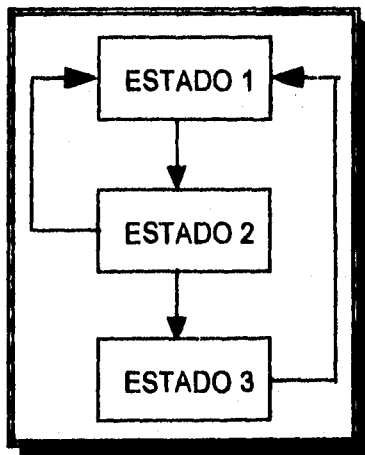


Figura 3.19 Cambios de estado

A pesar de que la figura 3.19 proporciona información interesante acerca del comportamiento dependiente del tiempo de un sistema, no dice algo que pudiera resultar ser muy importante: cuáles son los estados *inicial* y *final* del sistema. De hecho, la figura 3.19 es un modelo de estado estable de un sistema que ha estado siempre activo y que continuará siempre estándolo.

La mayoría de los sistemas tienen un estado inicial y un estado final reconocibles; esto se muestra en la figura 3.20

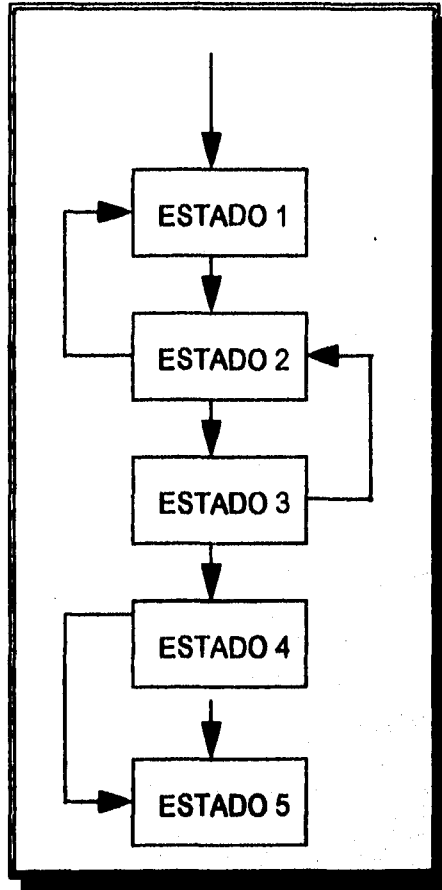


Figura 3.20 Estados inicial y final.

El estado inicial típicamente suele ser el que se dibuja en la parte superior del diagrama, aunque no es obligatorio. Lo que realmente identifica al estado 1 en la figura 3.20 como inicial es la flecha "desnuda" que no está conectada a ningún otro estado.

De manera similar, el estado final suele ser el que se dibuja en la parte inferior, pero tampoco esto es obligatorio. Lo que realmente identifica al estado 5 como final es la ausencia de una flecha que salga de él. En otras palabras, una vez que se llega al estado 5 ya no se puede ir a ninguna otra parte.



El sentido común indica que un sistema sólo puede tener un estado inicial; sin embargo, puede tener múltiples estados finales.

Los diversos estados finales son mutuamente excluyentes, lo cual significa que sólo uno de ellos puede ocurrir durante alguna ejecución del sistema.

La figura 3.21 muestra un ejemplo en el que los posibles estados finales son el 4 y el 6.

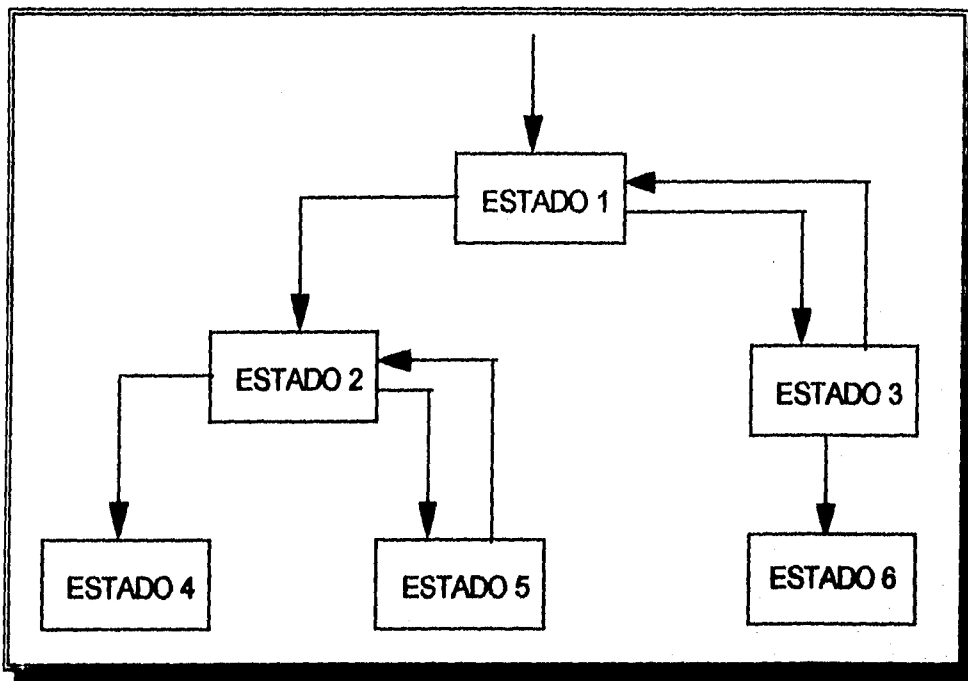


Figura 3.21 Estados finales múltiples de un sistema

Dado que se usa un DTE para construir un modelo esencial, también se puede suponer que los cambios de estado ocurren instantáneamente; es decir, no se requiere tiempo observable para que el sistema cambie de un estado a otro.

Cuando los diseñadores y programadores empiezan a construir un modelo de implantación, esto será un asunto real: normalmente en una computadora el cambio de una actividad del proceso a otra sí tarda algunos microsegundos, y se debe asegurar que suceda lo suficientemente rápido como para que el ambiente no se salga de control.

Finalmente, para tener completa la descripción de un DTE se necesitan añadir dos factores más: las *condiciones* que causan un cambio de estado y las *acciones* que el sistema toma cuando cambia de estado. Esto se ilustra en la figura 3.22, las condiciones y acciones se muestran junto a la flecha que conecta dos estados relacionados.

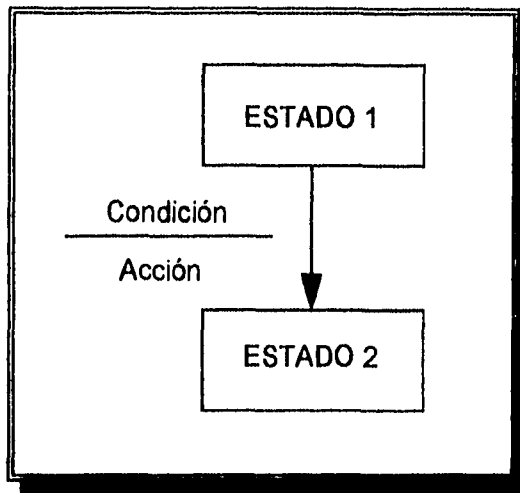


Figura 3.22 Muestra de condiciones y acciones.

Una condición es un acontecimiento en el ambiente externo que el sistema es capaz de detectar; típicamente es una señal, una interrupción o la llegada de un paquete de datos. Esto usualmente hace que el sistema cambie de un estado de esperar X a un estado de esperar Y; o de llevar a cabo la actividad X a llevar a cabo la actividad Y.

Como parte del cambio de estado, normalmente el sistema hará una o más acciones: producirá una salida, desplegará una señal en la terminal del usuario, llevará a cabo un cálculo, etc.

Así que las acciones que se muestran en un DTE son respuestas regresadas al ambiente externo o bien cálculos cuyos resultados el sistema recuerda (típicamente en un archivo de datos que se muestra en el DFD) para poder responder a algún acontecimiento futuro.

Es oportuno recurrir a los *diagramas particionados* cuando en un sistema complejo pueden existir docenas de estados distintos del sistema.

Tratar de poner todos todos estos estados en un mismo diagrama sería difícil, si no imposible. Por tanto, tal como se usaron los niveles y las particiones en los DFD, se pueden usar las particiones con los DTE.

Hay que advertir que en este caso, cualquier estado individual de un diagrama de mayor nivel puede convertirse en un estado *inicial* para un diagrama de nivel inferior que describe más a fondo ese estado de mayor nivel; y el o los estados finales en un diagrama de nivel inferior corresponden a las condiciones de salida en el estado asociado de nivel superior.

La figura 3.23 muestra un ejemplo de dos niveles de DTE para un sistema complejo.

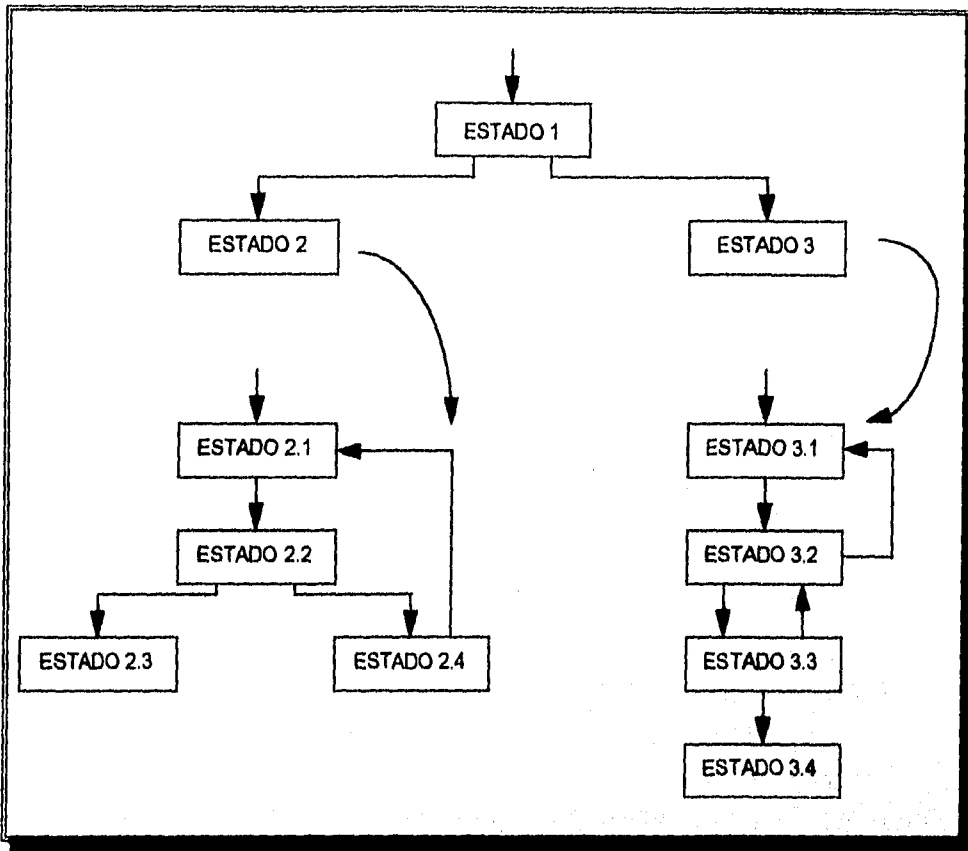


Figura 3.23 Dos niveles de un DTE

En otros casos, el analista puede requerir mostrar, explícitamente, cómo es que un DTE de menor nivel sale a algún lugar apropiado en el de nivel superior.

### CONSTRUCCIÓN DEL DIAGRAMA DE TRANSICIÓN DE ESTADOS

Para la construcción de un DTE se puede seguir uno de los dos siguientes enfoques:

- Se puede comenzar por identificar todos los posibles estados del sistema y representar cada uno como una caja separada en una hoja de papel. Luego, se pueden explorar todas las conexiones con significado, es decir, los cambios de estado entre las cajas.
- Como alternativa, se puede comenzar por el estado inicial, y luego metódicamente ir siguiendo un camino hasta el o los estados restantes; luego del o los estados secundarios, proseguir a los terciarios; etc.

El enfoque quedara determinado, en muchos casos, por el usuario con quien se trabaje, sobre todo si él es el unico que está familiarizado con el comportamiento dependiente del tiempo del sistema.

Cuando se termine de construir el DTE preliminar, deben seguirse las siguientes reglas para verificar la consistencia:

- Definir todos los estados. Se observa con cuidado el sistema para detectar si existe algún otro comportamiento observable, o alguna otra condición en la que el sistema pudiera estar, aparte de las que se han identificado.
- Asegurar que se puedan alcanzar todos los estados, o bien, identificar los estados que no tengan caminos que conduzcan a ellos para eliminarlos.
- Verificar que se pueda salir de todos los estados. El sistema puede tener uno o más estados finales con múltiples entradas a ellos; pero todos los demás estados deben tener un sucesor.
- Determinar todas las condiciones posibles a las que el sistema debe responder adecuadamente en cada estado. Uno de los errores más comunes cuando se construye un DTE es el identificar los cambios de estado cuando únicamente ocurren condiciones normales, pero no se especifica el comportamiento del sistema ante condiciones inesperadas.

Por ejemplo, si tomamos en cuenta el modelo de la figura 3.24, se espera que el usuario presione una tecla de función en su terminal para causar un cambio de un estado 1 a un estado 2, y una tecla diferente para ir del estado 2 al 3. Pero ¿qué pasa si el usuario presiona la misma tecla dos veces seguidas? ¿O alguna otra tecla?

Si no se especifica el comportamiento del sistema, existe una buena posibilidad de que los diseñadores y programadores no lo programen tampoco, y el sistema tendrá un comportamiento impredecible bajo una variedad de circunstancias.

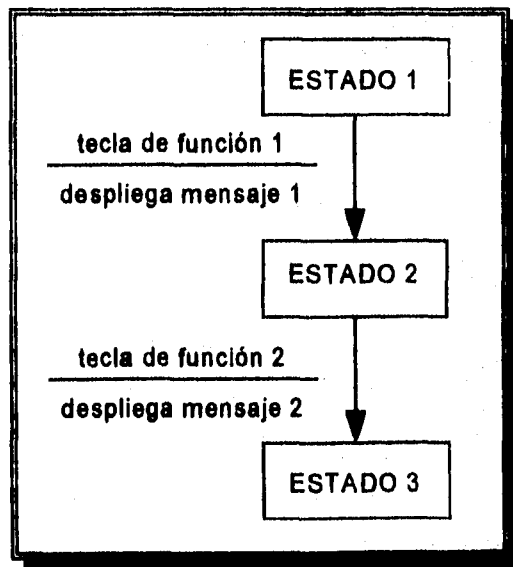


Figura 3.24 DTE incompleto

### **3.10 El Diccionario de Datos**

El diccionario de datos es otra herramienta del modelado del sistema; si bien es cierto que no tiene el atractivo gráfico de los DFD, es indispensable.

Sin el diccionario de datos, el modelo de los requerimientos del usuario no puede considerarse completo; todo lo que se tendría es un borrador rudimentario, una "visión del artista" del sistema.

La importancia del diccionario de datos a menudo les pasa de largo a muchas personas, pues no han utilizado algún tipo de diccionario durante muchos años. Es seguro que sin él, llegará el momento en que el usuario lector se extraviará y no podrá estar seguro de que entendió los detalles de la aplicación.

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

- Describe el *significado* de los flujos y archivos que se muestran en los DFD.
- Describe la *composición* de agregados de paquetes de datos que se mueven a lo largo de los flujos, es decir, paquetes complejos (por ejemplo el domicilio de un cliente), que puede descomponerse en unidades más elementales (como ciudad, estado y código postal).
- Describe la *composición* de los paquetes de datos en los archivos.
- Especifica los *valores* y *unidades* relevantes de piezas elementales de información en los flujos de datos y en los archivos de datos.
- Describe los detalles de las relaciones entre los archivos que se enfatizan en un diagrama de entidad-relación.

El hecho de utilizar una notación para el diccionario de datos representa una necesidad, ya que en la mayoría de los sistemas reales con los que se trabaja, los conjuntos o elementos de datos, serán lo suficientemente complejos como para que se necesite describirlos en términos de otros conceptos.

Los elementos complejos de datos se definen en términos de elementos más sencillos, y los sencillos en términos de los valores y unidades legítimos que pueden asumir.

Existen muchos esquemas de notación comunes utilizados por el analista de sistemas. El que se muestra en la tabla 3.2 es de los más comunes y utiliza varios símbolos sencillos.

=	está compuesto de
+	y
( )	optativo (puede estar presente o ausente)
{ }	iteración
[ ]	seleccionar una de varias alternativas
* comentario *	comentario o descripción
#	identificador (campo clave) para un archivo
	separa opciones alternativas en la construcción

Tabla 3.2 Notación del diccionario de datos.

Como puede apreciarse, los símbolos parecen algo matemáticos y pudiera pensarse que serían difíciles de entender. Sin embargo, la notación es bastante fácil de leer.

### DEFINICIONES =

La definición de un dato se introduce con el símbolo "=". En este contexto, el "=" se lee: "se define como", o "se compone de", o simplemente "significa".

Por ejemplo la notación  $Z = X + Y$  se puede leer de las siguientes formas:

Z esta descompuesto de X y Y.

o

Z se define como X y Y.

Para definir por completo un dato, se debe incluir lo siguiente:

- El significado del dato dentro del contexto de la aplicación del usuario. Por lo común se utiliza la notación: "\* significado o comentario \*".
- La composición del dato, si se compone de partes elementales con significado.
- Los valores que puede tomar el dato, si se compone de partes elementales con significado.
- Los valores que puede tomar el dato, si es un dato elemental que no puede descomponerse más.

Las *partes elementales de los datos* son aquellas para las cuales ya no existe una descomposición con significado dentro del contexto del ambiente del usuario.

Esto usualmente es una cuestión de aplicación y es algo que se debe explorar cuidadosamente con el usuario. Pero tal vez en algunos ambientes de usuario no se requiere tal descomposición, ni sea relevante, ni tenga significado.

Cuando se han identificado los datos elementales, deben introducirse al diccionario de datos. Como se indicó anteriormente, el diccionario de datos debe proporcionar una breve narrativa, encerrada entre caracteres "\*", que describa el significado del término en el contexto del usuario. Desde luego, habrá términos que se definan solos, es decir, cuyo significado es universal para todos los sistemas de información, o donde el analista pudiera estar de acuerdo en que no se necesita aclarar más.

Por ejemplo apellido paterno, peso actual, sexo, teléfono particular, etc.

Para tales casos se usa la notación "\*" para indicar "sin comentarios", ya que no se escribe algún texto explicando el significado del dato. Sin embargo, es importante especificar los valores y unidades de medida que pueden tomar.

### **DATOS OPCIONALES ( )**

Un dato opcional, como la frase implica, es aquel que puede estar o no presente en un dato compuesto.

Por ejemplo salón de clases = (ubicación) + (nombre del salón) significa que un salón puede ser identificado sólo por su ubicación o bien, sólo por su nombre o bien, por su ubicación y nombre o aunque no es usual por ninguno de los dos.

### **ITERACION { }**

La notación de iteración se usa para indicar la ocurrencia repetida de un componente de un dato. Se lee como "cero o más ocurrencias de".

Por ejemplo inscripción = nombre del alumno + domicilio del alumno + {materia} significa que la inscripción siempre debe contener un nombre de alumno, un domicilio y también cero o más ocurrencias de una materia. Es decir, es un alumno que está inscrito en una, o dos o más materias.

Cuando se quieren especificar límites inferior y/o superior se anotan en los extremos de las llaves; como se indica aquí 1{materia}5, un alumno se puede inscribir en 1, 2, 3, 4 ó 5 materias.

### **SELECCION [ ]**

La notación de selección indica que un dato consiste en exactamente un elemento de entre un conjunto de opciones alternativas. Las opciones se encierran en corchetes "[" y "]", y se separan por una barra vertical "|".

Por ejemplo sexo = [Femenino | Masculino]

## ALIAS

Un alias, como el término implica, es una alternativa de nombre para un dato. Esto es una ocurrencia común cuando se trata con diversos grupos de usuarios en diferentes departamentos o ubicaciones geográficas, que consisten en utilizar distintos nombres para decir lo mismo.

El alias se incluye en el diccionario de datos para que esté completo, y se relaciona con el nombre primario u oficial del dato.

Por ejemplo:            investigador =            \* alias de profesor \*

En este caso, la composición de investigador no se anota debido a que es un alias de profesor y de esta manera "hereda" la composición de éste último.

En un sistema mediano o grande, el diccionario de datos puede requerir una cantidad considerable de trabajo y tiempo.

Lo más recomendable es hacer uso de la computadora para introducir definiciones al diccionario, verificar que estén completas y consistentes, y producir reportes apropiados.

Se deben tener presentes, además, las siguientes limitaciones posibles:

- Estar forzado a limitar los nombres de datos a cierta longitud. Esto probablemente no sea un gran problema, pero podría el usuario insistir en un nombre como *nombre-completo-alumno* y que el paquete de elaboración del diccionario obligue a abreviarlo como *nom-com-alu*.
- Pudiera ser que el usuario no permita ciertos caracteres al elaborar el nombre del dato, como el guión ( - ) por ejemplo.
- El diccionario de datos siempre debe estar completo y terminado antes de llevar a cabo la elaboración del pseudocódigo y obviamente del código de los programas. Para lograrlo quizá sea necesario asignar dos o más analistas para su elaboración.

### **3.11 Diseño Estructurado Moderno**

La capacidad que exhiban los programadores para implantar un sistema de alta calidad y libre de errores depende en gran medida de la naturaleza del diseño; de manera similar, la capacidad de los programadores de mantenimiento para realizar cambios en el sistema después de haberlo puesto en operación depende de la calidad del diseño.

El campo del diseño estructurado ofrece guías para ayudar al diseñador a determinar los módulos, y sus interconexiones, que mejor den atención a los requerimientos especificados por el analista.



Las dos reglas más importantes son las referentes al acoplamiento y la cohesión:

La *cohesión* se refiere al grado en el cual los componentes de un módulo (típicamente las instrucciones individuales que conforman un módulo) son necesarios y suficientes para llevar a cabo una sola función bien definida. En la práctica, esto significa que el diseñador debe asegurarse de no fragmentar los procesos esenciales en módulos, y también debe asegurarse de no juntar procesos no relacionados (que se representan por burbujas en el DFD) en módulos sin sentido. Los mejores módulos son aquellos que son funcionalmente cohesivos (es decir, módulos en los cuales cada instrucción es necesaria para poder llevar a cabo una sola tarea bien definida). Los peores módulos son los que son coincidentalmente cohesivos (es decir, cuyas instrucciones no tienen una relación significativa entre uno y otro).

El *acoplamiento* es el grado en el cual los módulos se interconectan o se relacionan entre ellos. Entre más fuerte sea el acoplamiento entre módulos en un sistema, más difícil es implantarlo y mantenerlo, pues entonces se necesitará un estudio cuidadoso para la modificación o cambio y modificación de algún módulo o módulos. En la práctica, esto significa que cada módulo debe tener interfases sencillas y limpias con otros, y que se debe compartir un número mínimo de datos entre módulos. También significa que un módulo dado no debe modificar la *lógica interna* o los datos de algún otro módulo; lo que se conoce como una conexión patológica.

Además de las características anteriores, el diseñador debe tomar en cuenta otros parámetros para optimizar el sistema. Algunos de ellos son los siguientes:

El *tamaño del módulo*. De ser posible, cada módulo debe ser lo suficientemente pequeño como para caber en una sola página o para que pueda desplegarse en una sola pantalla. Desde luego, a veces no es posible determinar qué tan grande va a ser un módulo hasta haberlo escrito, pero las actividades iniciales de diseño a menudo darán al diseñador una buena pista de que el módulo va a ser grande y complejo.

Si es así, debe partirse en uno o más niveles de submódulos. En raras ocasiones, los diseñadores crean módulos que son triviales.

Por ejemplo, módulos que consisten en sólo dos o tres renglones de código. En este caso, pueden juntarse varios en un solo supermódulo mayor.

El *alcance del control* se refiere al número de subordinados inmediatos que un módulo administrador puede llamar. Un módulo no debe poder llamar a más de una media docena de módulos de nivel inferior.

La razón es evitar la complejidad: si el módulo tiene, digamos, 25 módulos de nivel inferior, entonces probablemente contendrá tanta lógica compleja de programa (en la forma de declaraciones **IF** anidadas, o de iteraciones **DO-WHILE** anidadas, etc.) que nadie lo podrá entender.

La solución es introducir un nivel intermedio de módulos administradores, como haría un administrador de una organización humana si se ve en la necesidad de tratar de supervisar directamente a 25 subordinados inmediatos. Sin embargo, existe una excepción a esto conocida como centro de transacciones.

Si el módulo administrador toma una sola decisión para invocar a uno solo de sus subordinados, entonces su lógica probablemente es bastante sencilla. En este caso, no nos tenemos que preocupar acerca del alcance de control.

La regla del "*alcance del efecto/alcance del control*" sugiere que cualquier módulo afectado por el resultado de alguna decisión debe ser subordinado del módulo que toma la decisión. Es un tanto análogo a la regla de administración que establece que cualquier empleado afectado por los resultados de la decisión de algún administrador (es decir, dentro del alcance del efecto de la decisión) debe estar dentro del alcance de control del administrador (o sea, trabajando entre la jerarquía de personas que se reportan con el administrador).

Violar esta regla en un ambiente de diseño estructurado usualmente lleva pasos innecesarios de banderas y condiciones (lo cual incrementa el acoplamiento entre módulos), la toma redundante de decisiones o, en el peor de los casos, conexiones patológicas entre módulos.

### **3.12 Diseño de la Base de Datos**

El diseño de la base de datos se lleva a cabo por medio de las siguientes actividades:

- Pasar el modelo E-R a tablas relacionales para producir el diseño inicial.
- Refinar el diseño inicial para producir un diseño completo de la base de datos.

La etapa de diseño de la base de datos produce especificaciones de diseño para una base de datos relacional, incluyendo definiciones para tablas relacionales, índices, vistas y espacio de almacenamiento.

En este punto es necesario documentar cada tabla relacional en un mapa de instancias como el que se muestra en la figura 3.25.

Los tipos válidos de llaves son PK para una columna llave primaria y FK para la columna llave foránea.

Se usan sufijos para distinguir entre múltiples columnas FK en una tabla, por ejemplo, FK1 y FK2. Etiquetar múltiples columnas con el mismo sufijo

Se usa NN para una columna que debe ser definida como NO NULA.

Se usa U para la columna que debe ser única.

	columna 1	columna 2	columna 3	...	columna n
Tipo Llave					
Nulos/Único					
Ejemplo					

Figura 3.25 Mapa de instancias

Si múltiples columnas deben ser únicas en combinación, etiquetarlas con un sufijo, por ejemplo U1.

Se etiqueta una columna sencilla PKO como NN, U.

Se etiquetan múltiples columnas PK (compuestas) como NN, U1 o como NN, U1, U.

Para producir el diseño inicial de la base de datos se llevan a cabo los siguientes pasos:

- Mapear las entidades para las tablas.
- Mapear atributos para columnas y documentar datos simples.
- Mapear identificadores únicos a llaves primarias.
- Mapear relaciones a llaves foráneas.
- Elegir opciones de arco (si las hay).
- Elegir opciones de subtipo (si las hay).

## MAPEAR LAS ENTIDADES

Mapear la tabla para cada entidad implica crear un mapa de instancias para la nueva tabla. Por ejemplo, para una entidad llamada EJEMPLO se creará una tabla llamada EJEMPLO.

El nombre de la tabla debe ser fácil de identificar con el nombre de la entidad. El nombre en plural de una entidad se usa algunas veces porque la tabla debe contener un grupo de renglones.

Una entidad simple no es un subtipo o supertipo.

### **MAPEAR ATRIBUTOS A COLUMNAS**

Se mapea cada atributo de la entidad a una columna en su tabla correspondiente. Se establecen los atributos obligatorios para columnas NO NULAS (NN). El nombre de los atributos deben ser cortos pero significativos.

El nombre de las columnas debe ser fácil de identificar en un modelo E-R.

Es importante prevenir al usuario de no usar las palabras reservadas de SQL para nombres de columnas.

Usar abreviaciones consistentes que no causen confusión al usuario y al programador.

Los nombres de columnas cortos o pequeños reducirán el tiempo requerido para el comando de SQL "parsing".

### **MAPEAR UIDS A LLAVES PRIMARIAS**

Asignar cualquier atributo(s) que sea parte del UID de la entidad a columnas PK, las cuales deben etiquetarse.

Todas las columnas etiquetadas con PK deben etiquetarse también con NN y U.

También se asigna un UID que incluya múltiples atributos a una PK compuesta y se etiquetan estas columnas NN y U1.

Si una entidad incluye una relación, agregar columnas de llaves foráneas para la tabla y señalarlas como parte de la llave primaria.

Se debe escoger un nombre único para cada columna FK, y etiquetar la(s) columna(s) PK, NN y FK.

Si existen múltiples columnas FK en una tabla, usar sufijos para distinguirlos, por ejemplo, FK1 y FK2. Etiquetar múltiples columnas llave con el mismo sufijo.

Las PK compuestas deben ser únicas en combinación y deben ser etiquetadas como U1.

Siempre se deben agregar ejemplos de datos para las columnas FK.

### **MAPEAR RELACIONES PARA LLAVES FORÁNEAS**

Para una relación de entidades M:1 tomar el PK de la tabla (1) y ponerlo en la tabla (M).

Se elige un nombre único para la columna FK y etiquetar la(s) columna(s) FK.

Para las relaciones *debe ser*, se etiqueta la columna como NN y se agregan ejemplos.

Si el PK de la tabla incluye una llave foránea (FK), las columnas FK que soportan la relación, pueden ser agregadas en el mapeo de UIDS a llaves primarias.

Para una relación obligatoria 1:1, se colocan las FK como únicas en el mapa de instancias en donde la relación es obligatoria y se usa el constraint NO NULO para forzar la condición obligatoria.

Si una relación 1:1 es opcional en ambas direcciones mover la FK en la tabla en cualquiera de las dos tablas de la relación. La FK debe ser siempre única, pero permitiendo NULOS.

Para una relación recursiva 1:M se agrega una columna FK a la tabla. Esta columna FK debe referenciar valores de la columna PK.

La columna FK hace referencia a un renglón en la misma tabla y su nombre debe reflejar la relación.

Una FK recursiva nunca debe ser NO NULA.

Para una relación recursiva 1:1 se agrega una FK única a la tabla. Esta columna FK debe referenciar un valor de la columna PK.

La combinación de columnas PK y FK siempre debe ser única para asegurar la relación 1:1. Poniendo el PK y FK como únicos (U) se garantiza que la combinación será única.

## **ESCOGER OPCIONES DE ARCO**

Los Arcos representan un tipo de llave foránea de alternativa múltiple.

Se debe escoger entre las alternativas de diseño de Arco explícito y de Arco genérico para "mapear" arcos a llaves foráneas.

También se utilizan estos dos tipos de diseño para implementar múltiples llaves foráneas cuando un arco atraviesa un conjunto de relaciones 1:1.

Los arcos solamente pueden atravesar el final de las relaciones que son o todas obligatorias o todas opcionales.

El diseño de Arco Explícito crea una columna de llave foránea para cada relación que incluye el arco. También este diseño soportará llaves foráneas múltiples con diferentes formatos. El software de aplicación debe forzar a una relación de exclusividad entre las llaves foráneas.

El diseño de Arco Genérico crea una columna de llave foránea única y una columna para marcar la relación del arco. Como las relaciones son exclusivas, solamente un valor de FK existirá para cada renglón en la tabla.

Si las relaciones debajo del arco son obligatorias, se hacen ambas columnas NO NULAS (NOT NULL). Las llaves foráneas deben compartir el mismo formato para todas las tablas referenciadas. La relación de exclusividad se fuerza automáticamente.

## ESCOGER OPCIONES DE SUBTIPOS

Se debe escoger entre las alternativas de diseño de una sola tabla, diseño de tablas separadas e implementación del Arco para "mapear" subtipos a tablas.

No olvidar que los subtipos heredan todos los atributos del supertipo y sus relaciones, que pueden tener atributos y relaciones propias y, por último, que deben ser mutuamente excluyentes.

En relación al *diseño de subtipo en una sola tabla*, se crea una sola tabla para el supertipo y se mapean los subtipos dentro de esta tabla. La tabla sencilla contendrá instancias de todos los subtipos.

Los pasos para llevar a cabo este diseño son:

- Crear una sola tabla para el supertipo.
- Crear una columna para cada uno de los atributos del supertipo.
- Crear una columna TIPO para identificar a que subtipo pertenece cada renglón.
- Crear una columna para cada uno de los atributos del subtipo.
- Crear columnas FK para cada una de las relaciones del supertipo.
- Crear columnas FK para cada una de las relaciones del subtipo.

El diseño de subtipo de una sola tabla requiere una nueva columna *tipo* para identificar el subtipo al que pertenece cada renglón.

Se debe usar un diseño de subtipo en una sola tabla cuando hay pocas relaciones y atributos propios. Las ventajas que se tienen son que el acceso al supertipo es directo y que el subtipo puede ser accesado y modificado usando vistas. Pero las desventajas son que los requerimientos del subtipo NO NULO no se pueden forzar a nivel de base de datos y que la lógica de las aplicaciones tendrán que manejar diferentes conjuntos de atributos, dependiendo del TIPO.

En cuanto al ***diseño de subtipo en tablas separadas*** se debe mapear una tabla para cada subtipo ya que cada una de ellas contendrá solamente instancias de un subtipo.

Los pasos para realizar este tipo de diseño son:

- Crear una tabla para cada subtipo.
- En cada tabla subtipo, crear columnas para los atributos del subtipo.
- En cada tabla subtipo, crear columnas para los atributos del supertipo.
- En cada tabla subtipo, crear columnas FK para las relaciones del subtipo.
- En cada tabla subtipo, crear columnas FK para las relaciones del supertipo.

Se usa un diseño de subtipo de tablas separadas cuando hay muchas relaciones y atributos de subtipos específicos.

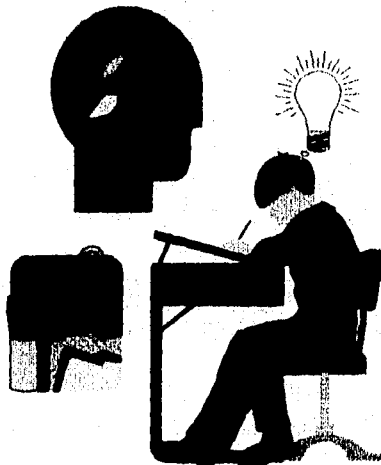
Las ventajas de este diseño son que la opcionalidad de los atributos se fuerza a nivel de la base de datos y la lógica de las aplicaciones no requiere de revisión para los subtipos.

Las desventajas son que la consulta al supertipo requiere de un operador de **UNIÓN**, las vistas que enlazan dos tablas (mediante un join), son solamente de consulta, el código del programa de aplicación debe ser específico para cada tabla individual y el mantenimiento de los UID's de los subtipos es difícil de implementar.

# CAPÍTULO 4

**ANÁLISIS Y DISEÑO DEL SISTEMA DE CONTROL Y  
AUTOMATIZACIÓN DE DOCUMENTOS A GRAN ESCALA  
(SICADE)**

---





## 4.1 DFD-0 Sistema de Control y Automatización de Documentos a gran Escala (SICADE)

De acuerdo a lo tratado en el tercer capítulo, el Diagrama de Flujo de Datos del SICADE tiene como objetivo mostrar en forma gráfica la problemática de la Coordinación Administrativa y también las propuestas tanto del usuario como del analista para mejorar los procedimientos manuales y los automatizados con el fin de delimitar la frontera hombre-máquina. El DFD de Contexto se muestra en la figura 4.1.

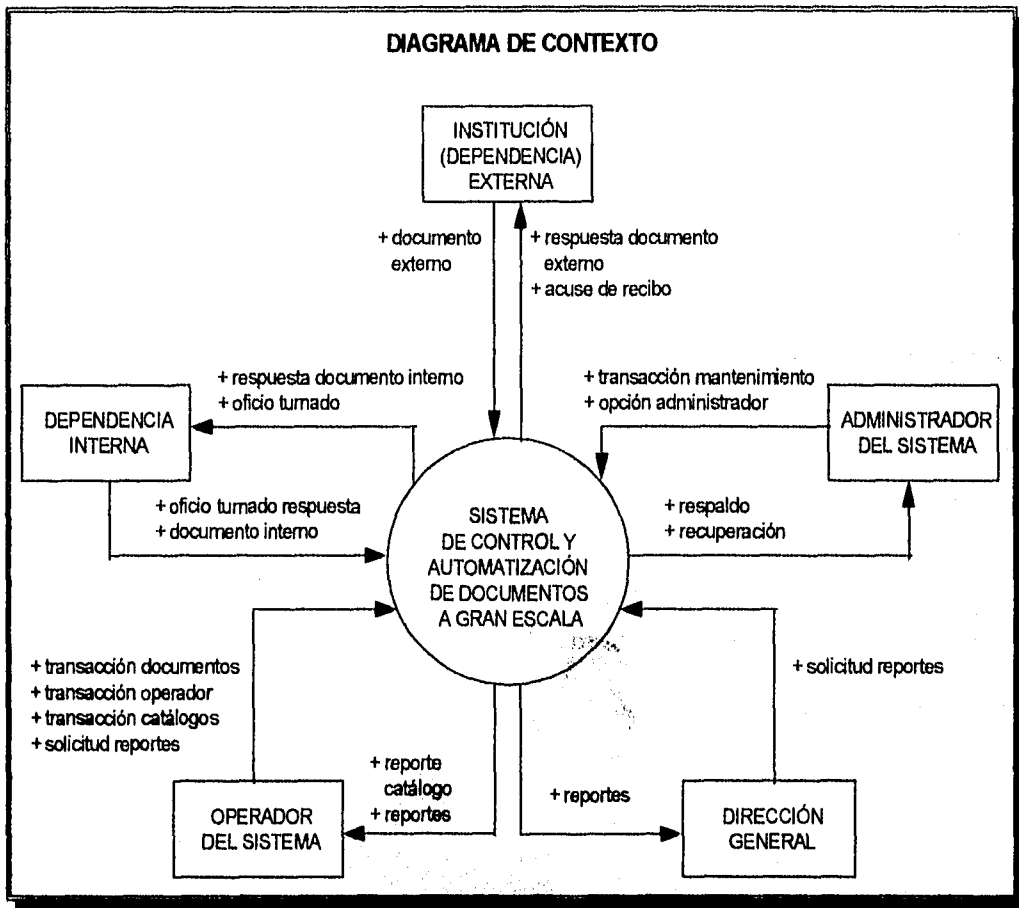


Figura 4.1 Diagrama de Contexto. SICADE.

En este DFD se identifican cinco entidades externas: Institución (dependencia) Externa, Dirección General, Dependencia Interna, Operador del Sistema y Administrador del Sistema. Asimismo se visualizan los flujos de datos que las entidades entregan al sistema y a su vez los que éste les regresa. El detalle de cada flujo aparece en el Diccionario de Datos.

Después de esquematizar las principales entidades y flujos, se procede a general el DFD de nivel 0 (cero), el cual se muestra en la figura 4.2.

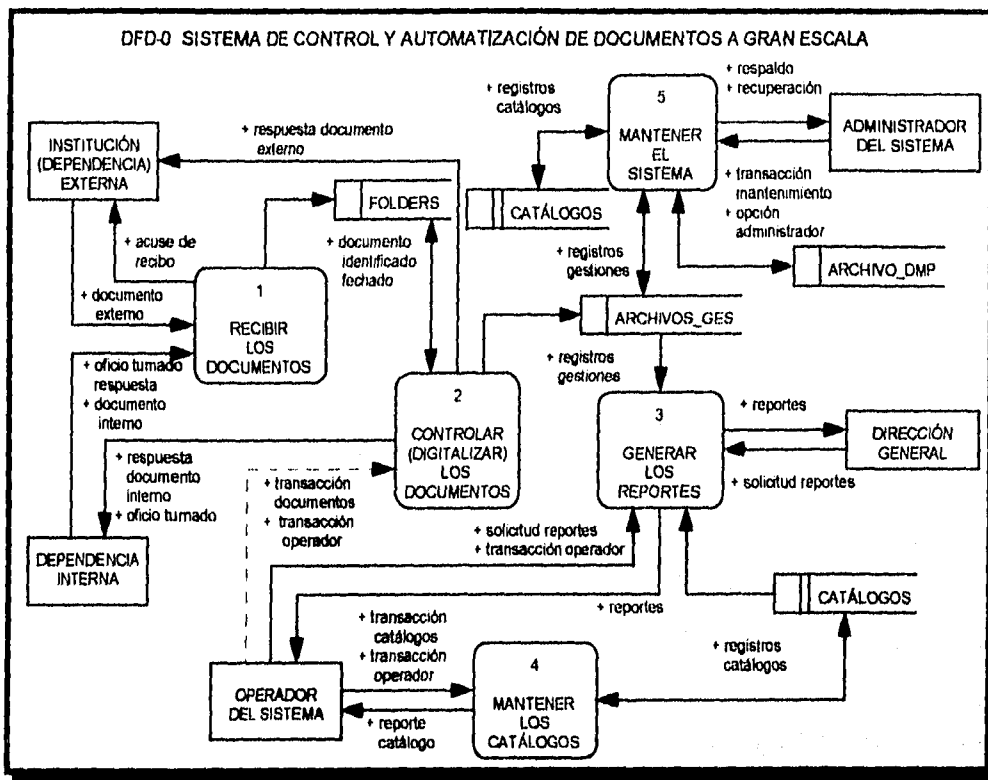


Figura 4.2 Diagrama de Flujo de Datos Nivel 0 de SICADE.

En este diagrama se observan cinco procesos principales: Recibir los documentos, Mantener los catálogos, Controlar (digitalizar) los documentos, Generar los reportes y Mantener el sistema.

También se aprecian tres archivos generales: Folders, Catálogos y Archivos\_ges, los cuales se detallan en el diccionario de datos.

Enseguida se describirá lo relacionado con cada uno de los procesos, es decir, cuales flujos de datos recibe, cuales emite y su significado.

Asimismo cada una de las entidades con las cuales tiene alguna relación.

## 4.2 Recibir los documentos

Lo relacionado con el proceso "Recibir los documentos" se representa en el DFD-1, en la figura 4.3.

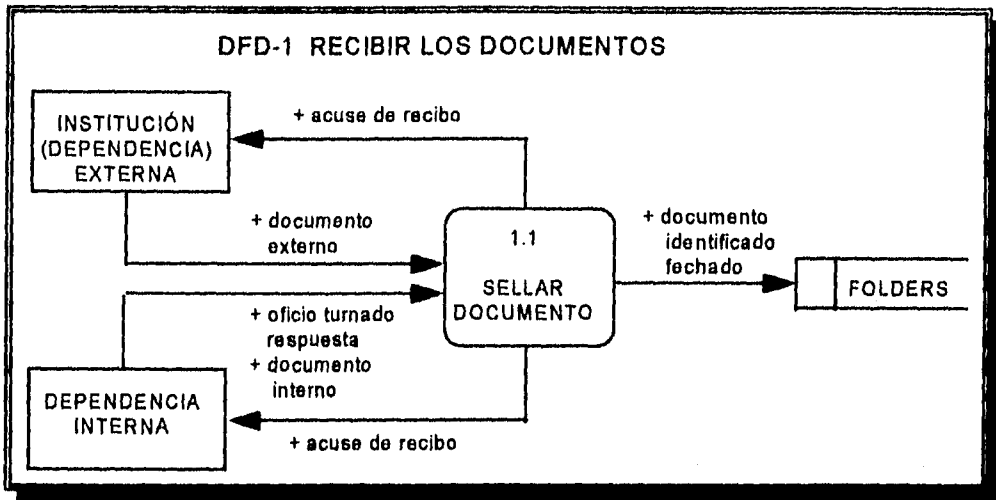


Figura 4.3 DFD-1 Recibir los documentos.

La Institución (dependencia) Externa, en adelante denominada Institución externa, representa a cada una de las empresas públicas o privadas que de alguna forma mantienen una relación con la empresa propietaria de SICADE, en lo sucesivo llamada SHCP (Secretaría de Hacienda y Crédito Público).

Tales Instituciones como la Secretaría de Relaciones Exteriores, Secretaría de Gobernación, ORACLE de México, etc. envían un "documento externo" a la SHCP en el que exponen el asunto que desean sea atendido y resuelto a la brevedad posible.

La Dependencia interna representa a cada una de las Subdirecciones, Gerencias, Áreas, Departamentos, etc. que tienen la facultad y capacidad para resolver un asunto en específico, así como de hacer peticiones propias a otra dependencia interna. Es por ello que esta entidad envía una respuesta para cada asunto que le llega, mediante el "oficio turnado respuesta", al proceso "Recibir los documentos". También envía un "documento interno" en el que describe sus requerimientos.

El proceso "Recibir los documentos" representa un procedimiento manual, en el cual se recibe el "documento externo" por parte de la Institución externa y el "oficio turnado respuesta" y "documento interno" por parte de la Dependencia interna, aplicando un sello de recibido al original y al acuse de recibo.

Este último se devuelve a la Institución Externa o Dependencia interna y el primero, llamado "documento identificado fechado", se guarda en un folder para ser capturado posteriormente. Cabe señalar que al aplicar el sello, se imprime en el documento un número identificador y la fecha de recibido.

### 4.3 Controlar (Digitalizar) los documentos

El procedimiento "Controlar (digitalizar) documentos" tiene relación con las entidades Institución externa, Dependencia interna y con Operador del sistema, tal como se muestra en la figura 4.4.

La entidad Operador del sistema representa a la persona que utiliza o "maneja" el SICADE y está encargado de capturar los documentos, ya sea en forma manual o digitalizada. También actualiza los documentos según el seguimiento que tengan, elimina aquellas gestiones que han sido resultas, registra las transformaciones de los documentos y solicita consultas.

Las actividades citadas anteriormente quedan representadas por la "transacción documentos", mientras que la "transacción operador" engloba una serie de acciones que el Operador del sistema realiza al "Capturar gestión documento", "Actualizar documento", "Efectuar baja de gestión", "Registrar transformación" y "Generar consultas", independientemente de la tecnología de hardware y software que se utilice para implementar dicho sistema.

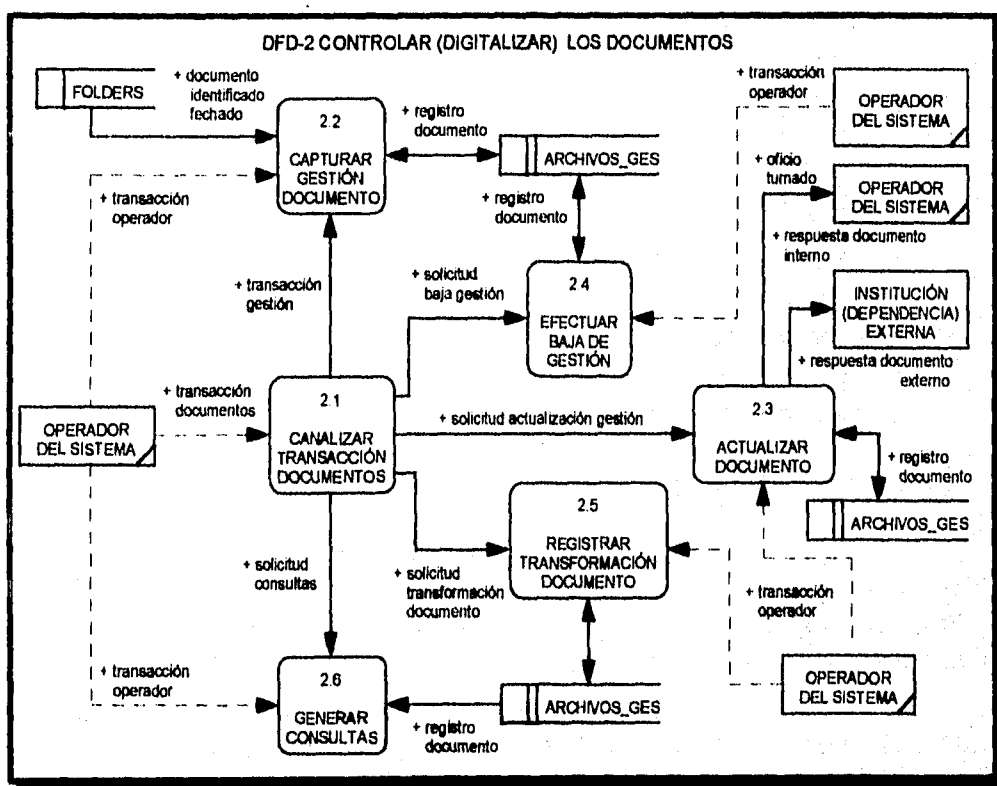


Figura 4.4 DFD-2 Controlar (digitalizar) los documentos.

Tales acciones son las siguientes:

- Limpiar forma de captura
- Limpiar campo
- Limpiar bloque
- Limpiar registro
- Limpiar todo
- Editar texto
- Cortar texto
- Copiar texto
- Pegar texto
- Listar opciones
- Interface de edición de texto
- Salir al sistema operativo
- Regresar al menú anterior
- Guardar captura
- Ir al campo siguiente
- Ir al campo previo
- Definir la consulta
- Ejecutar la consulta
- Duplicar campo
- Duplicar registro
- Ir al bloque siguiente
- Ir al bloque anterior
- Ir al registro siguiente
- Ir al registro previo
- Ir al primer registro
- Ir al último registro
- Aceptar captura
- Generar reportes

etc.

Así, el proceso "*Canalizar transacción documentos*" recibe del Operador del sistema la "*transacción documentos*". La función principal de dicho proceso es canalizar la transacción en forma adecuada al proceso correspondiente, expresado en términos de diseño, representa un menú de opciones.

Si la "*transacción documentos*" se refiere a una "*transacción gestión*" entonces el proceso "*Capturar gestión documento*" recibe dicha canalización para dar paso a la captura del documento especificado. Durante esta captura el Operador del sistema emite su "*transacción operador*" para auxiliarse y facilitar su labor.

Si se trata de una "*solicitud actualización gestión*", en el proceso "*Actualizar documento*" se llevan a cabo todas aquellas modificaciones que sufra un documento durante el seguimiento que se le da. Obviamente llegará el momento en el que se necesite registrar la solución al documento y de esta forma se entregue a la Institución externa la "*respuesta documento externo*".

Si se trata de una "*solicitud baja gestión*", en el proceso "*Efectuar baja de gestión*" se borra o elimina el documento cuyo seguimiento ha finalizado.

Si se trata de una "*solicitud transformación documento*", en el proceso "*Registrar transformación documento*" se lleva a cabo la transformación de la información a un formato que permita la manipulación de los datos que se están solicitando.

Si se trata de una "*solicitud consultas*", en el proceso "*Generar consultas*" se elaboran las consultas de acuerdo al tipo de consulta, como:

- Consultas generales
- Consultas por árbol
- Consultas alfabéticas
- Consultas especiales

Debe notarse que todos los procesos, excepto el primero, leen y escriben en el ARCHIVO\_GES todas aquellos documentos a los que se les da seguimiento.

#### **4.4 Generar los reportes**

El diagrama de flujo de datos de nivel uno del proceso "*Generar los reportes*" de la figura 4.2 se presenta en las figuras 4.5 y 4.6 debido a la cantidad de procesos que considera este proceso.

En ambas figuras, tanto en la figura 4.5 como en la 4.6 se observa que para cada proceso, las entidades Operador del sistema y Dirección general emiten sus peticiones a través del flujo de datos "*solicitud reportes*", el cual representa cualquiera de las siguientes opciones:

- Solicitud reporte turnados
- Solicitud reporte documentos turnados
- Solicitud reporte discursos
- Solicitud reporte viajes
- Solicitud reporte oficios
- Solicitud reporte documentos para firma
- Solicitud reporte decretos
- Solicitud reporte catálogo alfabético
- Solicitud reporte invitaciones
- Solicitud reporte estadísticas

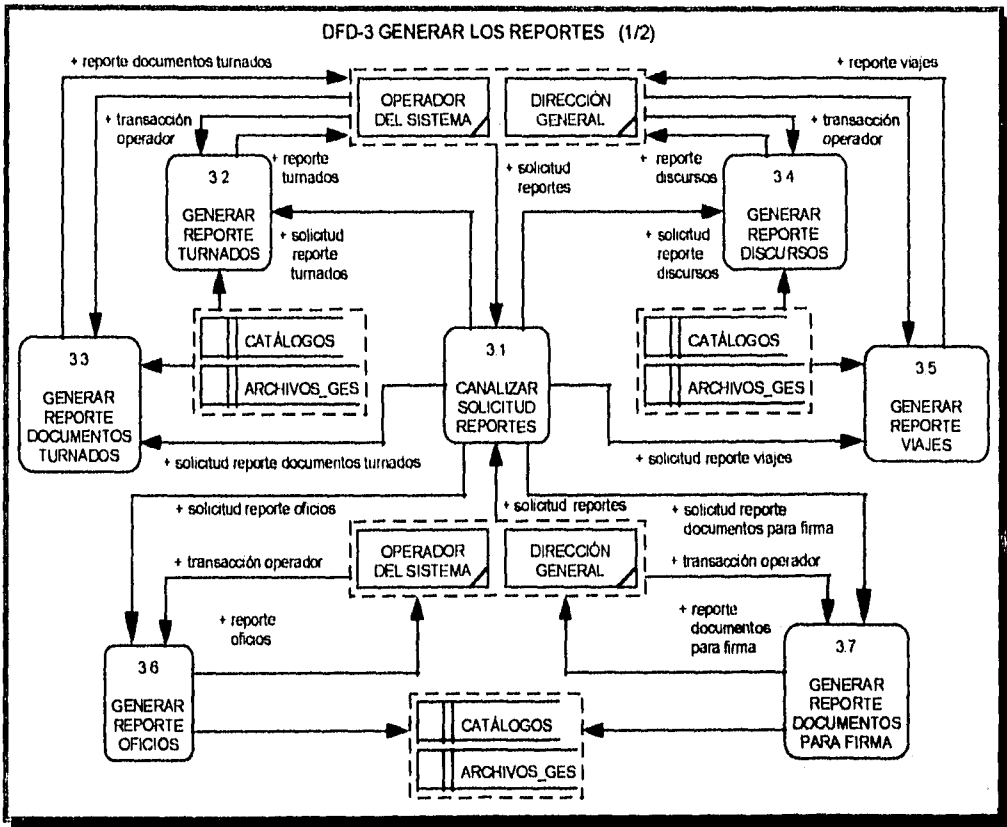


Figura 4.5 DFD-3 Generar los reportes (1/2)

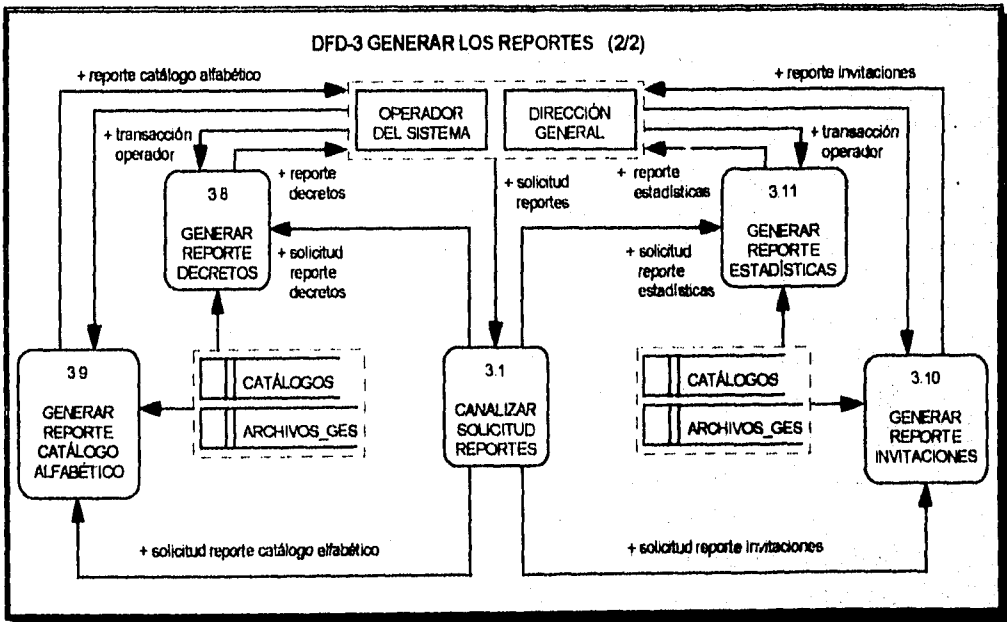


Figura 4.6 DFD-4 Generar los reportes (2/2)

Cabe señalar que la cantidad de reportes depende directamente de los clientes que manifiestan sus requerimientos.

También se debe advertir que el flujo de datos "transacción operador" se envía a cada proceso debido a que tanto el Operador del sistema como la Dirección general realizarán acciones similares a las que se mencionaron para la "transacción operador" en el DFD-2 figura 4.4.

Los archivos con los que se elaboran los reportes son Catálogos y Archivos\_ges.

Asimismo, cada proceso emite el reporte correspondiente a las entidades participantes en el DFD-3, tales reportes son:

- Reporte turnados
- Reporte documentos turnados
- Reporte discursos
- Reporte viajes
- Reporte oficios
- Reporte documentos para firma
- Reporte decretos
- Reporte catálogo alfabético
- Reporte invitaciones
- Reporte estadísticas

## **4.5 Mantener los catálogos**

El proceso "Mantener los catálogos" esta relacionado únicamente con la entidad Operador del sistema, tal como se muestra en la figura 4.7.

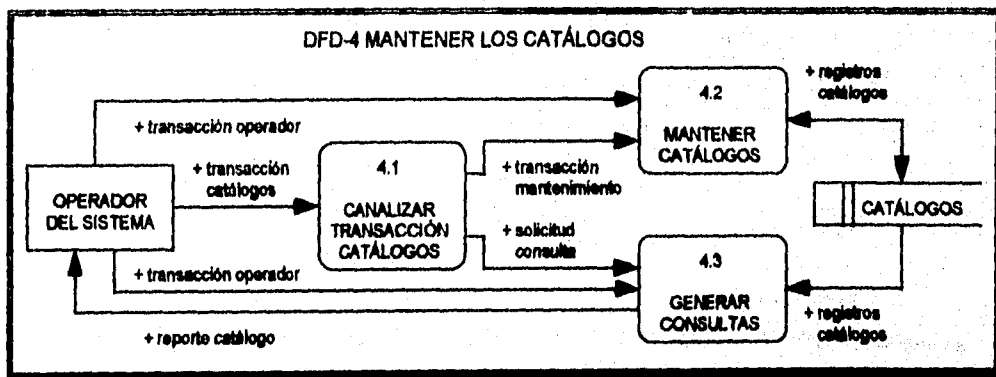


Figura 4.7 DFD-4 Mantener los catálogos

El DFD-4 en realidad es sumamente sencillo, ya que la entidad Operador del sistema mediante el flujo de datos "transacción catálogos" manifiesta tan solo dos requerimientos: "transacción mantenimiento" y "solicitud consulta"



El primero abarca las tres típicas acciones: alta, baja y modificación de registros, mientras que el segundo se refiere a consultar los siguientes catálogos:

- Clasificación
- Documentos
- Instrucciones
- Temas
- Turnados
- Expedientes
- Dependencias

Según la figura 4.7 el proceso "Mantener catálogos" recibe el flujo de datos "transacción mantenimiento" del procesos "Canalizar transacción catálogos" para posteriormente actualizar, mediante el flujo "registros catálogos", el archivo Catálogos que representa a los citados anteriormente.

El proceso "Generar consultas" recibe el flujo "solicitud consulta" para así generar un "reporte catálogo". Hay que notar que este proceso únicamente hace lectura del archivo Catálogos.

## 4.6 Mantener el Sistema

El DFD-5 Mantener el sistema se presenta en la figura 4.8.

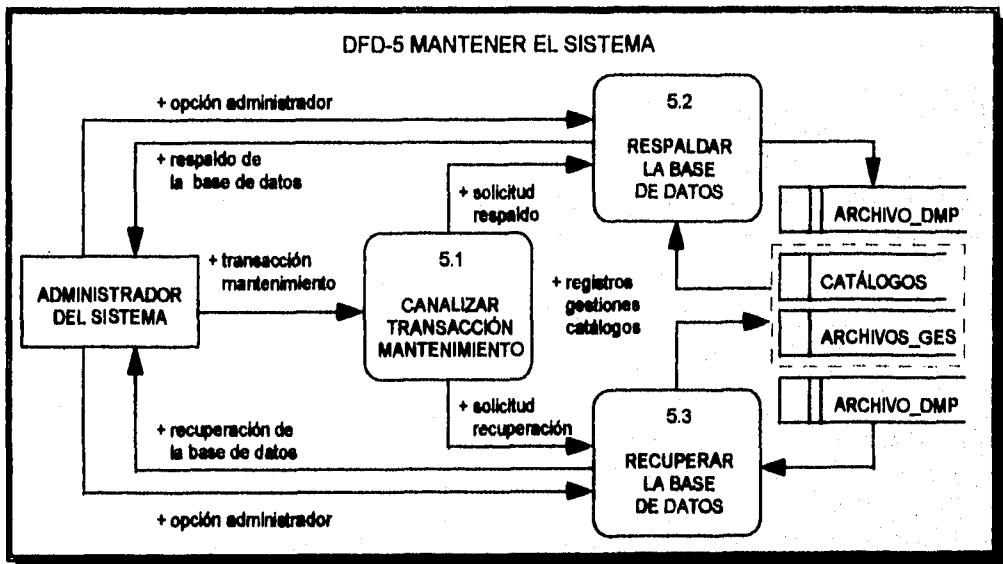


Figura 4.8 DFD-5 Mantener el sistema

En dicha figura se observa que el Administrador del sistema envía el flujo de datos "*transacción mantenimiento*" al proceso "*Canalizar transacción mantenimiento*", el cual por su parte envía la transacción al proceso correspondiente. Esto quiere decir que el flujo "*transacción mantenimiento*" esta compuesto por "*solicitud respaldo*" y "*solicitud recuperación*".

Si la "*transacción mantenimiento*" representa una "*solicitud respaldo*", el proceso "*Respalda la base de datos*" recibe tal flujo, para empezar a generar el respaldo con las opciones que ha dado el Administrador del sistemas mediante el flujo "*opciones administrador*".

Si la "*transacción mantenimiento*" representa una "*solicitud recuperación*", el proceso "*Recuperar la base de datos*" recibe tal flujo, para empezar a generar la recuperación con las opciones que ha dado el Administrador del sistemas mediante el flujo "*opciones administrador*".

Tales opciones al realizar el "export" o "import" son, por ejemplo, *export file, tables, owner, full database, grants, rows, constraints, indexes, etc.*

Cuando se realiza un export o respaldo se genera un ARCHIVO.DMP, el cual se utiliza para realizar el import o recuperación. Es evidente que los procesos leen los archivos Catálogos y Archivos\_ges cuando se hace un export, pero escriben en ellos cuando se hace un import.

## **4.7 Diseño de la Base de Datos**

Para dar paso al diseño de la base de datos, es indispensable establecer las relaciones existentes entre las entidades, ya que así se determina la opcionalidad y el grado entre ellas.

### **GESGRAL y DOCUMENTO**

- Cada GESGRAL debe ser escrita en uno y solamente un DOCUMENTO
- Cada DOCUMENTO debe contener una y solamente una GESGRAL

### **GESGRAL y CLASIF**

- Cada GESGRAL debe ser clasificada por una y solamente una CLASIF
- Cada CLASIF debe ser asignada a una o más GESGRAL

### **GESGRAL y DEPENDENCIA**

- Cada DEPENDENCIA debe tramitar una y solamente una GESGRAL
- Cada GESGRAL debe ser tramitada por una o más DEPENDENCIAS

### **GESGRAL y GESRES**

- Cada GESRES debe ser resolución de una y solamente una GESGRAL
- Cada GESGRAL debe tener asignada una o más GESRES

### **GESGRAL y GESREFALF**

- Cada GESREFALF debe ser referencia de una y solamente una GESGRAL
- Cada GESGRAL debe ser referida por una o más GESREFALF

### **GESGRAL y GESREM**

- Cada GESREM debe ser remitente de una y solamente una GESGRAL
- Cada GESGRAL debe ser solicitada por uno o más GESREM

### **GESGRAL y GESTEM**

- Cada GESTEM debe ser tema de una y solamente una GESGRAL
- Cada GESGRAL debe tener asignado uno o más GESTEM

### **GESGRAL y GESTURNO**

- Cada GESTURNO debe ser turnado de una y solamente una GESGRAL
- Cada GESGRAL debe estar asignada a uno o más GESTURNOS

### **GESGRAL y GESEXP**

- Cada GESEXP debe ser expediente de una y solamente una GESGRAL
- Cada GESGRAL debe ser asignada a uno o más GESEXP

### **GESTEM y TEMA**

- Cada GESTEM debe tener asignado uno y solamente un TEMA
- Cada TEMA debe ser asignado a uno o más GESTEM

### **GESTURNO y TURNO**

- Cada GESTURNO debe tener asignado uno y solamente un TURNO
- Cada TURNO debe ser asignado a uno o más GESTURNO

### **GESTURNO e INSTRUC**

- Cada GESTURNO debe tener asignada una y solamente una INSTRUC
- Cada INSTRUC debe ser asignada a uno o más GESTURNO

### **GESEXP y EXPEDIENTE**

- Cada GESEXP debe tener asignado uno y solamente un EXPEDIENTE
- Cada EXPEDIENTE debe ser asignado a uno o más GESEXP

### 4.8 Modelo Entidad Relación

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

El modelo Entidad-relación se muestra en la figura 4.9.

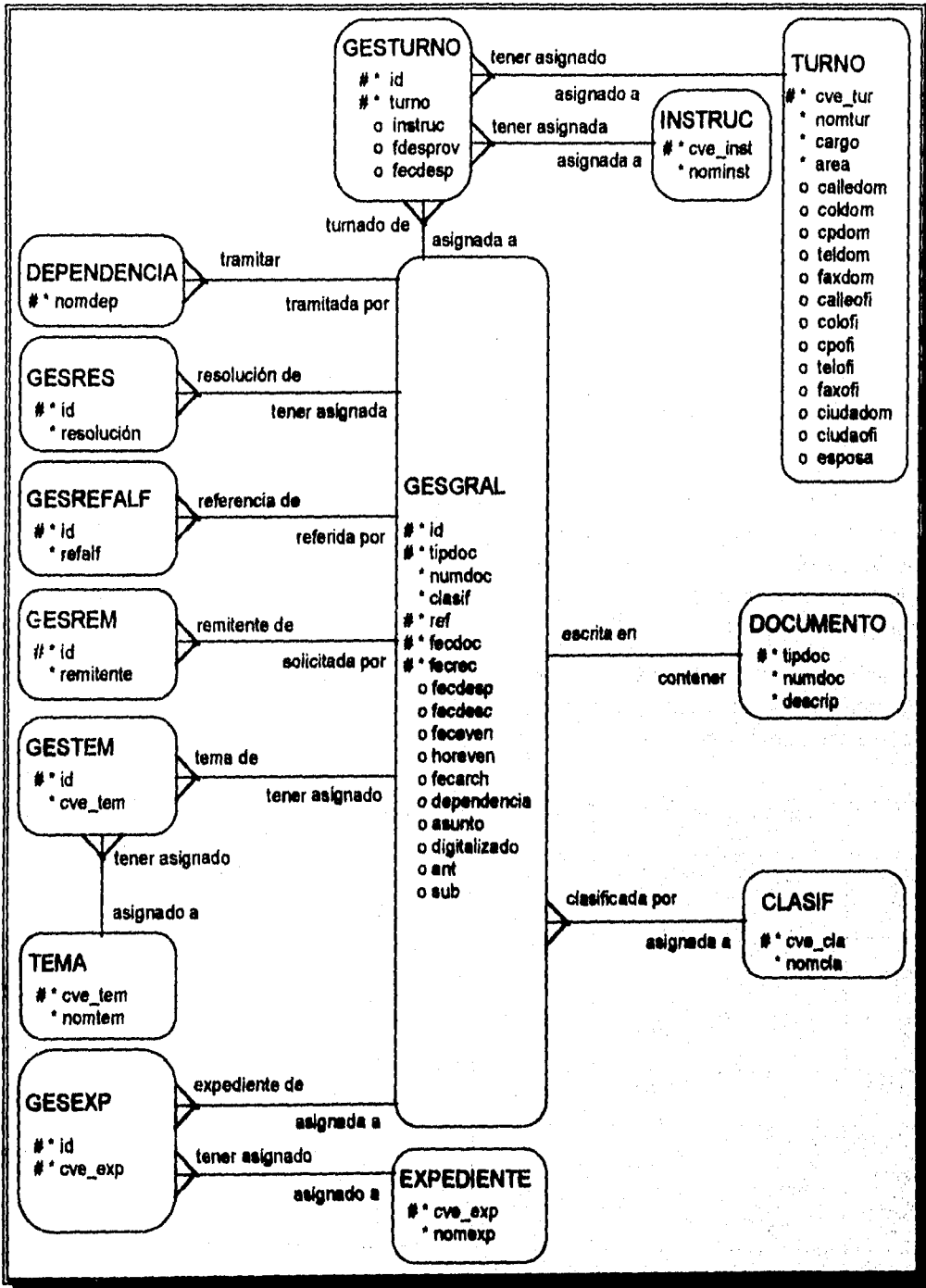


Figura 4.9 Modelo Entidad Relación.

### MATRIZ DE RELACIÓN

La matriz de relación se dividió en cuadrantes por razones de espacio, tal como lo muestra la figura 4.10:

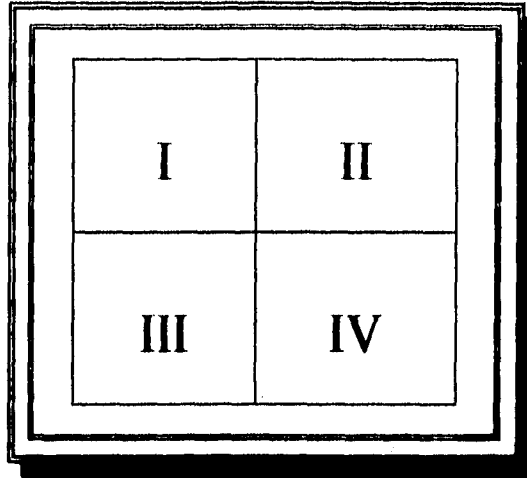


Figura 4.10 Cuadrantes de la Matriz de Relación

La tabla 4.1 representa el primer cuadrante de la Matriz de Relación. En ella se aprecian las relaciones que tiene la entidad GESGRAL con GESTURNO, DOCUMENTO, CLASIF y DEPENDENCIA, es decir, se ha interpretado el primer renglón. Obviamente GESGRAL no tiene relación alguna consigo misma, tampoco con INSTRUC y TURNO.

Asimismo, se observa la relación entre GESTURNO y GESGRAL, entre DOCUMENTO y GESGRAL, etc. Todo depende de la forma en que se lea, ya sea por renglón o columna.

	GESGRAL	GESTURNO	INSTRUC	TURNO	DOCUM	CLASIF	DEPENDENCIA
GESGRAL	—	asignada a	—	—	escrita en	clasificada por	tramitada por
GESTURNO	tornado de	—	tener asignada	tener asignado	—	—	—
INSTRUC	—	asignada a	—	—	—	—	—
TURNO	—	asignado a	—	—	—	—	—
DOCUM	contener	—	—	—	—	—	—
CLASIF	asignada a	—	—	—	—	—	—
DEPEN	tramitar	—	—	—	—	—	—

Tabla 4.1 Cuadrante I de la Matriz de Relación.

En la tabla 4.2 podemos ver que al continuar interpretando el primer renglón, las entidades GESRES, GESREFALF, GESREM, GESTEM y GESEXP mantienen relación con GESGRAL.

	GESRES	GESREFALF	GESREM	GESTEM	TEMA	GESEXP	EXPED
GESGRAL	tener asignada	referida por	solicitada por	tener asignado	—	asignada a	—
GESTURNO	—	—	—	—	—	—	—
INSTRUC	—	—	—	—	—	—	—
TURNO	—	—	—	—	—	—	—
DOCUM	—	—	—	—	—	—	—
CLASIF	—	—	—	—	—	—	—
DEPEN	—	—	—	—	—	—	—

Tabla 4.2 Cuadrante II de la Matriz de Relación.

Ahora en la tabla 4.3 se interpreta la primera columna en donde es fácil determinar que GESRES, GESREFALF, GESREM, GESTEM y GESEXP tienen relación únicamente con GESGRAL.

	GESGRAL	GESTURNO	INSTRUC	TURNO	DOCUM	CLASIF	DEPEN
GESRES	resolución de	—	—	—	—	—	—
GESREFALF	referencia de	—	—	—	—	—	—
GESREM	remitente de	—	—	—	—	—	—
GESTEM	tema de	—	—	—	—	—	—
TEMA	—	—	—	—	—	—	—
GESEXP	expediente	—	—	—	—	—	—
EXPED	—	—	—	—	—	—	—

Tabla 4.3 Cuadrante III de la Matriz de Relación.

En el último cuadrante, se observa que al especificar la relación que tiene GESTEM con TEMA desde el punto de vista renglón, es "tener asignado". En cambio desde el punto de vista columna, la relación es "asignado a".

De forma similar, la relación entre GESEXP y EXPED desde el punto de vista renglón es "tener asignado" y desde el punto de vista columna es "asignado a".

El último cuadrante, está esquematizado en la tabla 4.4

	GESRES	GES-REFALF	GESREM	GESTEM	TEMA	GESEXP	EXPED
GESRES	—	—	—	—	—	—	—
GESREFALF	—	—	—	—	—	—	—
GESREM	—	—	—	—	—	—	—
GESTEM	—	—	—	—	tener asignado	—	—
TEMA	—	—	—	asignado a	—	—	—
GESEXP	—	—	—	—	—	—	tener asignado
EXPED	—	—	—	—	—	asignado a	—

Tabla 4.4 Cuadrante IV de la Matriz de Relación.

## 4.9 El Diccionario de Datos del SICADE

En seguida se presenta como ejemplo, únicamente el diccionario de datos del Diagrama de Contexto (figura 4.1), ya que un listado completo resultaría muy extenso.

**Nombre del flujo:** documento externo  
**Composición:** = (oficio | gestión | regalo | invitación | discurso)  
**Descripción:** \*Documento que la Institución Externa manda al SICADE\*  
**Origen:** Institución Externa  
**Destino:** SICADE

**Nombre del flujo:** respuesta documento externo  
**Composición:** = oficio  
**Descripción:** \*Documento que contiene la respuesta que satisface la petición de la Institución Externa\*  
**Origen:** SICADE  
**Destino:** Institución Externa

**Nombre del flujo:** acuse de recibo  
**Composición:** = (copia oficio | copia gestión | copia regalo | copia invitación | copia discurso)  
**Descripción:** \*Copia del documento que la Institución Externa envía al SICADE. Esta copia tiene un sello de recibido con la fecha.  
**Origen:** SICADE  
**Destino:** Institución Externa

**Nombre del flujo:** transacción mantenimiento  
**Composición:** = (solicitud respaldo | solicitud recuperación)  
**Descripción:** \*Se refiere al respaldo o recuperación de la información del SICADE\*  
**Origen:** Administrador del Sistema  
**Destino:** SICADE

<b>Nombre del flujo:</b>	opción administrador
<b>Composición:</b>	= (username+password+database+export file+(browse)+(tables owners  full database)+(specify)+(grants rows constraints indexes)+(advanced)+(record length buffer size record export in system data compress when imported  keep components consistent increment type statistics type generate log file log file)+(import file+import mode)
<b>Descripción:</b>	*Características u opciones con las que se define un respaldo o una recuperación de información del SICADE*
<b>Origen:</b>	Administrador del sistema
<b>Destino:</b>	SICADE
<b>Nombre del flujo:</b>	respaldo
<b>Composición:</b>	= archivo.dmp
<b>Descripción:</b>	*Archivo que contiene la información respaldada.*
<b>Origen:</b>	SICADE
<b>Destino:</b>	Administrador del sistema
<b>Nombre del flujo:</b>	recuperación
<b>Composición:</b>	= (tablas+usuarios+índices)
<b>Descripción:</b>	*Es la información del SICADE contenida en tablas, usuarios e índices.*
<b>Origen:</b>	SICADE
<b>Destino:</b>	Administrador del sistema
<b>Nombre del flujo:</b>	solicitud reportes
<b>Composición:</b>	= (solicitud reporte tumados solicitud reporte documentos tumados solicitud reporte discursos solicitud reporte viajes solicitud reporte oficinas solicitud reporte documentos para firma solicitud reporte decretos solicitud reporte catálogo alfabético solicitud reporte invitaciones solicitud reporte estadísticas)
<b>Descripción:</b>	*Es el medio escrito o hablado por el cual se solicita la generación de algún reporte.*
<b>Origen:</b>	Dirección General
<b>Origen:</b>	Operador del sistema
<b>Destino:</b>	SICADE
<b>Nombre del flujo:</b>	reportes
<b>Composición:</b>	= (reporte tumados reporte documentos tumados reporte discursos reporte viajes reporte oficinas reporte documentos para firma reporte decretos  reporte catálogo alfabético reporte invitaciones reporte estadísticas)
<b>Descripción:</b>	*Documentos que reportan la información clasificada de las entidades del SICADE*
<b>Origen:</b>	SICADE
<b>Destino:</b>	Dirección General
<b>Destino:</b>	Operador del sistema
<b>Nombre del flujo:</b>	reporte catálogo
<b>Composición:</b>	= (reporte catálogo clasificación reporte catálogo documentos reporte catálogo instrucciones reporte catálogo temas reporte catálogo tumados  reporte catálogo expedientes reporte catálogo dependencias)
<b>Descripción:</b>	*Listados de los catálogos que contienen las clasificaciones, tipos de documentos, instrucciones, temas, tumados, expedientes y dependencias.*
<b>Origen:</b>	SICADE
<b>Destino:</b>	Operador del sistema



**Nombre del flujo:** transacción catálogos  
**Composición:** = (transacción mantenimiento|solicitud consulta)  
**Descripción:** \*Se refiere a la petición de mantenimiento de algún catálogo, o bien, de alguna consulta.\*  
**Origen:** Operador del sistema  
**Destino:** SICADE

**Nombre del flujo:** transacción operador  
**Composición:** = (Limpiar forma|limpiar campo|limpiar bloque|limpiar registro|editar texto| cortar texto|copiar texto|pegar texto|listar opciones|interface de edición de texto|salir al sistema operativo|regresar al menú anterior|guardar captura| ir al campo siguiente|ir al campo previo| definir la consulta|ejecutar la consulta|duplicar campo|duplicar registro|ir al bloque siguiente|ir al bloque anterior|ir al registro siguiente|ir al registro previo|ir al registro previo|ir al primer registro|ir al último registro|aceptar captura|generar reportes)  
**Descripción:** \*Representa una serie de herramientas que facilitan la captura o edición de los datos.\*  
**Origen:** Operador del sistema.  
**Destino:** SICADE

**Nombre del flujo:** transacción documentos  
**Composición:** =(transacción gestión|solicitud baja gestión|solicitud actualización gestión| solicitud transformación documento|solicitud consultas)  
**Descripción:** \*Son la operaciones que se realizan en los documentos o gestiones.\*  
**Origen:** Operador del sistema.  
**Destino:** SICADE

**Nombre del flujo:** documento interno  
**Composición:** = (oficio | gestión | regalo | invitación | discurso)  
**Descripción:** \*Documento que la Institución Interna manda al SICADE\*  
**Origen:** Dependencia Interna  
**Destino:** SICADE

**Nombre del flujo:** oficio tomado  
**Composición:** = (oficio | gestión | regalo | invitación | discurso)  
**Descripción:** \*Documento que SICADE asigna a una Dependencia interna para su seguimiento o solución.\*  
**Origen:** SICADE  
**Destino:** Dependencia interna

**Nombre del flujo:** oficio tomado respuesta  
**Composición:** = (oficio | gestión | regalo | invitación | discurso)  
**Descripción:** \*Documento que la Dependencia interna envía al SICADE para actualizar información.\*  
**Origen:** Dependencia interna  
**Destino:** SICADE

**Nombre del flujo:** respuesta documento interno  
**Composición:** = (oficio | gestión | regalo | invitación | discurso)  
**Descripción:** \*Es la contestación del SICADE al documento interno.\*  
**Origen:** SICADE  
**Destino:** Dependencia interna

## 4.10 El Diccionario de la Base de Datos del SICADE

También se presenta el diccionario de la base de datos con el fin de ilustrar a detalle cómo esta compuesta la información de las entidades del sistema.

<b>Clasif</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_CLA	VARCHAR2(2)	Clave de clasificación
NOMCLA	VARCHAR2(15)	Nombre de clasificación

<b>Dependencia</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
NOMDEP	VARCHAR2(50)	Nombre de la dependencia

<b>Documento</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
TIPDOC	VARCHAR2(1)	Tipo de documento
NUMDOC	NUMBER(6)	Numero de documento
DESCRIP	VARCHAR2(20)	Descripción

<b>Expediente</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_EXP	VARCHAR2(15)	Clave de expediente
NOMEXP	VARCHAR2(50)	Nombre de expediente

<b>Instruc</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_INST	VARCHAR2(1)	Clave de instrucción
NOMINST	VARCHAR2(45)	Descripción de la instrucción

<b>Tema</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_TEM	VARCHAR2(4)	Clave del Nombre del Tema
NOMTEM	VARCHAR2(45)	Nombre del tema

<b>Turno</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
CVE_TUR	VARCHAR2(4)	Clave del tumado
NOMTUR	VARCHAR2(45)	Nombre del tumado
CARGO	VARCHAR2(45)	Cargo
AREA	VARCHAR2(10)	Área
CALLEDOM	VARCHAR2(100)	Calle del domicilio
COLDOM	VARCHAR2(50)	Colonia del domicilio
CPDOM	VARCHAR2(5)	Código postal del domicilio
TELDOM	VARCHAR2(50)	Teléfono del domicilio
FAXDOM	VARCHAR2(50)	Fax del domicilio
CALLEOFI	VARCHAR2(100)	Calle de la oficina
COLOFI	VARCHAR2(50)	Colonia de la oficina
CPOFI	VARCHAR2(5)	Código postal de la oficina
TELOFI	VARCHAR2(100)	Teléfono de la oficina
FAXOFI	VARCHAR2(100)	Fax de la oficina
CIUDADOM	VARCHAR2(25)	Ciudad del domicilio
CIUDAOFI	VARCHAR2(25)	Ciudad de la oficina
ESPOSA	VARCHAR2(45)	Esposa

<b>GesGral(ID)</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
ID	NUMBER(10)	Numero de identificador
TIPDOC	VARCHAR2(1)	Tipo de documento
NUMDOC	NUMBER(6)	Numero de documento
CLASIF	VARCHAR2(2)	Clasificación
REF	VARCHAR2(15)	Referencia del documento
FECDOC	DATE	Fecha del documento
FECREC	DATE	Fecha de recepción
FECDESP	DATE	Fecha de despacho
FECDESC	DATE	Fecha de descargo
FECEVEN	DATE	Fecha del evento
HOREVEN	VARCHAR2(5)	Hora del evento
FECARCH	DATE	Fecha de archivo
NOMDEP	VARCHAR2(50)	Clave de la Dependencia
ASUNTO	LONG	Asunto
DIGITALIZADO	VARCHAR2(1)	Digitalizado
ANT	NUMBER(10)	Antecedente
SUB	NUMBER(10)	Subsecuente

<b>Gesexp(ID)</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de Identificador
CVE_EXP	VARCHAR2(15)	Clave del Expediente

<b>GesRefAlf</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
REFALF	VARCHAR2(45)	Referencia alfabética

<b>GesRem</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
REMITENTE	VARCHAR2(45)	Remitente

<b>GesRes</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
RESOLUCIÓN	LONG	Resolución final del documento

<b>GesTem</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
CVE_TEM	VARCHAR2(4)	Clave del Nombre del Tema

<b>GesTurno</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
TURN0	VARCHAR2(4)	Turno
INSTRUC	VARCHAR2(1)	Instrucción
FDESPROV	DATE	Fecha de despacho
DESCARGO	DATE	Fecha de descargo

# CAPÍTULO 5

**MANUAL DE CAPACITACIÓN DEL SISTEMA DE  
CONTROL Y AUTOMATIZACIÓN DE DOCUMENTOS A  
GRAN ESCALA (SICADE)**

---



Este manual permite tener, una descripción detallada del uso y la funcionalidad de SICADE. El objetivo principal es explicar los procesos que involucran las opciones del sistema, así como mantener una estricta política de operación para lograr una completa homogeneidad de información por parte del usuario; del cual depende el seguimiento, asignación y terminación de los documentos u oficios a través del tiempo.

Es importante recalcar, que este manual permite al usuario apoyarse, de manera operativa, en el control y la automatización de información, ayudándolo en la toma de decisiones por medio de sugerencias y descripciones de los principales mecanismos del SICADE.

La descripción de todas las opciones del sistema permiten observar la descripción total de los parámetros requeridos, mostrando las posibles opciones a seleccionar para poder insertar los valores correctos, además de puntualizar los posibles errores que se generan, cuando estas reglas o directrices son violadas. El usuario es el encargado de ingresar información al sistema, por lo que es recomendable que primero lea este manual, ya que le servirá para poder entender y asimilar las políticas de seguimiento e incorporarse al ambiente operativo del SICADE.

El Sistema de Control y Automatización de Documentos a gran Escala, es un sistema totalmente enfocado a controlar y administrar tanto la evolución como el seguimiento de las diferentes etapas que puede tomar un documento u oficio. Un documento u oficio es el medio de recepción de peticiones, confirmaciones, invitaciones, obsequios, decretos, documentaciones y gestiones los cuales van dirigidos hacia una persona, la cual no puede determinar la solución de todos estos documentos de una manera eficiente, generando que se pierdan secuencias y sobre todo, la objetividad para la cual se realizan los documentos. Es así que esta persona debe delegar la responsabilidad de solucionar los documentos u oficios hacia otras personas, sin perder el control de las determinaciones tomadas para la solución de los documentos delegados.

Por esto SICADE a sido desarrollado con la finalidad de tener al mayor detalle de los acontecimientos de los documentos recibidos a través del tiempo, generando la posibilidad de saber del estado actual de los documentos en cualquier momento, además de una serie de reportes orientados a detectar las situaciones críticas, tales como documentos no solucionados o contestados, fechas de elaboración del documento, recepción, despacho, descargo y archivo.

Además permite enlazar documentos u oficios que hacen mención o referencia de otros documentos recibidos con anterioridad, en virtud de no tener una solución satisfactoriamente finalizada.

El obtener la relación de las personas relacionados con cada documento también es importante, ya que con la relación de los documentos sin conclusión se podrá obtener un panorama general de la toma de decisiones que se deben realizar para poder solventar con mayor precisión todos los documentos u oficios satisfactoriamente.

Es así que existen tres estados de la información que manipula SICADE, Recepción, Seguimiento y Resolución; esto es en función del tiempo, debido a las necesidades y facultades que deben tomar los documentos y turnados, respectivamente. Los turnados, son las personas a los cuales se les delega la capacidad de respuesta hacia los documentos, según la funcionalidad que desempeñen dentro y fuera de los ambientes que involucran las aplicaciones de SICADE en un entorno de tomas de decisión ejecutivas.

## **5.1 Descripción de los Estados o Etapas de la Información**

- **Recepción de la Documentación Inicial**

Es la parte primaria y elemental de la información, donde SICADE recibe su alimentación. En esta parte la recepción de información debe incorporarse al sistema de una manera minuciosa, detallando el interés u objetivo del documento por medio de los parámetros de recolección tales como fechas, relevancia y tipos de los documentos, objetivos, temas y remitentes.

- **Seguimiento y Asignación de Documentos**

El poder asignar los documentos hacia un turnado, una vez estudiado los objetivos que pretenden buscar los documentos. Permiten decidir por los turnados competentes con referencia hacia el tema de los documentos.

- **Resolución Final de los Documentos**

Es la determinación de la solución final de los documentos, los cuales pasan de un estatus de documento turnado a documento con resolución. El cual puede y debe ser accesado posteriormente para futuras implicaciones o referencias con el asunto determinado.

## **5.2 Descripción de la Funcionalidad del Sistema**

Se describen a detalle las funciones de SICADE a partir de las opciones (menús) que proporciona el sistema. La figura 5.1 es el menú de la pantalla general el cual permite el acceso a todas las funciones, procedimientos y reportes del sistema. Una vez que se han ingresado las palabras asignadas para el acceso al sistema.

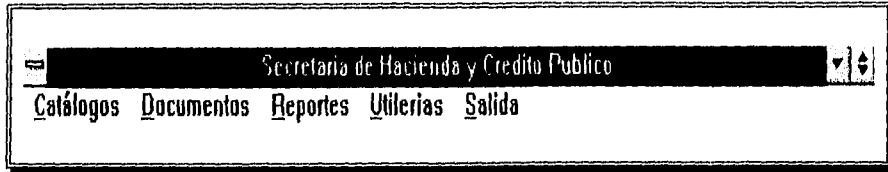


Figura 5.1 Menú General

Es importante mencionar que la opción de Catálogos como la de Documentos muestran las mismas opciones, aunque realizan diferentes tareas. Únicamente resaltaremos los puntos mas importantes de diferencia entre ambos grupos de opciones o pantalla, para evitar la duplicidad de términos y funciones.

La primera opción es Catálogos, la cual se encuentra subdividida en Altas, Bajas , Cambios y Consultas. La primera opción únicamente permitirá realizar modificaciones a los catálogos de clasificaciones de documentos, documentos, instrucciones, temas, turnados, expedientes y dependencias donde :

**Catálogos.** Son esquemas de información detallada y organizada con la finalidad de agrupar documentos con las mismas características, para poder brindar al agente una fácil clasificación de cualquier documento nuevo; que se desee ingresar, coadyuvando a complementar la descripción de la información. En la figura 5.2 se muestran las opciones del primer menú asignado a Catálogos.

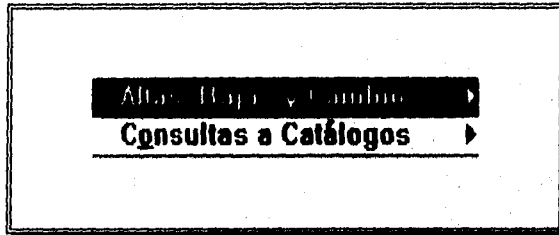


Figura 5.2 Menú de Catálogos

**Alta, Bajas y Cambios de Catálogos.** Actualizaciones de la información de los catálogos. Se puede ingresar una nueva forma de documento que reúna las características donde describan a un grupo de estos, también puede modificar y eliminar las características dependiendo de los criterios tomados; es importante mencionar que cuando es eliminado un tipo de catálogo, los documentos que están asignados a este catálogo, primeramente deben ser reasignados a otro catálogo y después eliminar el catálogo. Esto puede generar inconsistencia en la integridad de los datos por lo que se sugiere seguir los pasos antes mencionados.

**Consultas de Catálogos.** Permite generar reportes y muestras de la información en dos formas distintas de despliegue: pantalla e impresa, organizándola por medio de sus características principales o por un criterio arbitrario. Las opciones asignadas a esta opción se pueden ver en la figura 5.3.



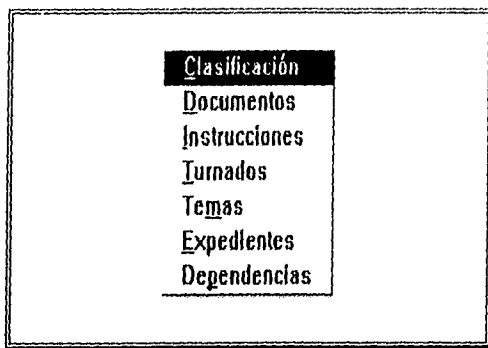


Figura 5.3 Menú de Altas, Bajas y Cambios de Catálogos ó Consultas de Catálogos

**Clasificación.** Consiste en agrupar la información de acuerdo a la prioridad o necesidad que requiera el documento. Se requiere de dos parámetros clave y nombre, para poder realizar las transacciones tanto de modificación/inserción como de consulta. El primero es para tener una referencia rápida de la descripción de cualquier clasificación (nombre). Donde nombre podrá tener una longitud máxima de veinte y cinco caracteres y la clave de dos caracteres. Únicamente debe haber una clave por cada nombre y viceversa, debido a que se requiere que cada catálogo sea distinto de los demás, el objetivo de tener la clasificación depende directamente de la importancia del documento u oficio (De carácter urgente, aviso, etc.).

**Documentos.** Descripción de todas las posibles variedades de clasificación de la información. Se requiere de dos parámetros para poder realizar transacciones de modificación/inserción como de consulta. Se requiere de una clave, que será la referencia rápida y una descripción asignada a la referencia rápida, la cual describe el tipo de documento u oficio, tales como decretos, audiencia, documento, invitación, oficio, gestión, regalos o varios. Los valores que puede tomar la clave puede ser un carácter de longitud uno y la descripción un conjunto de caracteres o palabras menor o igual a veinte.

**Instrucciones.** Acciones a realizar por parte del turnado, las cuales determinan el seguimiento del asunto o el propósito del documento. Se requiere de una clave como de una descripción para la fácil manipulación de estas, donde la clave tiene una longitud máxima de un carácter y la descripción de cuarenta y cinco caracteres. cuando se requiere de la manipulación de estos datos ya sea para actualización/modificación o para reportes, no se borren instrucciones sin antes asociar otra instrucción a los documentos, ya que genera problemas con la integridad de la base de datos.

**Temas.** Idea u objetivo principal de la información contenida en un documento. Requiere de un solo parámetro para la manipulación de la información, es decir únicamente requiere de la descripción ya que el identificador se generara o suprimirá dinámicamente dependiendo de las operaciones realizados con los temas. La descripción puede ser de hasta de cuarenta y cinco caracteres.

**Turnados.** Lista de personas disponibles, las cuales darán respuesta al oficio o documento. Se requiere de una clave y la descripción de datos generales donde se puede ubicar al turnado, es decir sus direcciones y teléfonos de casa y oficina. Es importante que se ingrese esta información ya que cuando se da un seguimiento a un oficio o documento el envío de correspondencia, etc. se tiene que realizar de forma mediata.

**Expedientes.** Es la clave que se le asigna a cada institución de acuerdo a su organización jerárquica. Se requiere de una clave y una descripción o nombre de la institución, las cuales siempre serán asignadas de manera única, por lo que existen varios expedientes a un solo documento.

**Dependencias.** Son las entidades, instituciones o empresas solicitantes de servicios. Las cuales únicamente se requiere ingresar el nombre o descripción de la dependencia.

**Documentos.** Integración general de información, la cual permite la alimentación, alteración, eliminación y generación de reportes de las características de la misma, complementándose con identificadores para el seguimiento de los oficios, como lo podemos ver en el menú asignado a esta opción en la figura 5.4

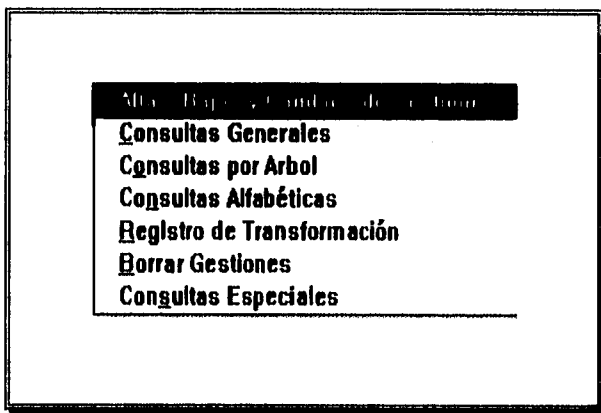


Figura 5.4 Menú de Documentos

**Altas y Cambios de Gestiones.** Registra y realiza movimientos de la información de un documento, con la finalidad de obtener una respuesta o contestación para este.

**Consultas Generales.** Búsqueda de información por medio de cualquier característica o atributo, para generar una vista de los datos requeridos, ya sea en pantalla como en una impresión.

**Consultas de Árbol.** Búsqueda de información, en la cual se presentan datos antecedentes y consecuentes, a la información citada.

*Consultas Alfabéticas.* Búsqueda de información por palabras de remitencia, referencia alfabética y temas.

*Registro de Transformación.* Genera oficios o documentos subsecuentes de otros.

*Bajas de Gestiones.* Elimina los registros de oficios o documentos, indicando el numero de documento deseado.

*Consulta Especial.* Búsquedas con características que describen de una manera específica al documento o al oficio, requiriendo de criterios específicos.

**Reportes.** Obtención de información recopilada con características requeridas por el usuario. Las funciones de esta opción podemos observarlas en la figura 5.5.

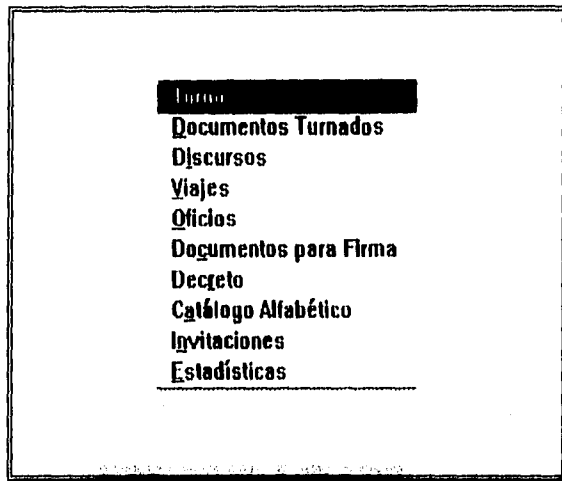


Figura 5.5 Menú de Reportes

*Turnos.* Reporta oficios o documentos en turno para su atención. Requiriendo de un rango de numero de documento.

*Documentos Turnados.* Genera un reporte por numero de documento mostrando el asunto, la instrucción, fecha y numero de documento del documento en cuestión.

*Relación de discursos.* Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de oficios o documentos, distintos tipos de turnados, clasificaciones e instrucciones para generar una relación de discursos con características del usuario.

*Programación de Viajes.* Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de documentos y clasificaciones para generar la programación de un viaje.

*Relación de Oficios Despachados.* Requiere de un rango de documento, distintos tipos de turnados y distintas clasificaciones e instrucciones , para reportar los oficios o documentos ya despachados.

*Documentos para Firma* Requiere de un rango de oficios o documentos, rango de fecha de recepción, distintos turnados, distintas clasificaciones, distintas instrucciones, para generar el reporte de documentos para firma.

*Proyectos de Decreto* Requiere de un rango de oficios o documentos, rango de fecha de recepción, distintos turnados, distintas clasificaciones, distintas instrucciones, para generar un proyecto de decreto.

*Catálogo Alfabético.* Requiere de un rango de numero de identificador para generar el catálogo o lista alfabética.

*Relación de invitaciones.* Requiere de un rango de numero de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de turnados, clasificaciones e instrucciones para generar una relación de invitaciones.

*Estadísticas* Requiere de un rango de identificador, tipo de documento y clasificación, para generar diferentes estadísticas de los documentos despachados.

**Utilerías.** Son las herramientas que ayudan a salvaguardar tanto la información capturada, como de todos sus elementos de que se compone el sistema para su buen funcionamiento. Las funciones de esta opción las podemos observar en la figura 5.6.

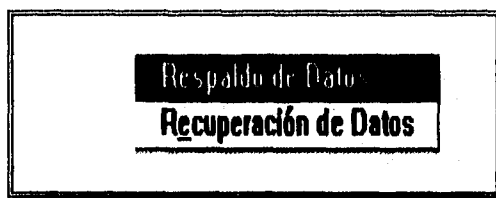


Figura 5.6 Menú de Utilerías

**Respaldo.** Crea una copia temporal de los oficios o documentos contenidos por el sistema.

**Recuperación.** Permite restaurar cualquier respaldo o copia temporal hacia la estructura (base de datos) de todos los oficios o documentos de respaldo.

### **5.3 Descripción de los Parámetros Requeridos por Pantalla del SICADE**

<b>SICADE</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Catálogos</i>	Información descriptiva de las características para los oficios o documentos	Sin Parámetros
<i>Documentos</i>	Recopilación de Información	Sin Parámetros
<i>Reportes</i>	Obtención de información recopilada con características requeridas por el usuario	Sin Parámetros
<i>Utillerías</i>	Herramientas que ayudan a salvaguardar la información	Sin Parámetros

Tabla 5.1 Menú Principal

<b>CATÁLOGOS</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Altas, Bajas y Cambios</i>	Menú de opciones para el tratamiento de oficios o documentos	Sin Parámetros
<i>Consultas</i>	Menú de opciones para la consulta de documentos	Sin Parámetros

Tabla 5.2 Catálogos

<b>ALTAS, BAJAS Y CAMBIOS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Clasificación</i>	Tipo de importancia que puede tomar un documento	Requiere de una clave asignada al tipo de importancia del documento así como de una descripción en la cual contenga la idea de clasificación. Ej. I+(Mayor interés), I-(Menor interés), etc.

Tabla 5.3 Altas, Bajas y Cambios

<b>ALTAS, BAJAS Y CAMBIOS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Documentos</i>	Tipo de Documentos	Requiere de una clave asignada al tipo del documento así como de una descripción en la cual contenga la idea del tipo de documento. Ej. Oficios, regalos, invitaciones, etc.
<i>Instrucciones</i>	Acciones que realiza el Turnado	Requiere de una clave asignada al documento la cual permite reconocer el tratamiento que a recibido la información, la cual deberá ser descrita por una idea concreta. Ej. Extensión de Decretos, Envío de Información, Aceptación de Audiencias, etc.
<i>Turnados</i>	Personas disponibles, las cuales darán respuesta al oficio o documento	Requiere de una descripción detallada de las personas disponibles a ser candidatos de asignación. Ej. Nombre completo, Dirección de casa, Dirección de Oficina, Teléfonos de Localización, etc.
<i>Temas</i>	Título asignado a los asuntos de un oficio o documento	Es indispensable asignar un tema a los documentos o a la información que se recibe cotidianamente, por lo que el tema identificara a los documentos de una forma particular para una posterior referencia a ellos. Requiere del Nombre del Tema únicamente ya que el sistema internamente asignara un identificador.

Tabla 5.3 Altas, Bajas y Cambios (Continuación)

<b>ALTAS, BAJAS Y CAMBIOS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Expedientes</i>	Es la clave que se le asigna a cada institución de acuerdo a su organización jerárquica	Requiere de una relación de identidad a nivel de institución, la cual tendrá la función de asociar instituciones o dependencias con sus diferentes departamentos o divisiones, generando identificadores secuenciales. Ej. para la dependencia X corresponde el identificador 1.2.3.4.5.0 y para sus direcciones corresponderán los identificadores comprendidos entre 1.2.3.4.5.1 y 1.2.3.4.5.9, esto depende de la magnitud de subdivisiones que contenga cada Institución o dependencia.
<i>Dependencias</i>	Nombres de las instituciones o empresas solicitantes	Es la definición o declaración de las dependencias involucradas con los oficios o documentos recibidos, requiere de un identificador y la descripción completa de la institución (nombre).

Tabla 5.3 Altas, Bajas y Cambios (Continuación)

<b>CONSULTAS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Clasificación</i>	Tipo de importancia que puede tomar un documento	Requiere de una clave asignada al tipo de importancia del documento así como de una descripción en la cual contenga la idea de clasificación. Ej. I+(Mayor interés),I-(Menor interés), etc.

Tabla 5.4 Consultas

<b>CONSULTAS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Documentos</i>	Tipo de Documentos	Requiere de una clave asignada al tipo del documento así como de una descripción en la cual contenga la idea del tipo de documento. Ej. Oficios, regalos, invitaciones, etc.
<i>Instrucciones</i>	Acciones que realiza el Turnado	Requiere de una clave asignada al documento la cual permite reconocer el tratamiento que a recibido la información, la cual deberá ser descrita por una idea concreta. Ej. Extensión de Decretos, Envío de Información, Aceptación de Audiencias, etc.
<i>Turnados</i>	Personas disponibles, las cuales darán respuesta al oficio o documento	Requiere de una descripción detallada de las personas disponibles a ser candidatos de asignación. Ej. Nombre completo, Dirección de casa, Dirección de Oficina, Teléfonos de Localización, etc.
<i>Temas</i>	Título asignado a los asuntos de un oficio o documento	Es indispensable asignar un tema a los documentos o a la información que se recibe cotidianamente, por lo que el tema identificara a los documentos de una forma particular para una posterior referencia a ellos. Requiere del Nombre del Tema únicamente ya que el sistema internamente asignara un Identificador.

Tabla 5.4 Consultas (Continuación)



<b>CONSULTAS(*)</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Expedientes</i>	Es la clave que se le asigna a cada institución de acuerdo a su organización jerárquica	Requiere de una relación de identidad a nivel de institución, la cual tendrá la función de asociar instituciones o dependencias con su diferentes departamentos o divisiones, generando identificadores secuenciales. Ej. para la dependencia X corresponde el identificador 1.2.3.4.5.0 y para sus direcciones corresponderán los identificadores comprendidos entre 1.2.3.4.5.1 y 1.2.3.4.5.9, esto depende de la magnitud de subdivisiones que contenga cada institución o dependencia.
<i>Dependencias</i>	Nombres de las instituciones o empresas solicitantes	Es la definición o declaración de las dependencias involucradas con los oficios o documentos recibidos, requiere de un identificador y la descripción completa de la institución (nombre).

Tabla 5.4 Consultas (Continuación)

(\*) Estos dos tipos de menú son directamente iguales, la diferencia estriba en la manera, en la cual es explotada la información, por un lado será posible modificar o intercambiar los datos; por el otro lado únicamente se podrá consultar la información referente.

<b>DOCUMENTOS</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Gestiones</i>	Registra y realiza movimientos de información de un documento.	Es la actividad de un documento el cual tiene un paso de transición entre no estar asignado a ser asignado a un turnado o la resolución de este. Llevando un control estricto en fechas de recepción o resolución, además de las personas involucradas con el documento.
<i>Consultas Generales</i>	Búsqueda de información	Permite obtener información de cualquier documento en cualquier estado de actividad, por medio de los diferentes parámetros de ingreso, tales como fechas de ingreso, dependencias involucradas, turnados involucrados, etc.

Tabla 5.5 Documentos

<b>DOCUMENTOS</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Consultas de Árbol</i>	Búsqueda de información, en la cual se presentan datos antecedentes y consecuentes a la información citada	La búsqueda de información es importante, ya que relaciona los documentos involucrados intrínsecamente, mostrando un entorno de seguimiento para cada documento. Es importante proporcionar el identificador del documento, ya que por medio de este realiza las búsquedas de relación entre documentos.
<i>Consultas Alfabéticas</i>	Búsqueda de información por palabra de remitencia, referencia alfabética y temas	Por el gran volumen de información, la búsqueda de títulos es difícil de recordar, es así que se permite hacer referencias únicamente por palabras aisladas, obteniendo los documentos o información de una manera mas eficiente y segura.
<i>Registro de Transformación</i>	Genera oficios o documentos subsiguientes de otros	Debido al tiempo de asignación como el de resolución, es importante que los documentos sean resueltos. Sin embargo las instituciones generan documentos secundarios sobre el mismo tema o problema, por lo que si la información es nuevamente encausada, generara una duplicidad de información así como de trabajo, por lo que los registros de transformación permiten reorganizar la información, sometiéndola únicamente a un análisis y resolución, generando nuevos documentos con relación a los anteriores recibidos.
<i>Bajas</i>	Borra los registros de oficios o documentos	Cuando un documento es inconsistente, ya sea por falta de información o por duplicidad de esta es necesario que sea retirado o excluido del demás grupo de documentos. Esto lo realiza por medio del identificador asignado a cada documento, asegurándose que únicamente sea retirado el documento requerido.
<i>Consultas Especiales</i>	Realiza las consultas por tipo de documento, rango de identificador, clasificación, instrucción y tomado	LA agrupación de información es importante, ya que con ella podemos obtener información detallada del conjunto de documentos que cumplan con las características establecidas, a su vez, el rendimiento o desempeño de los tomados así como de los documentos.

Tabla 5.5 Documentos (Continuación)

<b>REPORTE</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Impresión de Turnados</i>	Genera la relación de personas o turnados con relación a un rango de numero de turnados (identificador).	Requiere de un rango de numero de documento para generar los oficios o documentos de turno.
<i>Documentos Turnados</i>	Genera la relación de todos los documentos turnados, dependiendo de un numero de documento asignado.	Reporte por numero de documento
<i>Relación de Discursos</i>	Este reporte es importante, ya que por medio de este se genera la relación de discursos así como las fechas de realización de estos. También realiza una interrelación con los turnados asignados para esto.	Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de oficios o documentos, distintos tipos de turnados, clasificaciones e instrucciones para generar una relación de discursos.
<i>Programación de Viajes</i>	Genera la relación de documentos que requieren de desplazamientos contra los turnados asignados, para poder determinar itinerarios o rutinas de trabajo.	Requiere un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de documentos, distintos tipos de clasificaciones para generar la programación de viajes
<i>Relación de Oficios Despachados</i>	Este reporte permite obtener dentro de un rango preestablecido los documentos que han sido despachados o resueltos, dentro de un lapso de tiempo.	Requiere un rango de documento, distintos tipos de turnados, distintas clasificaciones, y distintas instrucciones para generar una relación de oficios o documentos despachados.
<i>Documentos para Firma</i>	Los documentos de firma son considerados dentro de permisos, avisos, etc. los cuales deben generar resoluciones con avisos o acuses de recibo, o lo que es importante generar una relación de estos para su oportuna solución.	Requiere un rango de oficios o documentos, rango de fecha de recepción, distintos turnados, distintas clasificaciones, distintas instrucciones para generar una documentación de firma
<i>Proyectos de Decreto</i>	Es la relación de documentos que requieren de una solución general, cambios de lineamientos o nuevas normas para la solución de nuevos problemas, a los cuales se enfrentan los turnados.	Requiere un rango de numero de oficios o documentos, rango de fecha de recepción, distintos turnados, clasificaciones e instrucciones para generar un proyecto de decreto
<i>Catálogo Alfabético</i>	Este catálogo genera una relación de los temas alfabéticamente por medio del identificador único de documentos.	Requiere un rango de numero de identificador para generar el catálogo o lista alfabética

Tabla 5.6 Reporte

<b>REPORTE</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Relación de Invitaciones</i>	Este reporte genera la relación de invitaciones por medio de un lapso de tiempo asignados a sus turnados correspondientes.	Requiere un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de turnados, clasificaciones e instrucciones, para generar una relación de invitaciones.
<i>Estadísticas</i>	Genera aun relación general de documentos clasificados por tipos de documentos y por resoluciones, obteniendo incidencias de los documentos por medio de sus identificadores.	Requiere un rango de identificador, tipo de documento y clasificación, para generar estadísticas

Tabla 5.6 Reporte (Continuación)

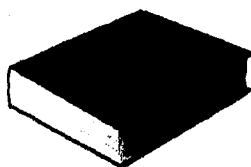
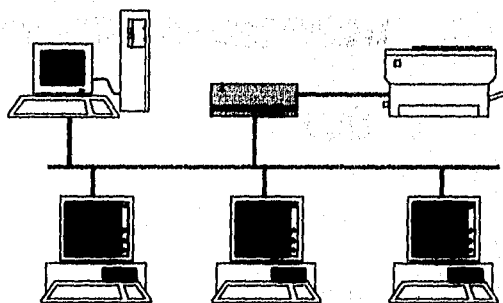
<b>UTILERÍAS</b>		
<b>Nombre</b>	<b>Descripción</b>	<b>Parámetros Requeridos</b>
<i>Respaldo</i>	Genera el respaldo de la base de datos, en forma total para generar un archivo plano en un directorio específico. Debe mencionarse que este proceso deberá ejecutarse periódicamente para tener un respaldo, reciente y no sufrir pérdidas considerables si llegase a sufrir algún daño la base de datos	Ver Manual Técnico Cap. 6
<i>Recuperación</i>	Este proceso permite recargar la base de datos contra información de periodos anteriores, debido a una incorrecta acentuación de información o por pérdidas de integridad de la base de datos.	Ver Manual Técnico Cap. 6
<i>Manejo de Datos</i>	Permite la integración e interrelación de toda la base de datos para poder generar un archivo con características determinadas por el usuario, para poder generar relaciones de reportes; explotadas por otras herramientas dedicadas a este objetivo, tales como hojas de calculo o sencillamente puede funcionar como retroalimentador de otro sistema si así se requiere periódicamente.	Ver Manual Técnico Cap. 6

Tabla 5.7 Utilerías

# CAPÍTULO 6

## MANUAL DEL USUARIO DEL SISTEMA DE CONTROL Y AUTOMATIZACIÓN DE DOCUMENTOS A GRAN ESCALA (SICADE)

---



El Sistema de Control y Automatización de Documentos (SICADE) es una poderosa herramienta para la administración de documentos de una empresa. Con él se puede dar seguimiento a la información que recibe el director general de una empresa, en este caso el C. Secretario de la Secretaría de Hacienda<sup>1</sup>, y asignar el trabajo a cada subordinado dependiendo del asunto el cual trate el oficio.

SICADE se utiliza para almacenar la información, generar distintos reportes, digitalizar los documentos confidenciales, respaldar y recuperar registros de la base de datos. Para estas dos últimas actividades lo desarrollaran él o los administradores de la base de datos, los cuales tendrán una clave propia para poder realizar estas actividades.

Este capítulo incluye la descripción detallada de cada menú: catálogos, documentos, reportes, utilerías y salida, así como el procedimiento para iniciar el sistema, desde el punto de vista del usuario.

### 6.1 Inicio del SICADE

Al arrancar el sistema Windows, aparecerá el ícono de SICADE en la ventana del grupo de Coordinación Administrativa, tal como se muestra en la figura 6.1.

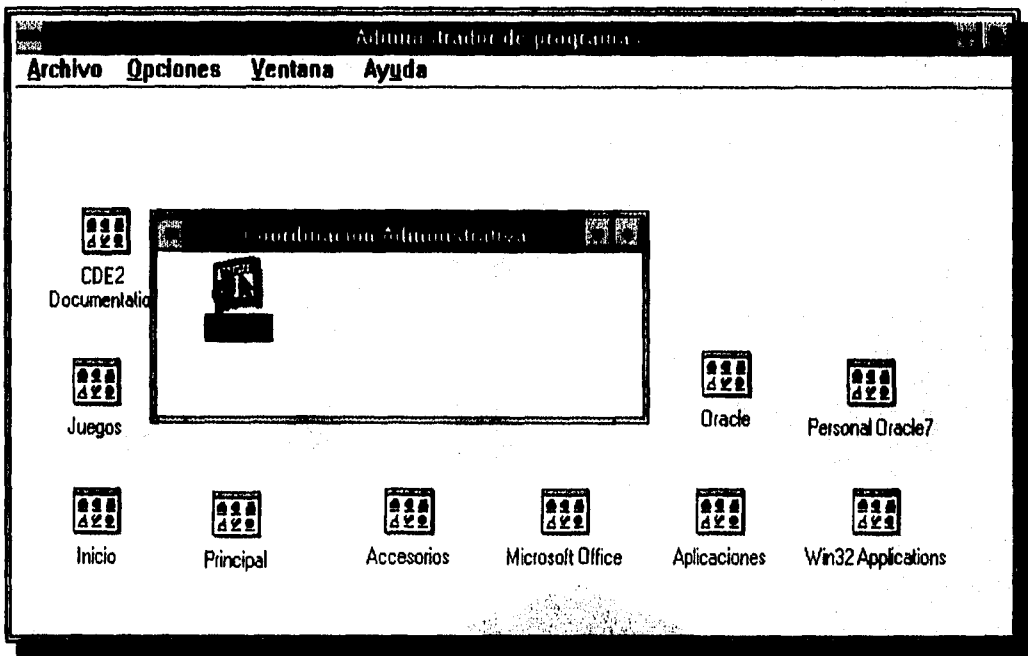



Figura 6.1 Grupo e ícono principal del sistema

<sup>1</sup> El sistema se ha aplicado a la Secretaría de Hacienda y Crédito Público

Para iniciar el Sistema de Control y Automatización de Documentos:

- Haga doble-clic en el ícono del SICADE.
- O bien, presione las teclas de dirección para mover el cursor de selección hasta el ícono del SICADE después presione , ahora aparecerá una ventana tal como se muestra, en la figura 6.2, aquí el usuario o administrador proporcionará la clave y password correspondiente.

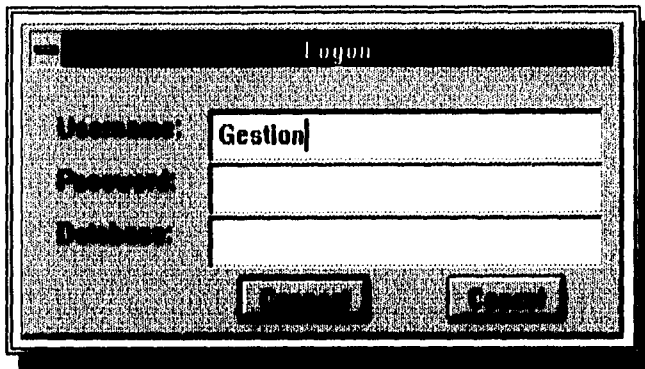


Figura 6.2 Ventana de seguridad del sistema

La ventana principal del sistema aparecerá después de haber proporcionado los datos respectivos, esta se muestra en la figura 6.3. Este Sistema, cuenta con un menú principal que se integra con cinco diferentes opciones, las primeras tres las ejecutan los usuarios del sistema y la otra junto con las anteriores el administrador, mismas que a su vez se apoyan en menús subsecuentes que se irán describiendo en forma secuencial.

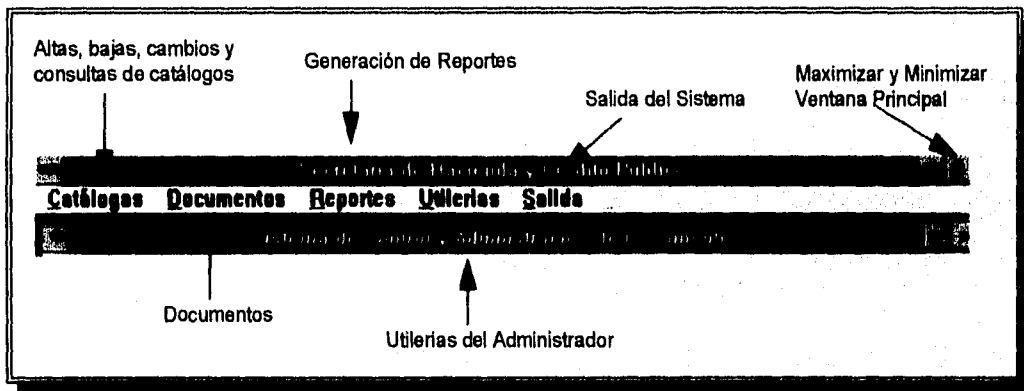


Figura 6.3 Menú Principal del Sistema de Control y Automatización de Documentos

En la figura 6.4 se observar la estructura general del sistema, básicamente está formado por cuatro menús principales, cada uno de éstos tiene submenús, los cuales permiten la entrada a los formatos de captura y consulta.

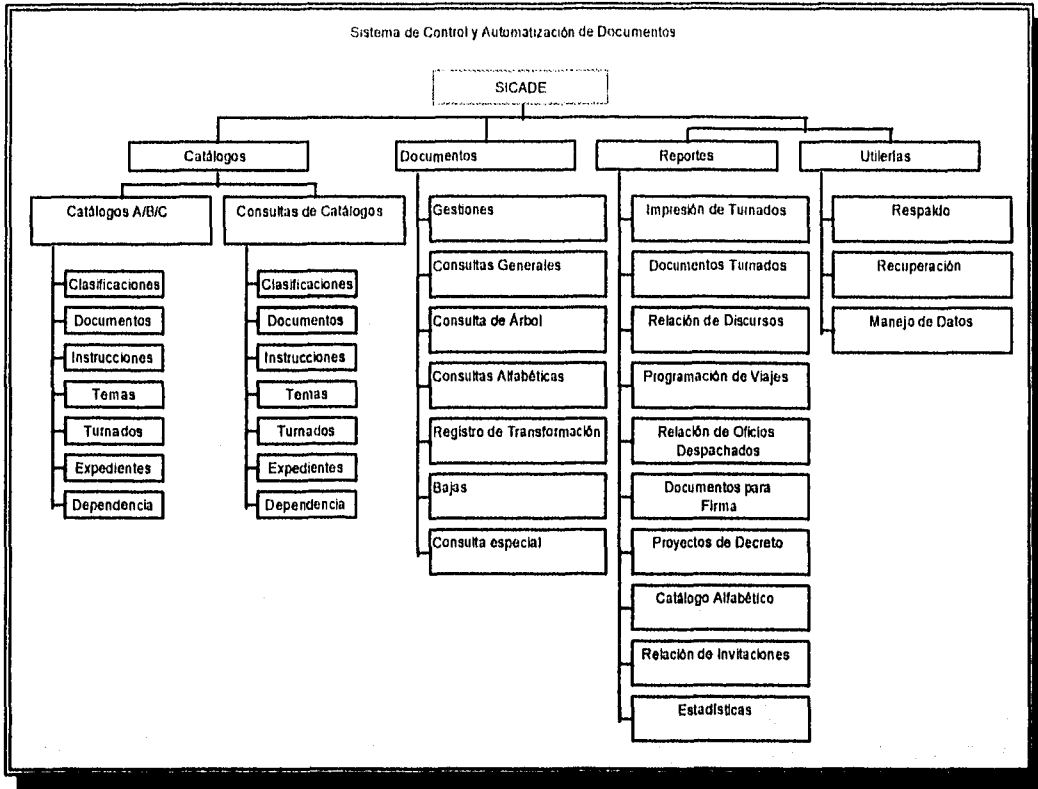


Figura 6.4 Estructura Principal del Sistema

## 6.2 Altas, bajas y actualización de catálogos

Para poder acceder a la opción de Altas, Bajas y Cambios de Catálogos, se realiza lo siguiente:

- Presione las siguientes teclas **Alt** **C**
- Con las teclas de dirección ubicarse en el menú y pulsar **↓**.
- O bien, con el mouse presionar clic para seleccionar.

Como se podrá observar en la figura 6.5 se presenta un submenú, en la primera opción el usuario podrá realizar altas, bajas y actualizar los catálogos, en cambio en la segunda opción podrá generar reportes de los catálogos.



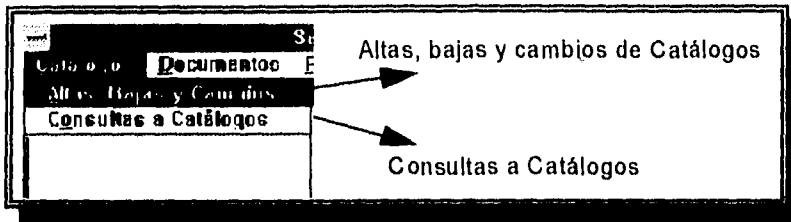


Figura 6.5 Submenú del menú de Catálogos

A continuación se explicará únicamente el primer submenú, el cual desplegará las opciones mostradas en la figura 6.6; con sólo oprimir **↓** o **↑** presentará en pantalla éstas alternativas.

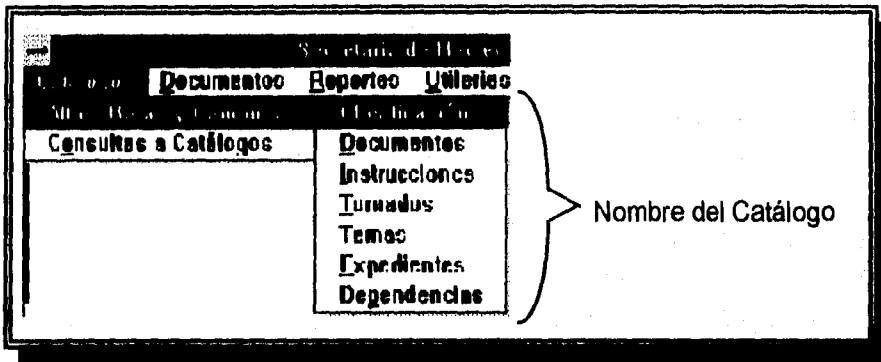


Figura 6.6 Opciones del submenú altas, bajas y cambios de catálogos.

Existen dos formas para poder acceder a cada una de estas opciones:

- Con las teclas de dirección **↓** **↑** se localiza la opción deseada y se oprime **↓**, o
- Con la tecla de la letra que está subrayada dentro del menú.

En el primer caso para poder acceder al catálogo de clasificaciones se oprime **C** y se presentará la pantalla mostrada en la figura 6.7.

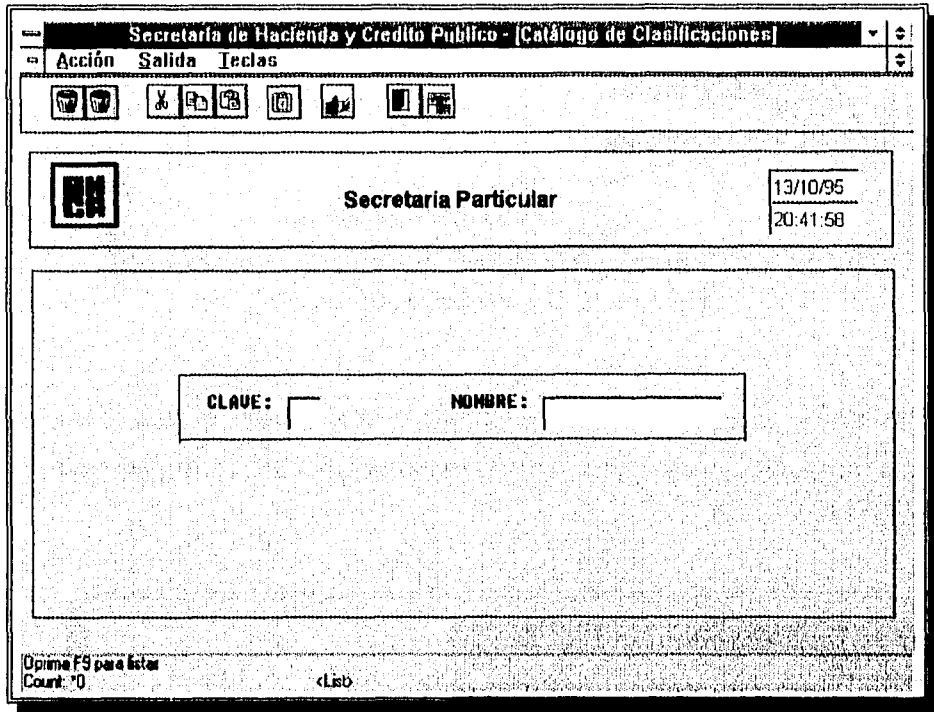


Figura 6.7 Pantalla de Catálogo de Clasificaciones

Como se puede observar en la figura 6.7, la pantalla se caracteriza por tener un menú, el cual se puede acceder por medio de la tecla **A** y la letra que se encuentra subrayada en el menú; además, contiene íconos, los cuales permiten mostrar diferentes funciones que se muestran en la figura 6.8, éstos permiten al usuario manejar distintas instrucciones con mayor facilidad; en la parte inferior de la pantalla, están presentes los mensajes principales ya sean de tipo informativo de las funciones de cada ícono, mensajes de las teclas principales de acuerdo al campo en donde se encuentre ubicado o desplegará el error si existe.

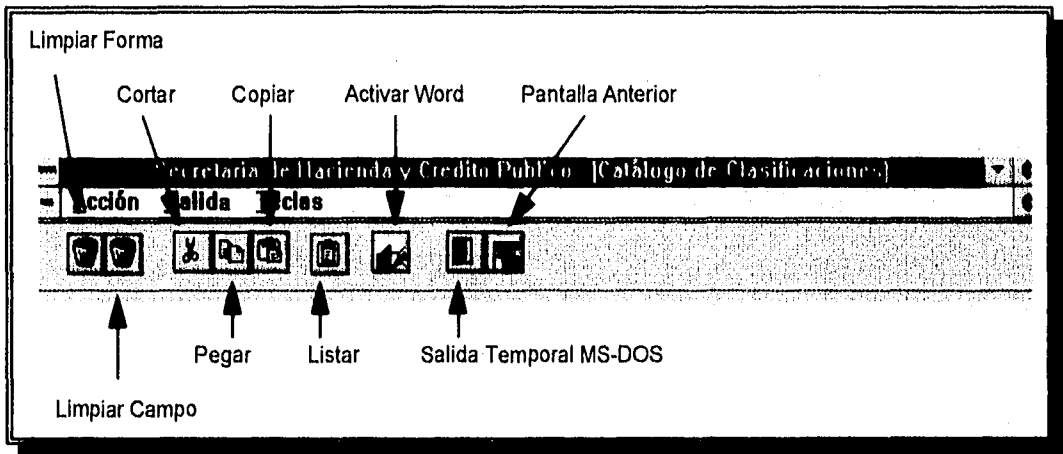












Figura 6.8. Funciones de los íconos del Catálogo de Clasificaciones.

Para poder dar de alta una clasificación nueva se tecldea la clave correspondiente y se oprime  o  para desplazarse al siguiente campo (Nombre) en donde se escribirá la descripción de la clasificación. Para poderlo salvar es necesario teclear , lo que automáticamente salvará el registro. Es necesario aclarar que no se permite duplicar la clave de las clasificaciones, por lo cual si el usuario llega a escribir por equivocación una clave existente, automáticamente desplegará la descripción de la clave, en este status se puede realizar la modificación de la descripción de la clasificación el cual se salvará oprimiendo también ; o bien, se puede borrar este registro de la siguiente manera:

- Se oprime  , donde desplegará el submenú presentado en la figura 6.9, con las teclas de dirección   se localiza Borrar o se oprimen las teclas  

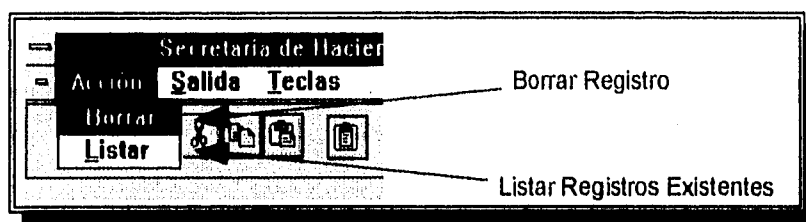




Figura 6.9. Submenú de Acción

- o se oprimen la teclas  <sup>2</sup>, como se muestra en la figura 6.10, la ayuda de teclas

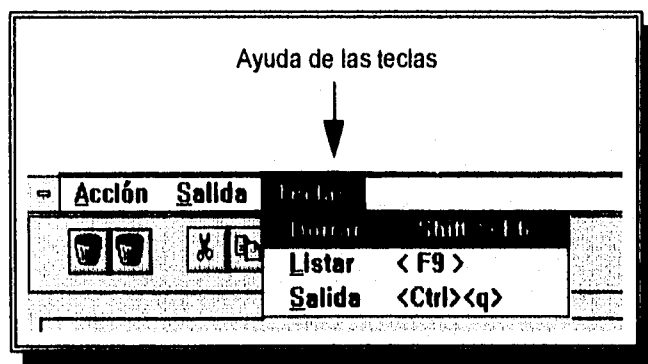



Figura 6.10 Ayuda de Teclas oprimiendo Alt + T

<sup>2</sup>  Es la tecla shift

Posteriormente se desplegará una ventana como la mostrada en la figura 6.11, donde se debe de confirmar la operación.

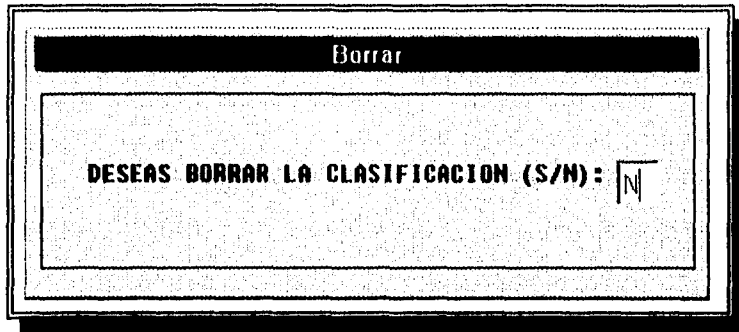


Figura 6.11 Ventana de confirmación para borrar un registro.

Para poder listar los registros es necesario oprimir **Alt** **A** y **L** o **F9**, lo cual presentará una lista tal y como se muestra en la figura 6.12, si la lista contiene una cantidad enorme de registros podemos localizar el registro en cuestión de la siguiente manera:

En el botón Find se colocarán la o las letras con las que inicie la clasificación, con el mouse presione el botón Find, automáticamente presentará todos los registros que contienen la condición propuesta.



Figura 6.12 Lista de clasificaciones

Nota: El signo % es un comodín que puede ir al principio, en medio o al final de la palabra.

Para poder salir de esta pantalla será necesario oprimir **Alt** **S**.

Es importante recordar que se pueden emplear los iconos que se muestran en la figura 6.8 con solo arrastrar el mouse y colocarlo en el icono correspondiente o en las opciones del menú antes mencionadas.

Esta funcionalidad se realiza para cada uno de los siguientes catálogos: documentos figura 6.13, instrucciones figura 6.14, temas figura 6.15, expedientes figura 6.16 y dependencias figura 6.17.

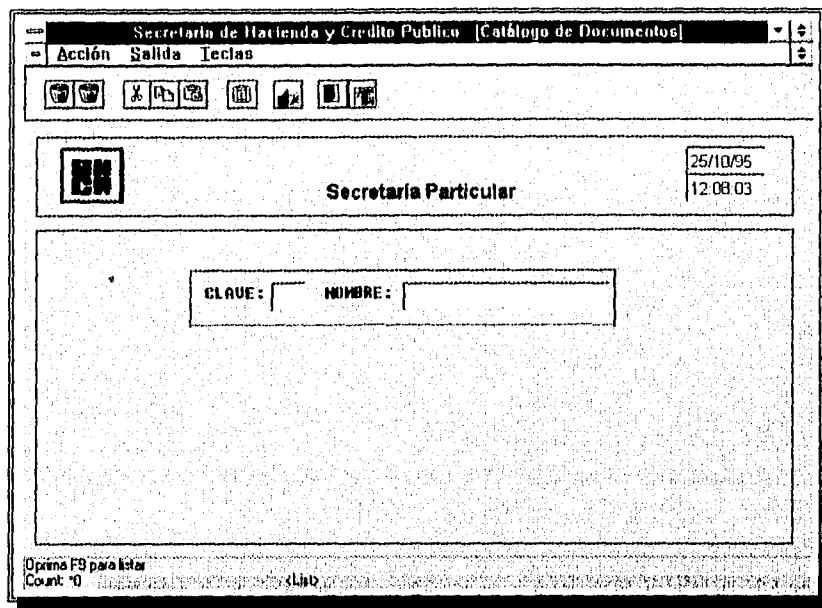


Figura 6.13 Pantalla del Catálogo de Documentos

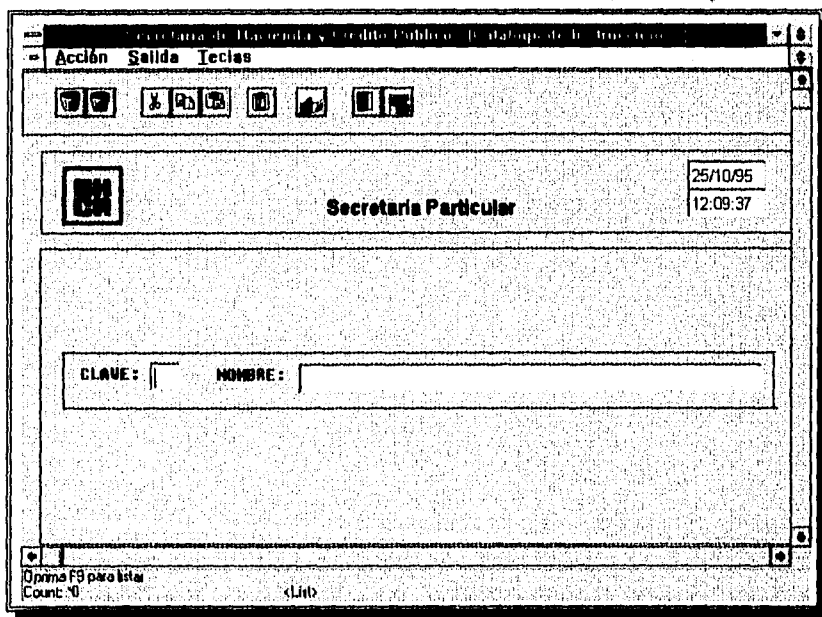


Figura 6.14 Pantalla del Catálogo de Instrucciones

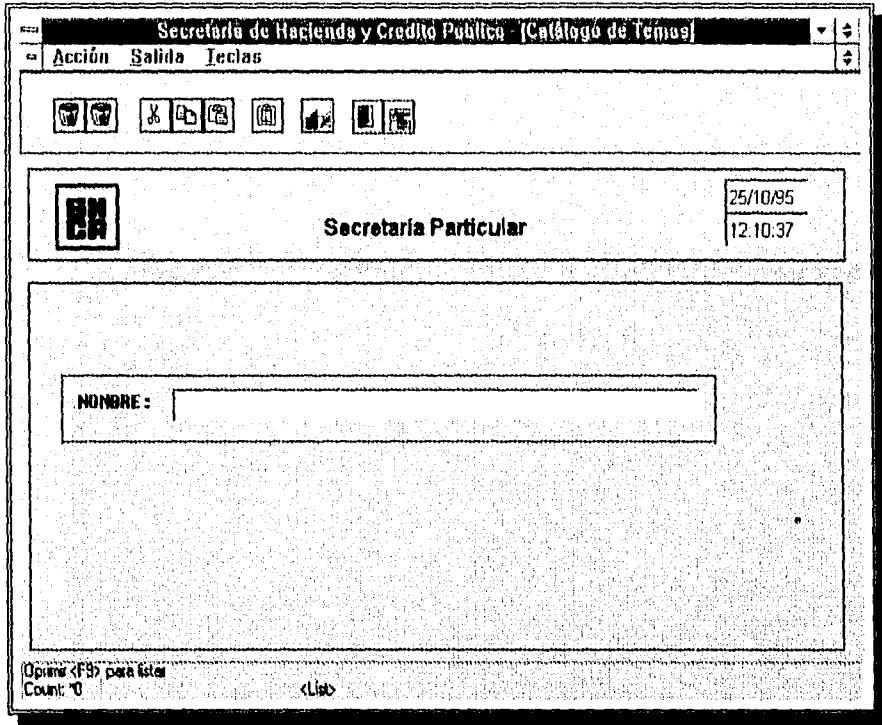


Figura 6.15 Pantalla del Catálogo de Temas

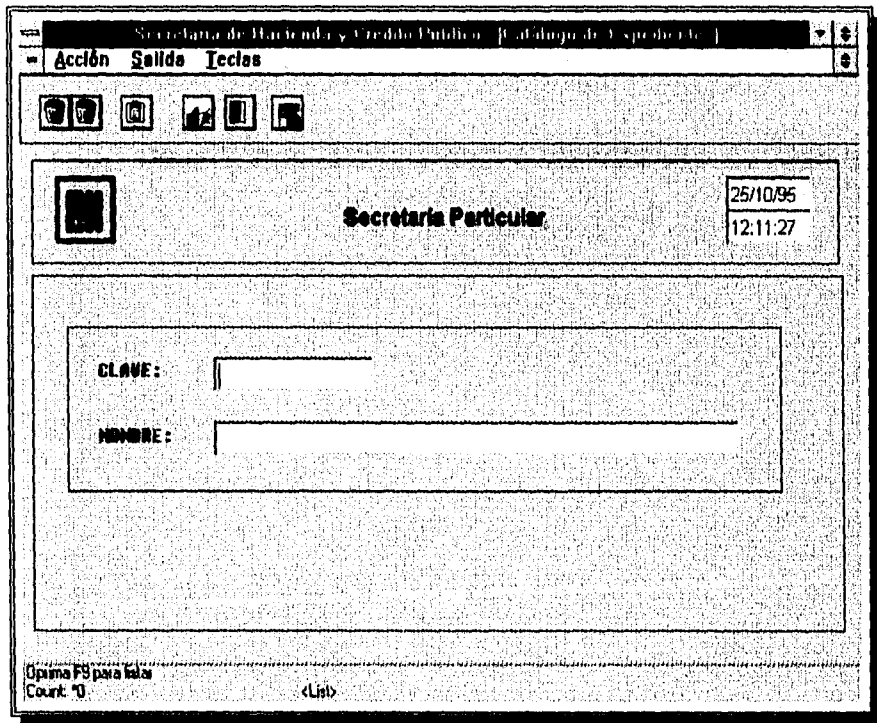


Figura 6.16 Pantalla del Catálogo de Expedientes

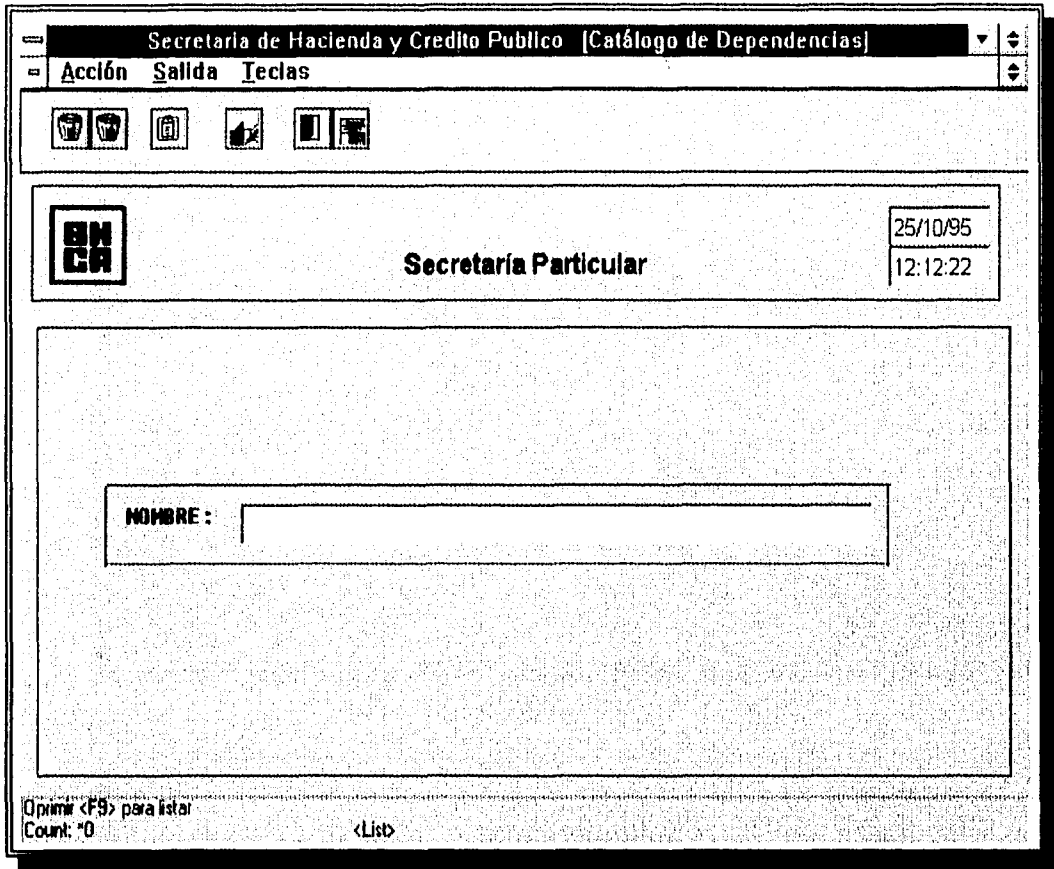



Figura 6.17 Pantalla del Catálogo de Dependencias

Como se puede observar en la figura 6.18, el catálogo de turnados, las funciones principales altas, bajas y cambios se manejan de distinta manera.

Para dar de Alta un registro, es necesario teclear la clave, con la tecla  se desplazará a los campos: nombre, cargo y área que son necesarios para poder dar de alta el registro nuevo.

Los campos calle, teléfono, fax de domicilio y oficina permiten activar un pequeño editor con las teclas **Ctrl** **e**, o bien, arrastrando el mouse hasta encontrar el ícono señalado en la figura 6.19 editor, otra forma sería utilizando el menú, en éste como en los casos anteriores, se oprime la tecla **Alt** y la letra inicial del menú, para este caso la letra **E**, presentando el menú mostrado en la figura 6.20

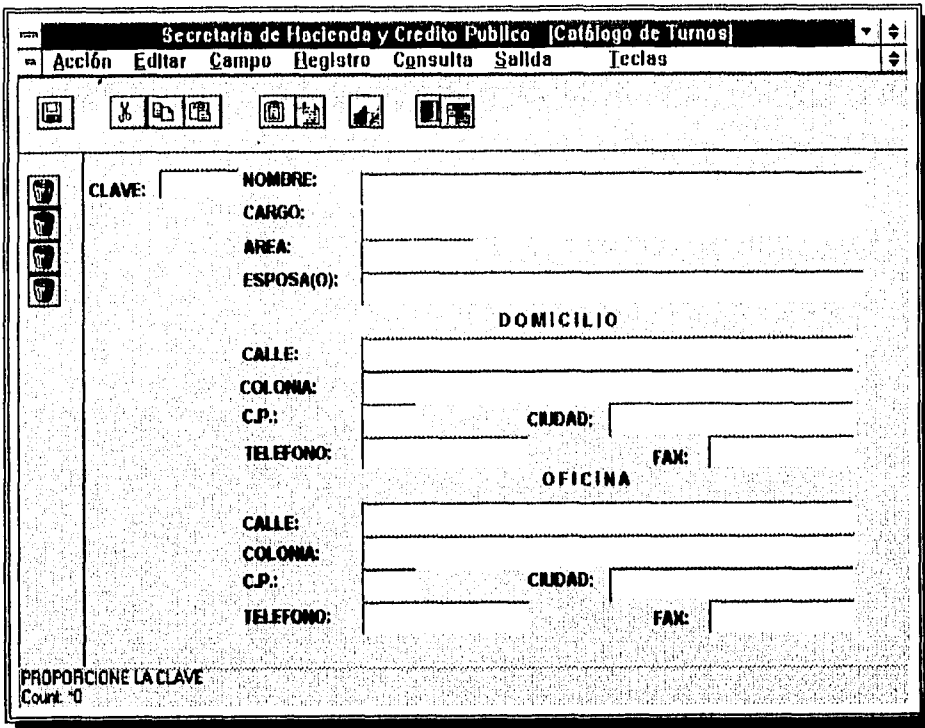


Figura 6.18 Pantalla de Catálogo de Turnados

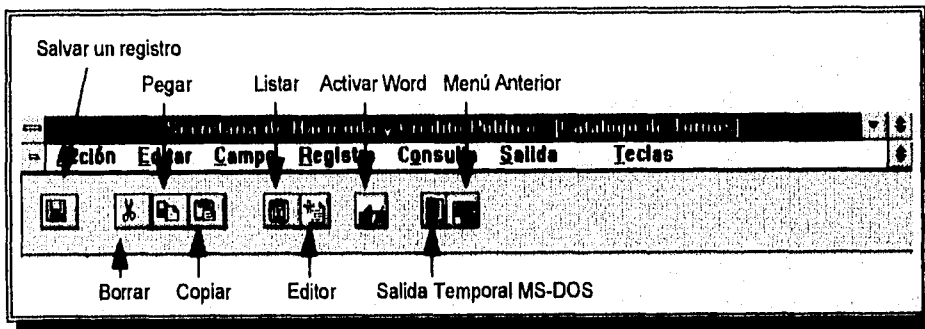


Figura 6.19 Iconos del Catálogo de Turnados

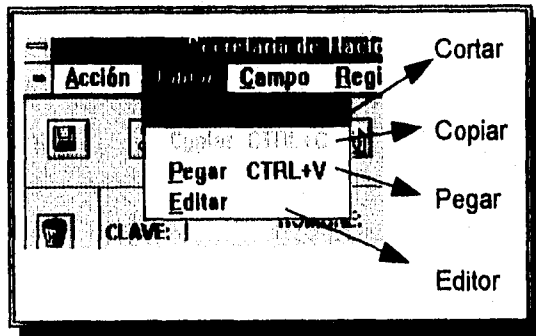


Figura 6.20 Menú de Edición



Para salvar el registro se oprime **F10**, en caso de utilizar el menú se oprimirá **Alt A**, el cual mostrará la pantalla mostrada en la figura 6.21. La primera opción del submenú limpia toda la pantalla con la alternativa de salvar los cambios o actualizaciones y en la segunda salva automáticamente todos los cambios; para el primero se oprime la tecla **L** y en el segundo la **A**.

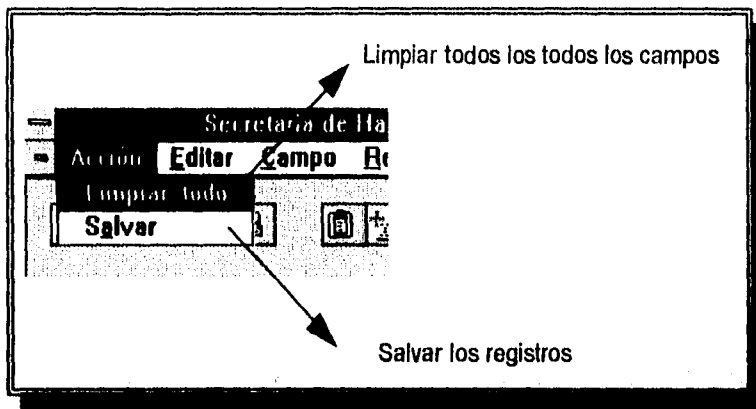


Figura 6.21 Menú de Acción

Si el usuario requiere realizar alguna consulta, se puede efectuar de dos maneras distintas: por clave y/o por consulta general de turnado con condición.

Para el primer tipo de consulta solo es necesario teclear la clave del turnado si existe, al desplazarse al siguiente campo automáticamente se desplegarán los demás datos que contiene este registro.

En el caso del segundo tipo de consulta se oprime **Ctrl** **Alt** **F7** presentando la pantalla mostrada en la figura 6.22. Primero el usuario tendrá que oprimir **F7** o **Alt** y la letra subrayada de consultas, en este caso la letra **O** para poder presentar el submenú de la figura 6.23, donde se elegirá la definición de consultas **D**.

Posteriormente el usuario le proporcionará la clave o una condición para obtención de claves empleando el "%" como comodín dentro de las claves, es decir, si se necesitan los datos de todos los turnados cuyas claves inicien con las letras "AL", en las claves se escribirá la siguiente condición "AL%".

Por último, al oprimir **F8** o la tecla **E** del menú de consultas se ejecutará la consulta, presentando la clave, nombre, cargo y área.

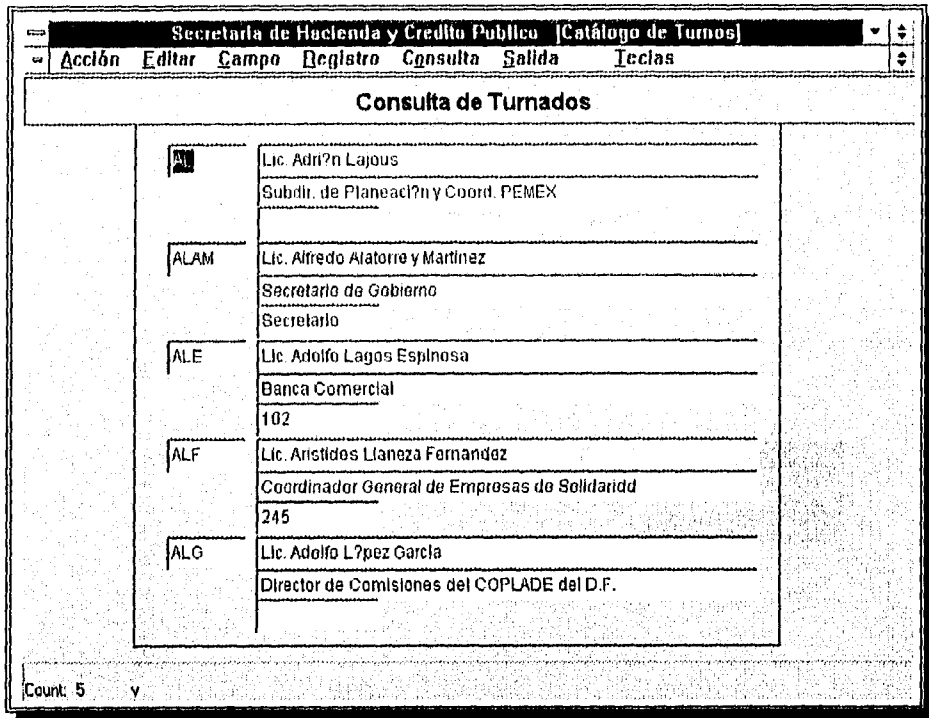


Figura 6.22 Pantalla de Consultas de Turnados

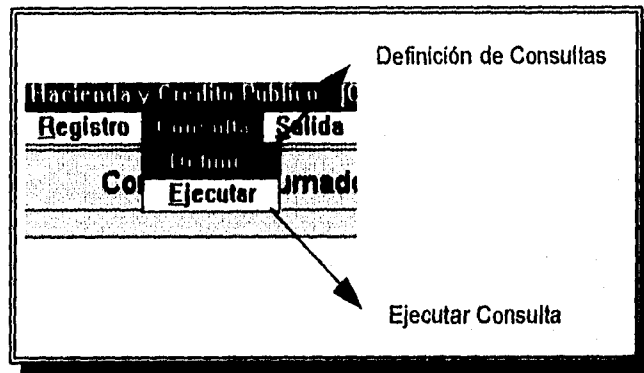


Figura 6.23 Menú de Consultas

Para poder desplazarse entre cada registro solo basta con oprimir las teclas de dirección , si de la lista se elige un registro. Al oprimir se escogerá el registro presentando a éste en la primera pantalla de turnados con cada uno de los datos que lo conforman.

En el caso que se desee borrar el registro solo basta con oprimir o desplegar el menú de registro, el cual presenta la operación de borrado donde desplegará la ventana de confirmación de operación como se mostró en la figura 6.11.

Como se puede observar, para poder regresar al menú anterior es necesario ubicarse en la opción de salida o bien con el mouse localizar el ícono del menú anterior, como se muestra en la figura 6.19. Se puede observar que en cada una de las pantallas de catálogos existe un menú de teclas, este presenta las funciones de cada tecla.

### **6.3 Consultas de Catálogos**

La consulta de catálogos permite observar los registros de clasificaciones, documentos, instrucciones, temas, turnados, expedientes y dependencias por pantalla y por impresora. Para poder acceder a la opción de Consultas de Catálogos, se realiza lo siguiente:

- Presione las siguientes teclas **Alt** **C**
- Con las teclas de dirección ubicarse en el menú y pulsar **↓**.
- O bien, con el mouse presionar clic para seleccionar.

Como se puede observar en la figura 6.24, la segunda opción permite generar los *reportes de catálogos*, con solo oprimir **↓** o **0** presentará en pantalla éstas alternativas.

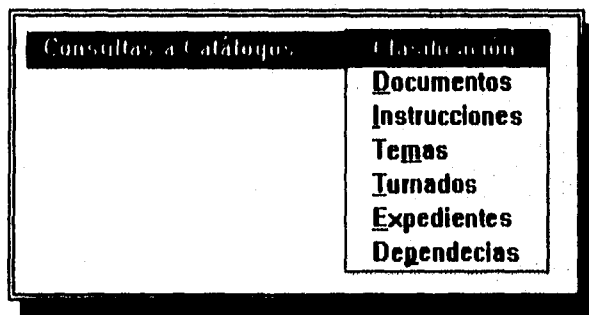


Figura 6.24 Opciones para las Consultas de Catálogos.

Existen dos formas para poder acceder a cada una de estas opciones:

- Con las teclas de dirección **↓** **↑** se localiza la opción deseada y se oprime **↓**, o
- Con la tecla de la letra que está subrayada dentro del menú.

En el primer caso para poder acceder al catálogo de clasificaciones si se oprime **C** y se presentará la pantalla mostrada en la figura 6.25

Como se puede observar, la pantalla se caracteriza por tener un menú, el cual se puede acceder por medio de la tecla **Alt** y la letra que se encuentra subrayada en el menú; además, contiene iconos, los cuales sirven para diferentes funciones, estas se muestran en la figura 6.26. Éstos permiten al usuario manejar distintas instrucciones con mayor facilidad; en la parte inferior de la pantalla, están presentes los mensajes principales, que son de tipo informativo, de las funciones de cada icono, tales como mensajes de las teclas principales de acuerdo al campo en donde se está ubicado o desplegará el error si existe.

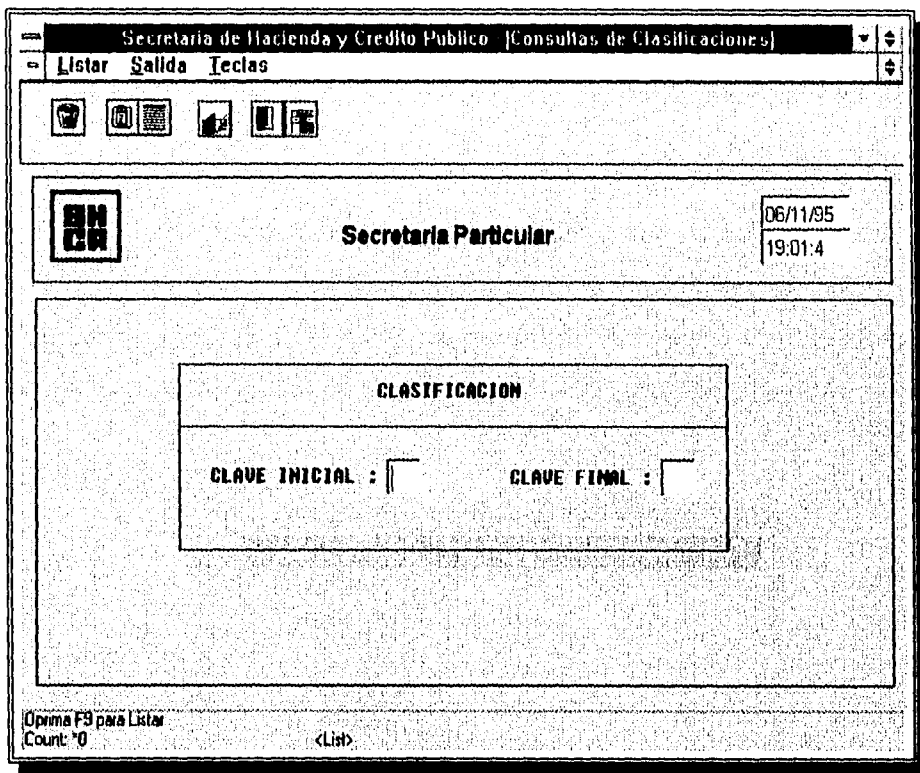


Figura 6.25. Catálogo de Clasificaciones

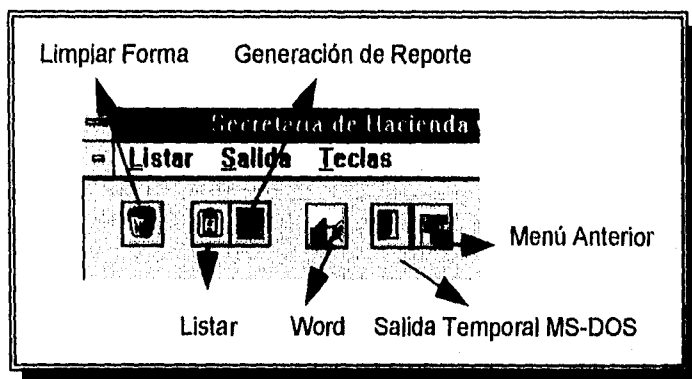


Figura 6.26 Iconos de pantalla de consultas (Clasificaciones)

Para llevar a cabo la generación del reporte se deben realizar los siguientes pasos:

- En el primer campo escribir la clave de la clasificación inicial u oprimir **F9** para listar las clasificaciones y elegir la que nos interesa con solo oprimir **↵**.
- Desplazarse al segundo campo y realizar los pasos descritos anteriormente.
- Oprimir **↵** o buscar el ícono de reporte, el cual se muestra en la figura 6.26.

En la figura 6.27 se muestra el reporte generado de clasificaciones, el cual se puede imprimir con solo presionar el ícono de Print, las funciones básicas de estos íconos se muestran en la figura 6.28

The screenshot shows a window titled 'Catálogo de Clasificaciones'. The menu bar includes 'File', 'Edit', 'Window', and 'Help'. Below the menu bar are buttons for 'Prev', 'Next', 'First', 'Last', 'Page:', 'Print', 'Close', and 'New'. The main content area displays the following information:

07-11-95 SECRETARIA DE HACIENDA Y CREDITO PUBLICO PAGINA: 1  
SECRETARIA PARTICULAR DE: 1  
CATALOGO DE CLASIFICACIONES

CLASIFICACION	DESCRIPCION
*+	INSTR. ESP. +
*-	INSTR. ESP. -
C+	CONOCIMIENTO +
C-	CONOCIMIENTO -
I+	IMPORTANTE +
I-	IMPORTANTE -
N+	NORMAL +
N-	NORMAL -
R+	RELEVANTE +

Figura 6.27 Reporte de Clasificaciones

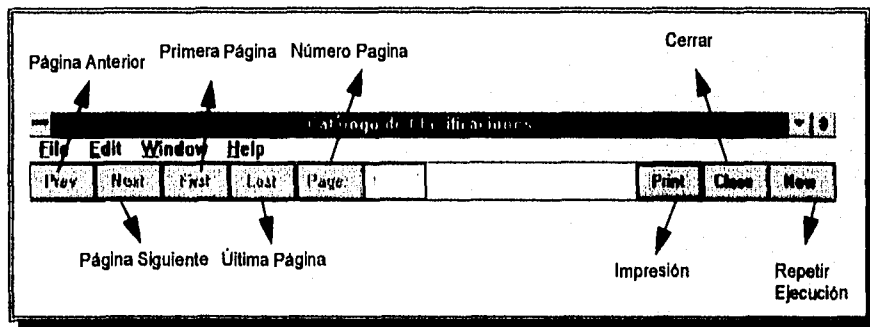


Figura 6.28 Funciones básicas de los íconos de reportes

Para configurar la impresora es necesario localizar el menú de Print Status, el cual se localiza en el menú File presionando **Alt F** el cual presentará la pantalla mostrada en la figura 6.29, la cual nos permitirá elegir el tipo de impresora de salida, en necesario aclarar que todos estos reportes generados desde los catálogos se imprimen por medio de una impresora láser.

En el caso de las consultas de documentos figura 6.30, dependencias figura 6.31, temas figura 6.32, expedientes figura 6.33, Instrucciones figura 6.34 se solicitará un rango de registros para poder generar el reporte de cada uno de éstos tal y como de describió anteriormente.

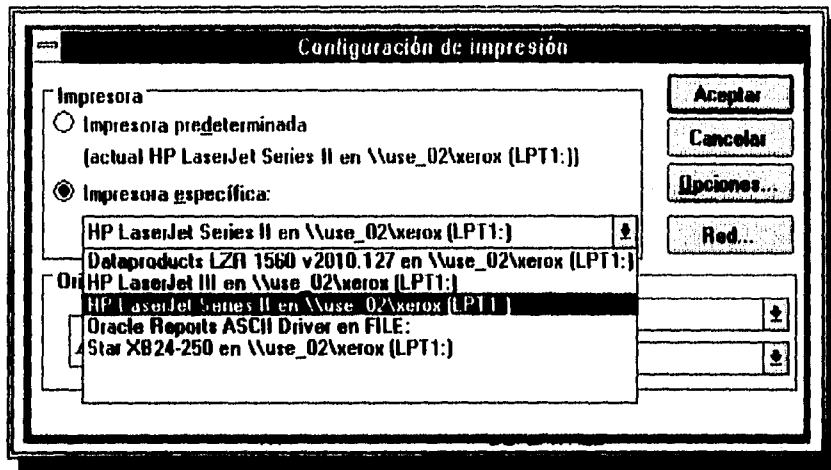


Figura 6.29 Pantalla de Configuración de Impresión

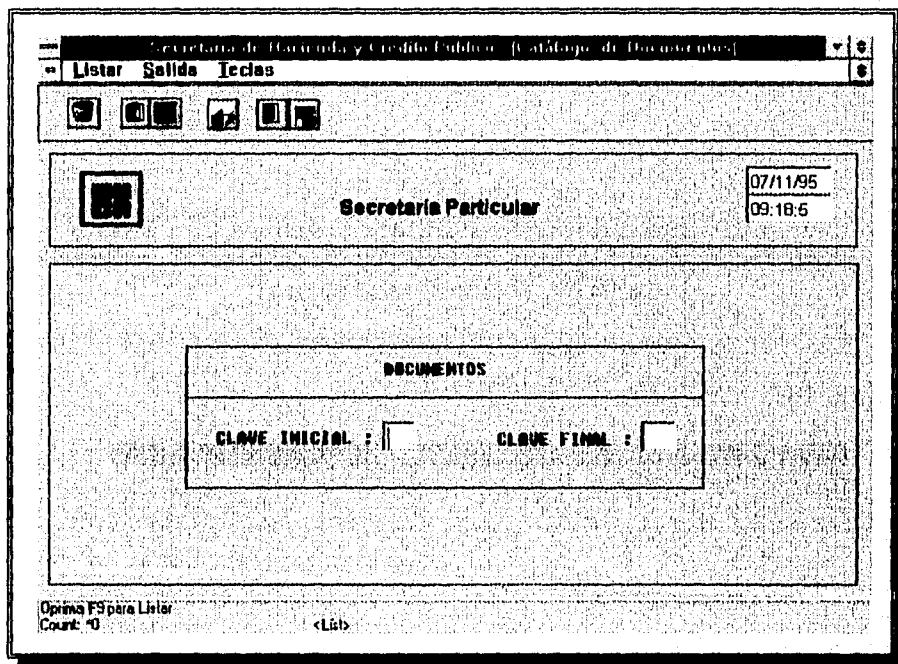


Figura 6.30 Catálogo de Documentos

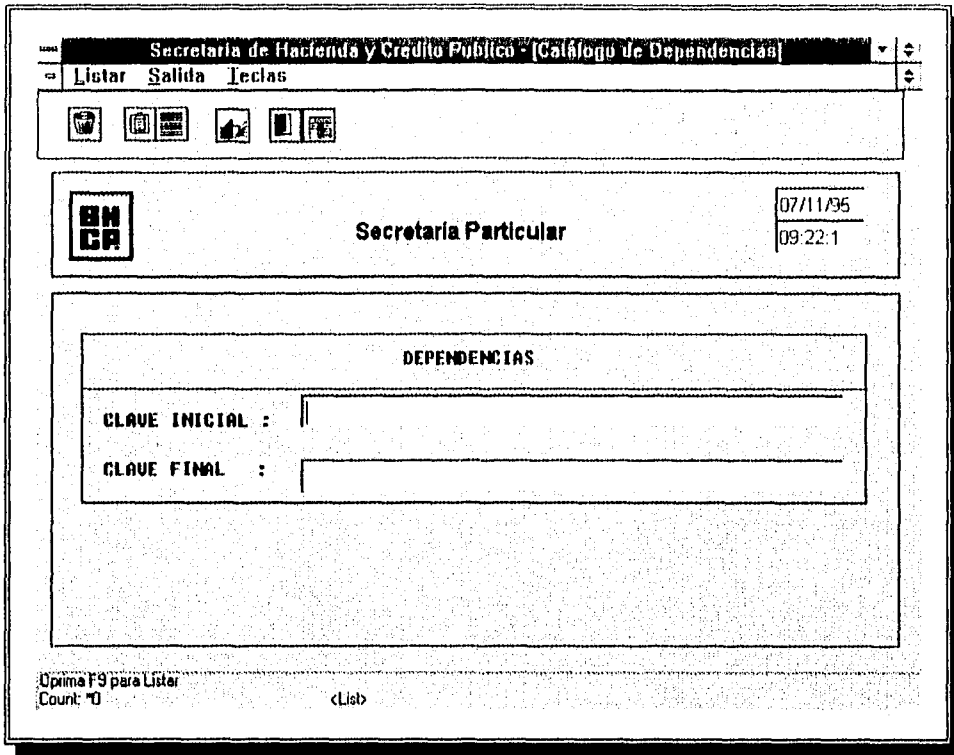


Figura 6.31 Catálogo de Dependencias

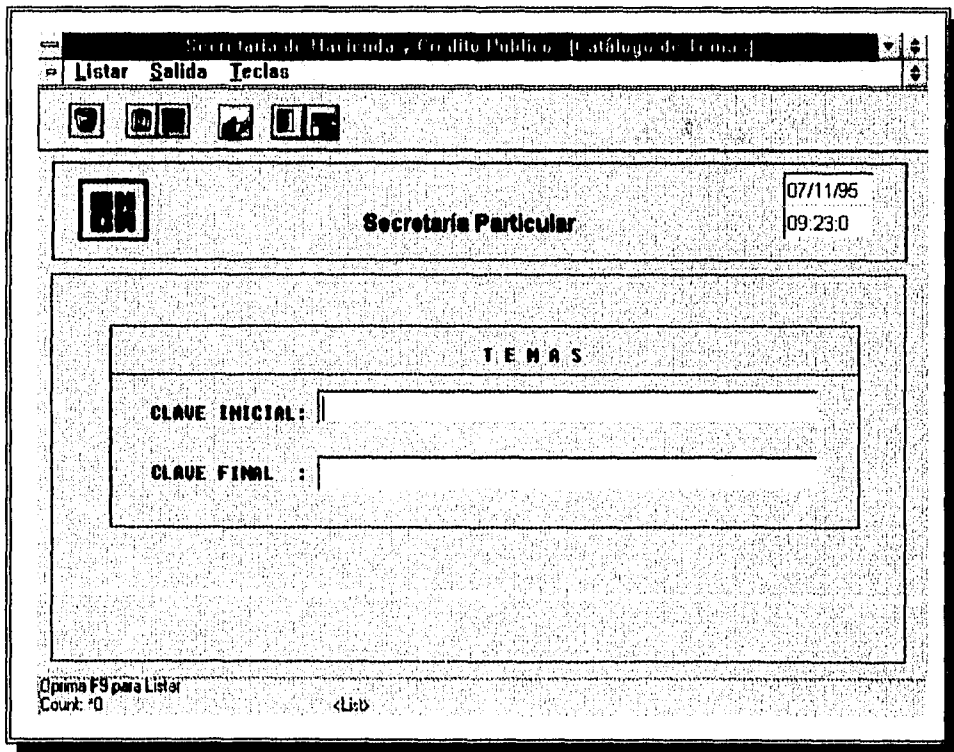


Figura 6.32 Catálogo de Temas

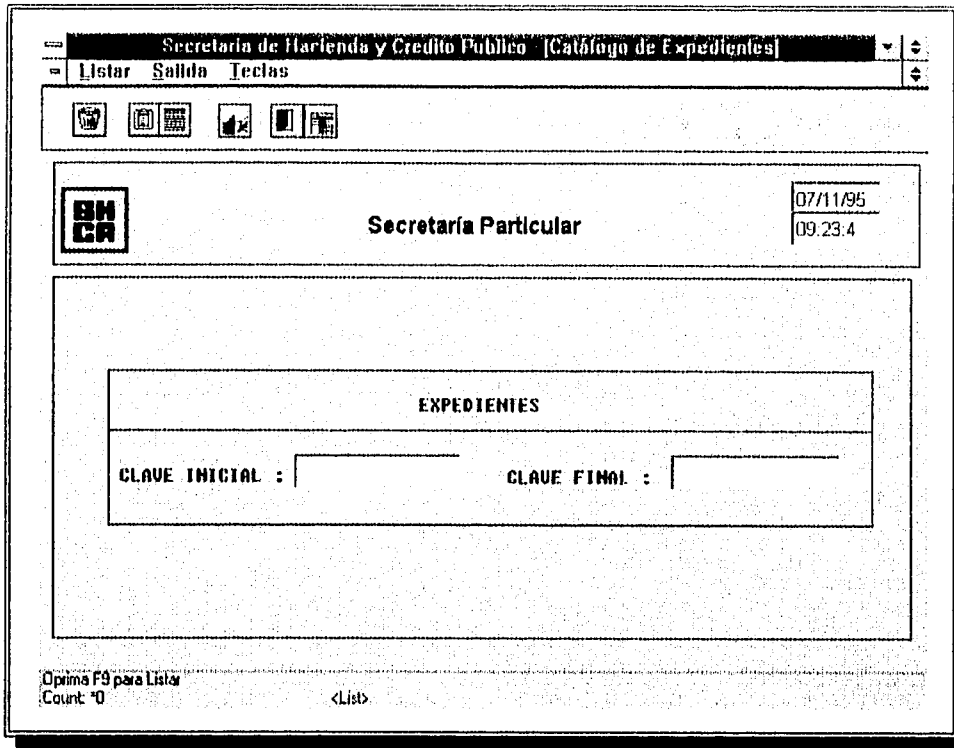


Figura 6.33 Catálogo de Expedientes

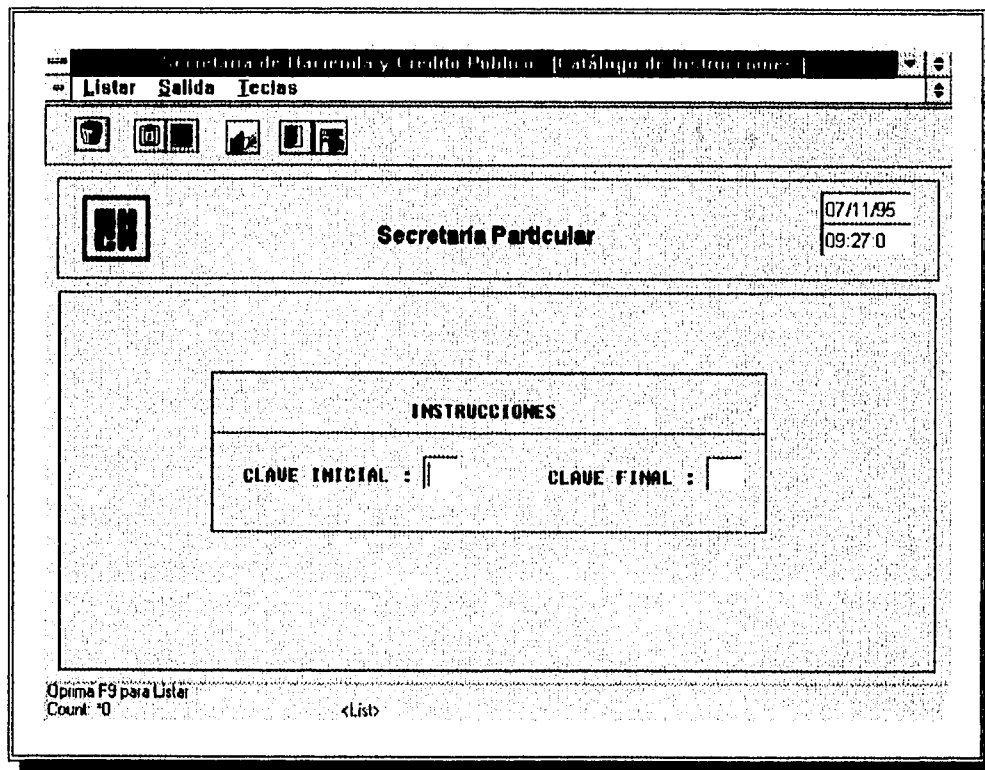


Figura 6.34 Catálogo de Instrucciones



En el caso del *Catálogo de Turnado*, se requieren los siguientes datos: clave, nombre, cargo y área, como se muestra en la figura 6.35. Se puede utilizar el comodín % según como al usuario le convenga, es decir, si desea obtener todos los registro cuya clave comienza con la letra 'A' se utiliza de la manera siguiente: A%. En el caso del nombre, si se desea obtener a aquellos turnados que alguno de sus apellidos sea "Padilla", el comodín se utiliza de la siguiente manera: %Padilla%, estos mismos principios se utilizan también para el área y el cargo.

Figura 6.35 Catálogo de Turnados

En esta pantalla se pueden generar tres distintos reportes, los cuales se listan en la figura 6.36.

Figura 6.36 Menú de Reportes de Turnados

Para obtener el "Reporte General", se debe proporcionar un título, el cual será requerido posteriormente de haber elegido este reporte. Se desplegarán Clave, Nombre, Cargo y Área, según se muestra en la figura 6.37.

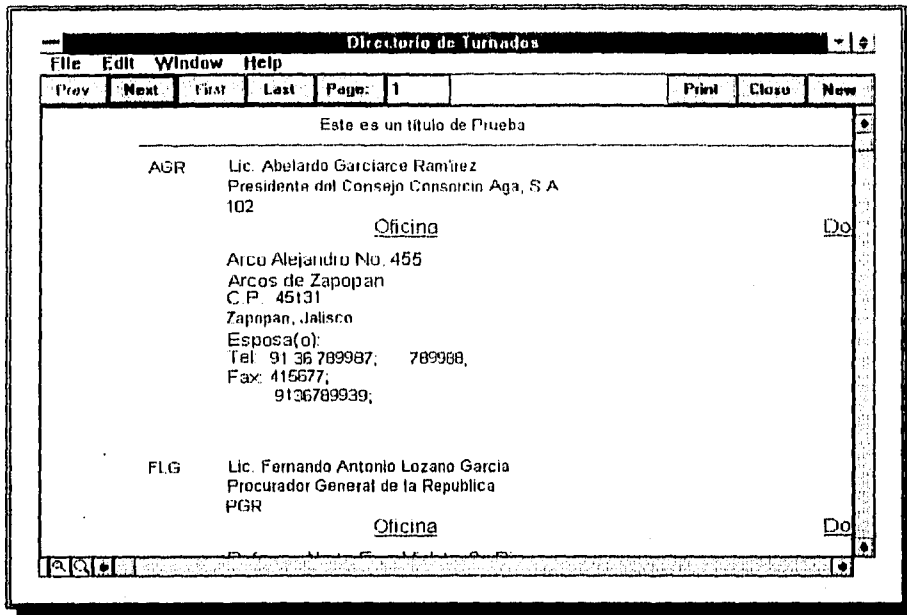


Figura 6.37 Reporte General de Turnados

Para el "Reporte de Catálogo", se generará inmediatamente después de la ventana mostrada en la figura 6.36, donde desplegará todos los campos del o los registros que se soliciten, tal como se muestra en la figura 6.38.

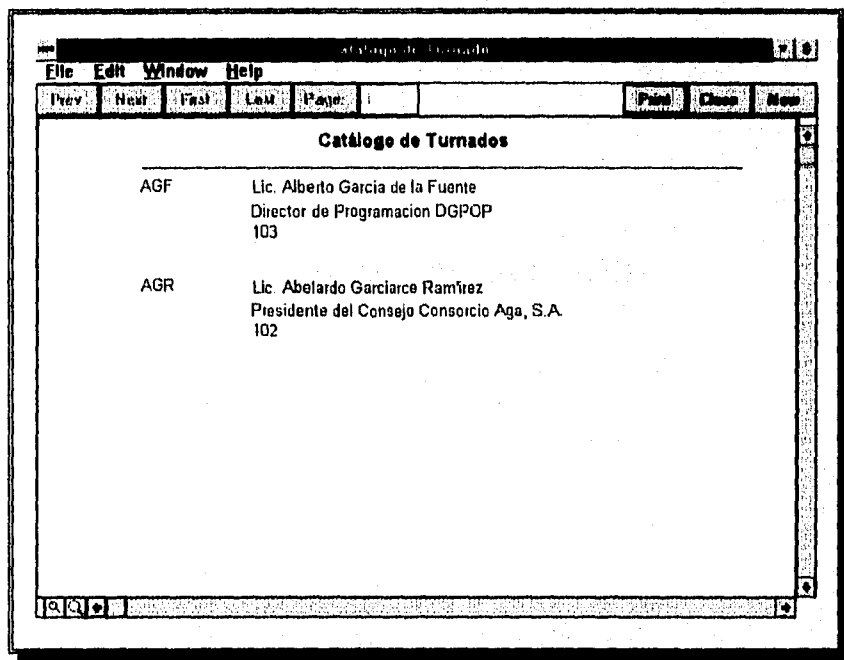


Figura 6.38 Reporte de Catálogo

Finalmente, el "Reporte de Etiquetas", tendrá la alternativa de imprimir 4 o 6 etiquetas, tal como se muestra en la figura 6.39, donde se presentarán los siguientes datos: nombre, cargo, domicilio completo como se observa en la figura 6.40, la información se podrá ver en pantalla. Si el usuario desea imprimir, únicamente cambiará las hojas de la impresora por etiquetas de 6 o 4 según sea el caso.

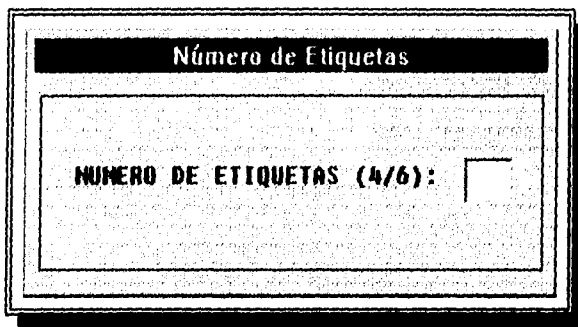


Figura 6.39 Menú de etiquetas

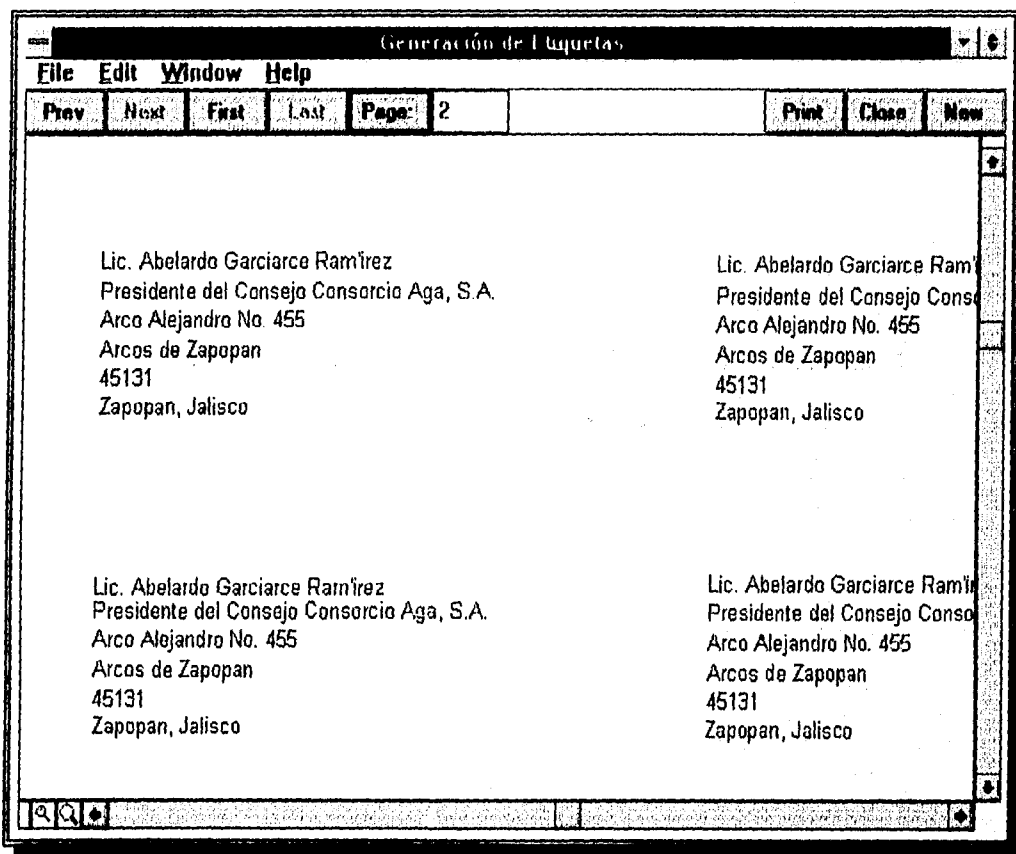


Figura 6.40 Pantalla de cuatro etiquetas

## 6.4 Documentos

La opción de Documentos se activa presionando **Alt** y la letra inicial de Documentos, es decir, la letra **D** se presentará el menú mostrado en la figura 6.41. Para activar cualquiera de los elementos de éste, es necesario oprimir la letra subrayada de cada uno de los elementos.

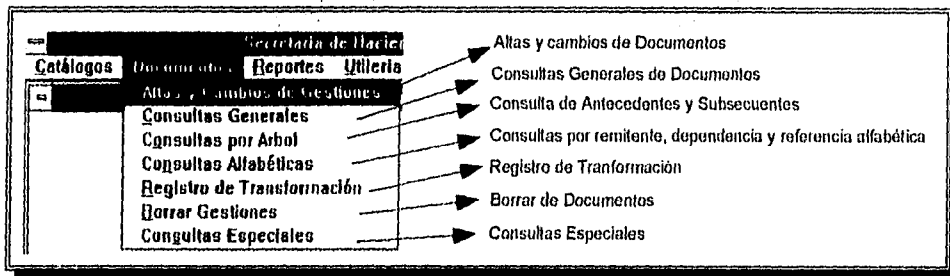


Figura 6.41 Menú de Documentos

Para acceder a *Altas y Cambios de Documentos* es necesario oprimir la tecla **A** y se presentará la pantalla que se muestra en la figura 6.42.

Esta pantalla permite registrar la documentación que comienza a referirse al tipo de documentos, es decir, si es gestión, oficio, invitación, regalos, etc; para poder listar las diferentes opciones que se muestran en la figura 6.43 se oprime **F9** o **Alt A** donde presentará el menú mostrado en la figura 6.44. Posteriormente se oprime la letra **L**, en caso de que se disponga de mouse se localiza el ícono de listar, la función de los iconos se muestra en la figura 6.45

Figura 6.42 Pantalla Completa de la pantalla de Altas y Cambios de Documentos

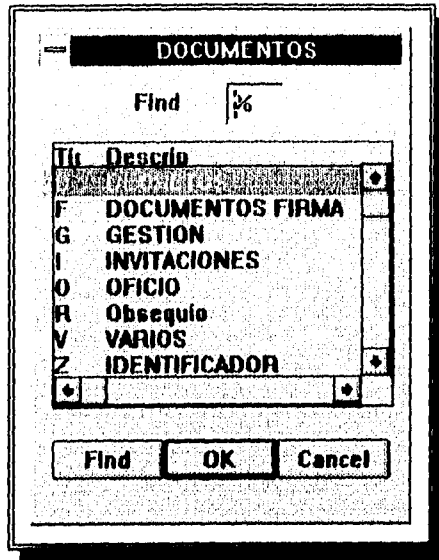


Figura 6.43 Catálogo de Documentos

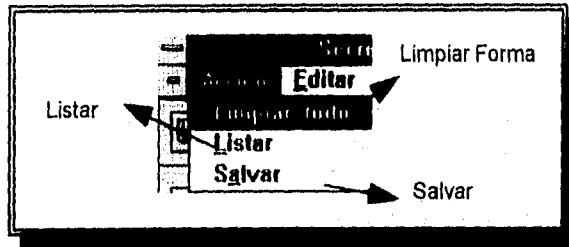


Figura 6.44 Menú de Acción

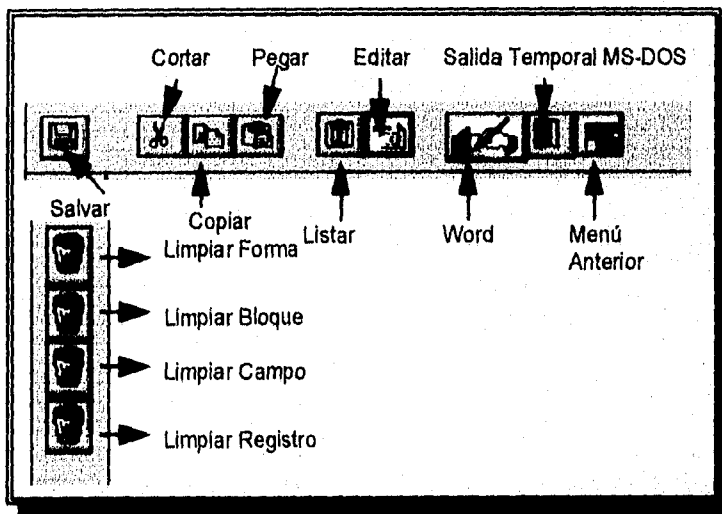


Figura 6.45 Grupo de Iconos de la pantalla de Altas y Cambios de Gestiones

Los campos que permite listar son clasificación figura 6.46, dependencia figura 6.47, tema figura 6.48, expediente figura 6.49, turno figura 6.50 e instrucción figura 6.51

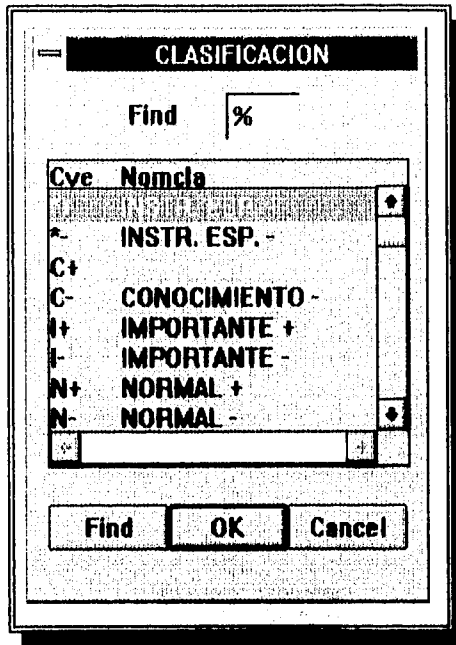


Figura 6.46 Menú de Clasificaciones

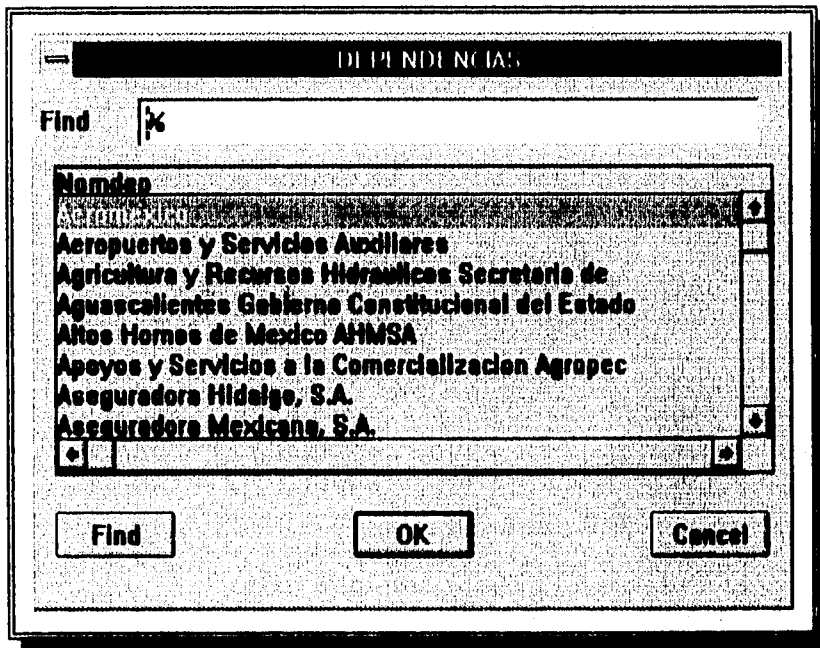


Figura 6.47 Menú de Dependencias

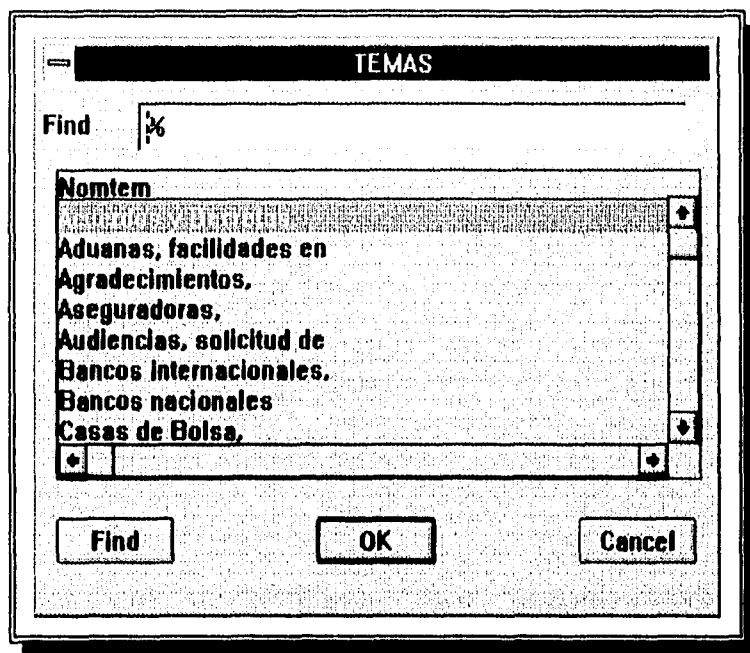


Figura 6.48 Menú de Temas

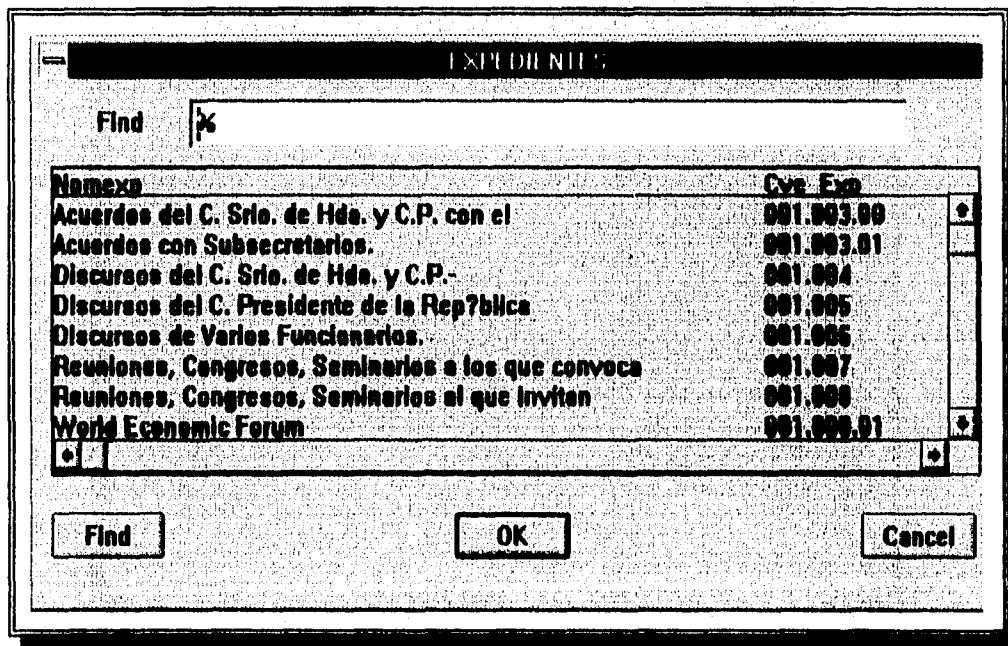


Figura 6.49 Menú de Expedientes

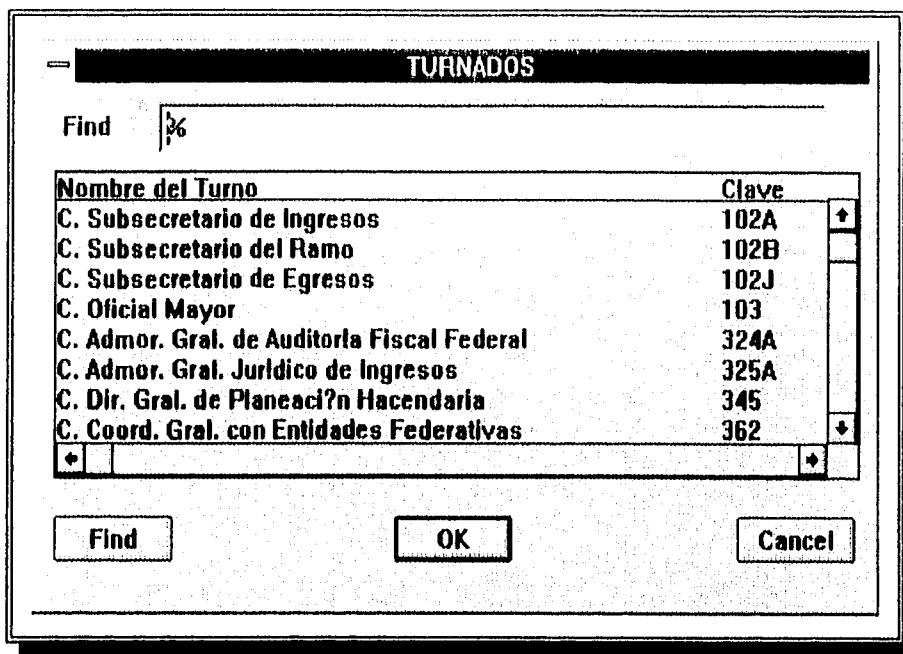


Figura 6.50 Menú de Turnados

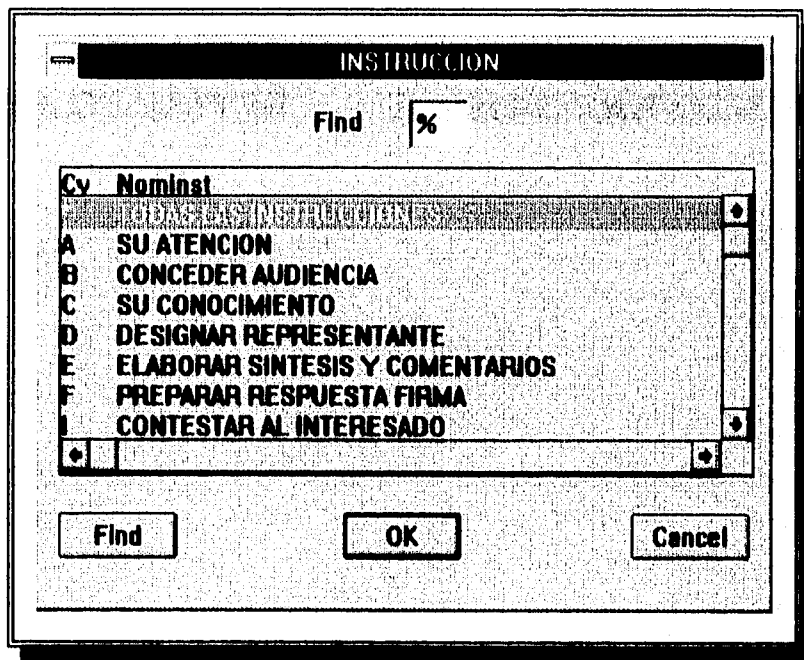


Figura 6.51 Menú de Instrucciones



Debemos recordar que estas listas manejan el comodín '%', el cual nos permite buscar un determinado número de registros colocando el signo al principio de la palabra o palabras, en medio o al final de éstas, posteriormente se oprime el botón de FIND, donde se desplegarán los registros que cumplen con la condición propuesta, cuando se oprime el botón OK, automáticamente cerrará la ventana y colocará el registro elegido en el campo correspondiente.

Para los campos asunto y resolución, se puede abrir un pequeño editor con las teclas **Ctrl** **e** o **Alt** **E**, en seguida se presentará el menú que se muestra en la figura 6.52, este permite manejar las funciones propias de windows: cortar, copiar, pegar un campo completo, almacenando este dato en el visor de portapapeles, además se puede emplear el ícono de editor, presentando el editor, el cual se muestra en la figura 6.53, aquí el usuario debe escribir el asunto o la resolución tal como desea que se presente en el reporte impreso.

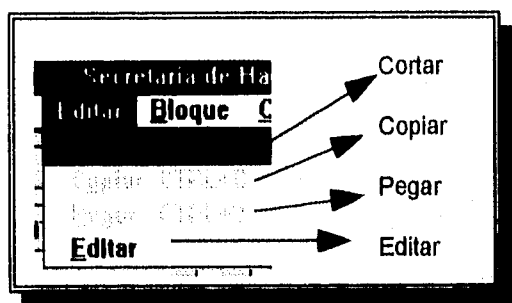


Figura 6.52 Menú de Editar

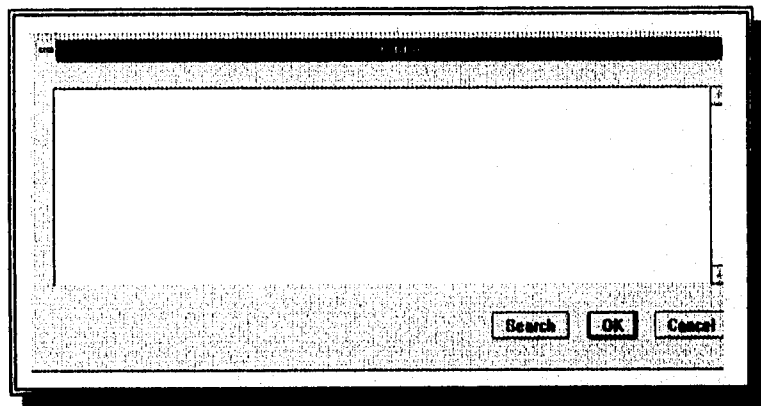


Figura 6.53 Editor de textos presentado en los campos de Asunto y Resolución

Otra función importante dentro de esta forma, es cuando el usuario tiene que generar un oficio para algún turnado, delegando la resolución del asunto lo puede copiar al visor de portapapeles, por medio del ícono acceder a word, aquí descargará lo que está guardado en el buffer o visor de portapapeles, únicamente escribirá los detalles que faltan para concluir un oficio.

En la figura 6.54, se observa el registro final de una captura normal, las fechas deben estar representadas por el día, mes y año separadas con una diagonal, existen campos que se deben de proporcionar, de lo contrario no permitirá desplazarse al siguiente campo.

Secretaría de Hacienda y Crédito Público (Control de Gestiones)

Acción Editar Bloque Campo Registro Consulta Salida Teclas

IDENTIFICADOR	TIPO	NUMERO	CLAS.	DOC.EXTERNO	ANTECEDENTE	SUBSECUENTE
1557	G	1234	Nr	AA-410		
F.DOCUM.	F.RECEP.	F.DESPACHO	F.DESCARGO	F.EVENTO	HORA	F.ARCHIVO
16/02/95	17/02/95	20/02/95	03/03/95			13/03/95
DEPENDENCIA: Asociacion Nacional de Facultades y Escuelas de Co						DIG
ASUNTO						
REMITENTE	Echenique Garcia Jose Antonio					
REF.AL.F.	Universidad Nacional Autonoma de Mexico					
TEMA	Designaciones, Informe de					
EXPEDIENTE	204					
TURNADO	CLAVE	INSTR	F.DESP.	F.DESC.		
	PN	L	20/02/95	03/03/95		
RESOLUCION						

Introducir datos, o presione <F5> para listar.  
Count: 1 v

Figura 6.54 Pantalla de un registro dado de alta

Para el caso de los menús bloque, campo y registro, permiten al usuario desplazarse al siguiente, anterior y borrar dependiendo del menú al cual nos referimos. En el caso de consultas, se puede extraer un grupo grande de registros. Para realizar las consultas se oprime **F7** y para la definición de la consulta **Alt** o **D**, se escriben las condiciones de los campos, los cuales realizarán las búsquedas para modificación o de tipo consulta, aquí se permite modificar a varios registros en una sola búsqueda desplazándose entre ellos por medio de las teclas de dirección **↓** **↑**. Se realizan las actualizaciones por medio del menú de acción u oprimiendo **F10** o manejando el ícono de salvar.

En el caso del menú de teclas, presenta al igual que los anteriores catálogos la ayuda de teclas, el cual se muestra en la figura 6.55.

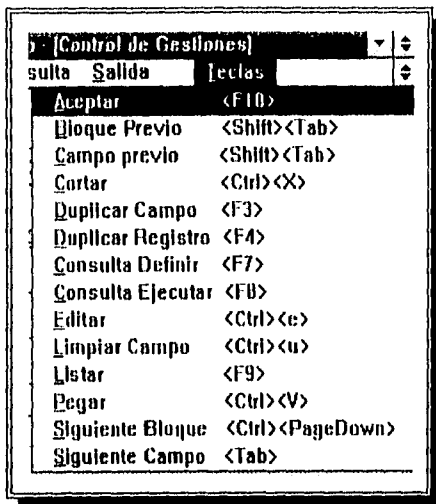



Figura 6.55 Menú de teclas

Las *Consultas Generales* es la segunda opción del menú de documentos, esta se muestra en la figura 6.56. Se puede escribir en cualquier campo formando una cierta condición, la cual se ejecutará por medio del ícono de Generación de Reporte o bien, se oprime  presentando automáticamente la pantalla mostrada en la figura 6.57, la cual espera la confirmación de la generación del reporte.

A screenshot of the 'CONSULTA GENERAL' screen in the SICADE application. The window title is 'Secretaría de Hacienda y Crédito Público [Consultas Generales]'. The menu bar shows 'Acción Salir Teclas'. The screen contains several input fields for search criteria:

- IDENTIFICADOR DE: 1233 A: 1235 TIPO CLASIF.
- NUMERO DE: A: DOC. EXTERNO
- F. DOCUM. DE: A: F. RECEP. DE: A:
- F. DESPACHO DE: A: F. DESCARGO DE: A:
- F. EVENTO DE: A: F. ARCHIVO DE: A:
- HORA DEPENDENCIA
- REMITENTE
- REF. ALF.
- TEMA
- EMPELENTE TURMADO A DISTRIC.

At the bottom left, there is a 'Count: 0' label.

Figura 6.56 Pantalla de Consultas Generales

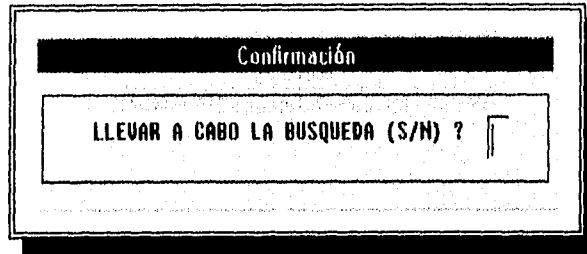


Figura 6.57 Confirmación de la búsqueda

La pantalla que mostrará el resultado de la búsqueda es la de la figura 6.58. Dentro de ésta se despliega un submenú, al oprimir la opción File, donde se puede configurar la impresora para los reportes impresos, también tiene íconos para determinar hasta qué página desea desplazarse, el tipo de reporte (el cual deseamos imprimir por impresora láser o de matriz), cerrar el reporte, desplazarse al primero o último registro.

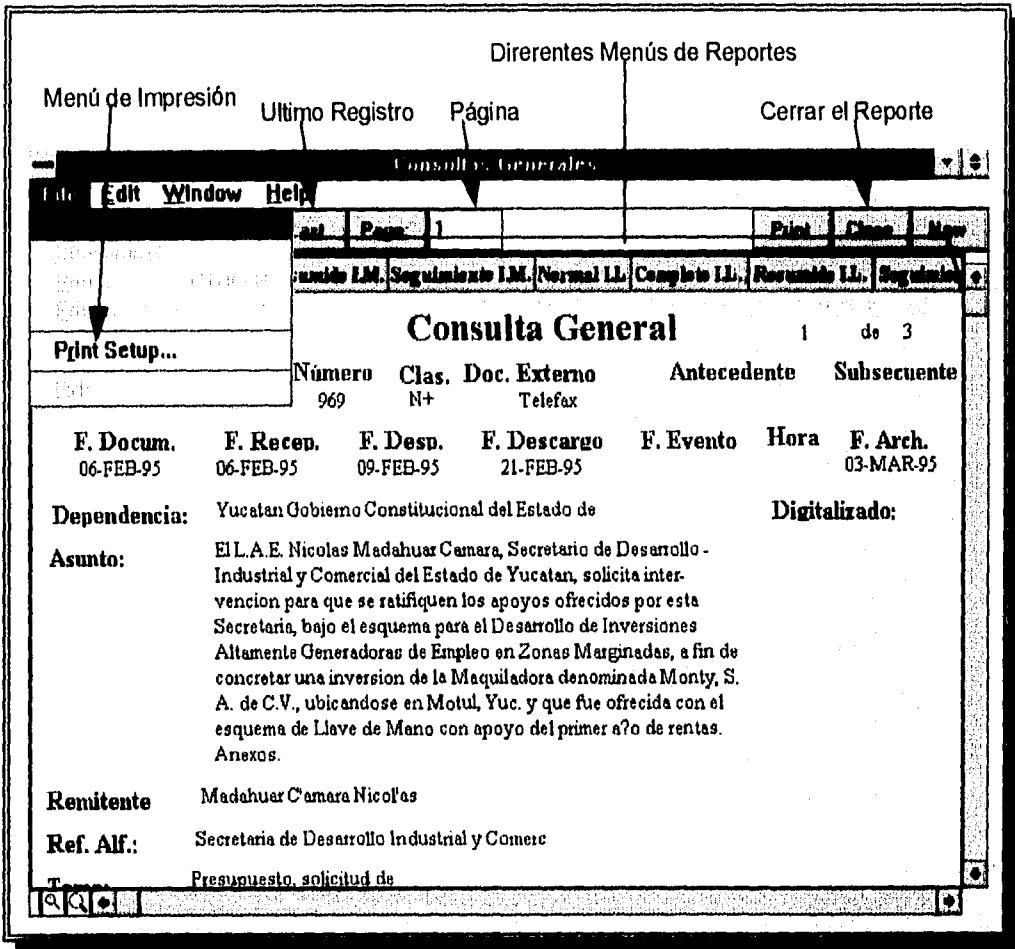


Figura 6.58 Generación de reporte por pantalla, presentación de distintos íconos

Para la configuración de impresora se tendrá un cuadro de diálogo como el que se ilustra en la figura 6.59, donde se requerirán datos de la impresora, orientación, tamaño de papel y el tipo de origen.

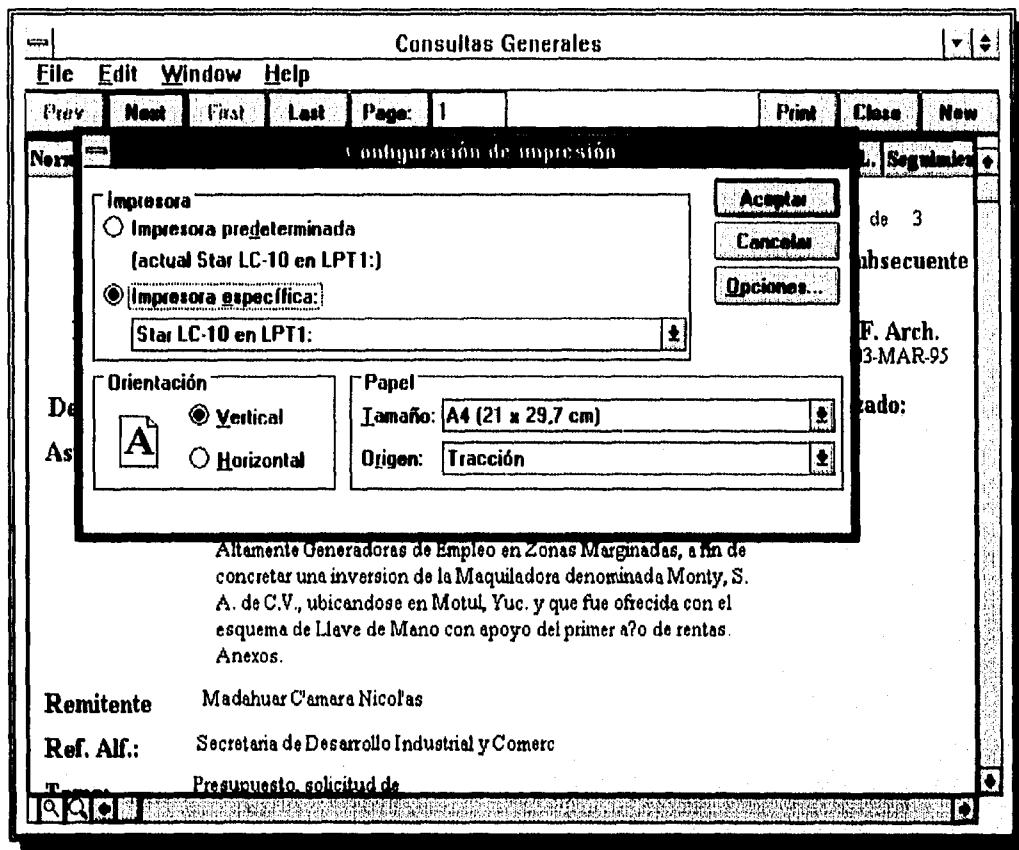


Figura 6.59 Presentación del manejo de impresión.

Para poder imprimir los distintos reportes de consultas generales, tales como: normal, completo, resumido y de seguimiento, existen íconos que se presentan en la pantalla de consultas generales.

En la figura 6.60 se presenta la pantalla de consultas alfabéticas, esta permite realizar búsquedas por medio de los campos siguientes: dependencia, referencia alfabética y remitente; en esta pantalla básicamente se proporcionará la palabra o palabras que deseamos buscar para cada uno de estos campos.

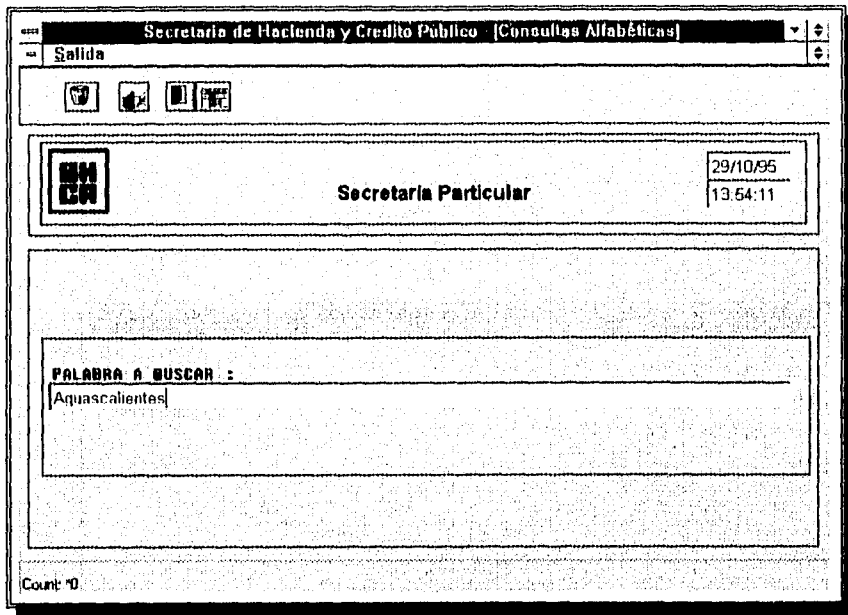


Figura 6.60 Consulta Alfabética

La pantalla mostrada en la figura 6.61, tiene las mismas características de la de Consultas Generales, es decir, permite generar los reportes normal, completo, resumido y de seguimiento, ya sea por impresora láser o de matriz. Además, permite configurar la impresora, desplazarnos de página en página o de registro en registro.

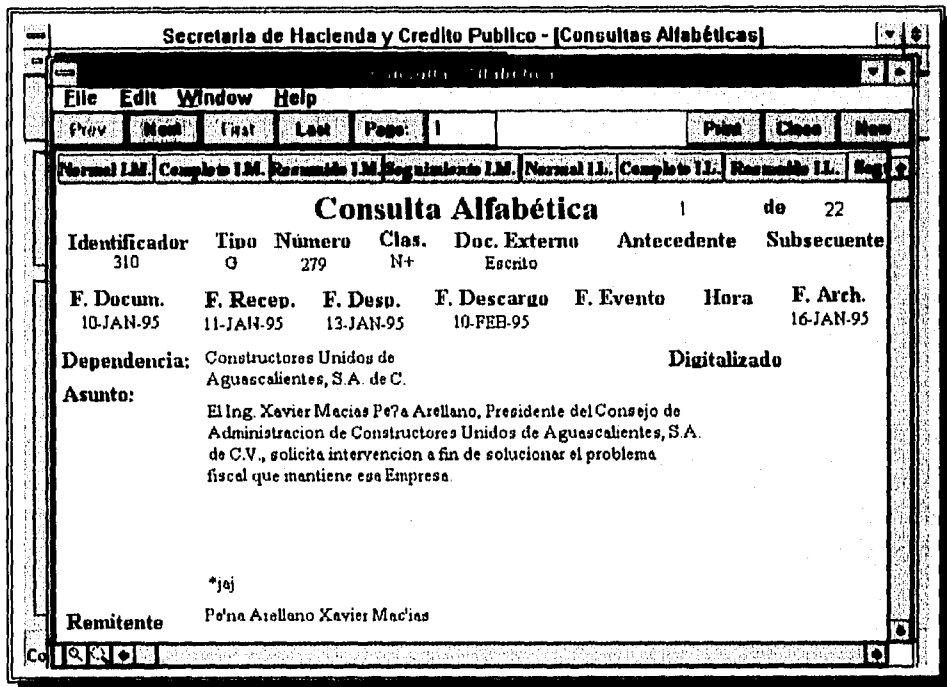


Figura 6.61 Pantalla de Consultas Alfabéticas

Se debe recordar que para que un documento tenga antecedente y subsecuente, se necesita lo siguiente: primero, que exista un primer documento con un asunto específico, el cual no se contestó en el tiempo establecido; segundo, la dependencia solicitante envía un segundo o tercer documento para recordar la contestación del primer documento.

Por lo tanto, tuvo que enviar dos o más documentos en relación a un mismo asunto. De acuerdo a lo anterior, la *Consulta por Árbol* pide un número de identificador, el cual es buscado con su respectivo antecedente y subsecuente. Así desplegará los campos necesarios para la consulta tal como se muestra en la figura 6.62

The screenshot shows a window titled 'Secretaría de Hacienda y Crédito Público [Consulta de Árbol]' with a 'Salida' menu. The main area displays a tree structure with three levels: ANTECEDENTE, DOCUMENTO ACTUAL, and SUBSECUENTE. Each level has fields for ID, TIPODOC, NUMDOC, CLASIF, REF, and TURNO. The 'DOCUMENTO ACTUAL' and 'SUBSECUENTE' levels are populated with specific data.

ANTECEDENTE											
ID	TIPODOC	NUMDOC	CLASIF	REF	TURNO						
DEPENDENCIA											
REMITENTE											
REFALF											
DOCUMENTO ACTUAL											
ID	123	TIPODOC	G	NUMDOC	58	CLASIF	1+	REF	Escrito	TURNO	PN
DEPENDENCIA	Grupo, Cigal S.A. de C.V.										
REMITENTE	Lanza Andrade Hector de la										
REFALF	Seminario Agropecuario'95										
SUBSECUENTE											
ID	5979	TIPODOC	0	NUMDOC	12356	CLASIF	1+	REF	Escrito	TURNO	
DEPENDENCIA	Grupo, Cigal S.A. de C.V.										
REMITENTE	Lanza Andrade Hector de la										
REFALF	Seminario Agropecuario'95										
No. DEL IDENTIFICADOR :											
Cant: 0											

Figura 6.62 Consultas de Árbol

En la figura 6.63 se muestra la pantalla de Registro de Transformación.

Esta pantalla permite generar el antecedente y subsecuente de un documento, aquí se debe proporcionar el número de identificador del documento, el cual va a ser antecedente, tipo de documento actual y el número.

Si los datos son correctos se generará el subsecuente, este pasará a ser el antecedente del segundo, esto se realiza con el fin de no volver a capturar la información del primer documento.

Figura 6.63 Registro de Transformación

El Registro de Transformación será capaz de identificar los documentos que tengan antecedente o subsecuente según sea el caso, si es así se desplegará un mensaje, el cual informará que no se puede generar el subsecuente porque ya existe un documento con estas características.

La opción de Bajas mostrada en la figura 6.64, como su nombre lo indica, permitirá borrar todos aquellos documentos que a consideración del usuario, resulten obsoletos.

Figura 6.64 Pantalla de bajas de gestiones





Para realizar la baja, se selecciona la opción correspondiente y se proporciona el número de identificador del documento que se desea borrar. Previamente se desplegarán los datos de acuerdo a la figura 6.64 al pulsar , éstos son útiles determinar si se debe borrar el registro. Finalmente se confirma la operación. La pantalla de *Consultas Especiales* se muestra en la figura 6.65, en ella el usuario realizará ciertas consultas para cada uno de los campos que presenta esta pantalla. Los documentos obtenidos se presentarán en pantalla, dando la opción de imprimirlos.

Figura 6.65 Pantalla de Consultas Especiales

En la figura 6.66, se presenta un ejemplo de salida en pantalla.

Figura 6.66 Reporte por pantalla de Consultas Especiales

## 6.5 Reportes

Cuando el usuario decide utilizar la opción de REPORTES, que se localiza en el menú principal, debe pulsar  para desglosar un submenú que contiene las siguientes funciones: impresión de turnos, documentos turnados, relación de discursos, programa de viajes, relación de oficios despachados, documentos para firma, proyectos de decreto, catalogo alfabético, relación de invitaciones y estadísticas. Estas opciones se muestran en la figura 6.67.

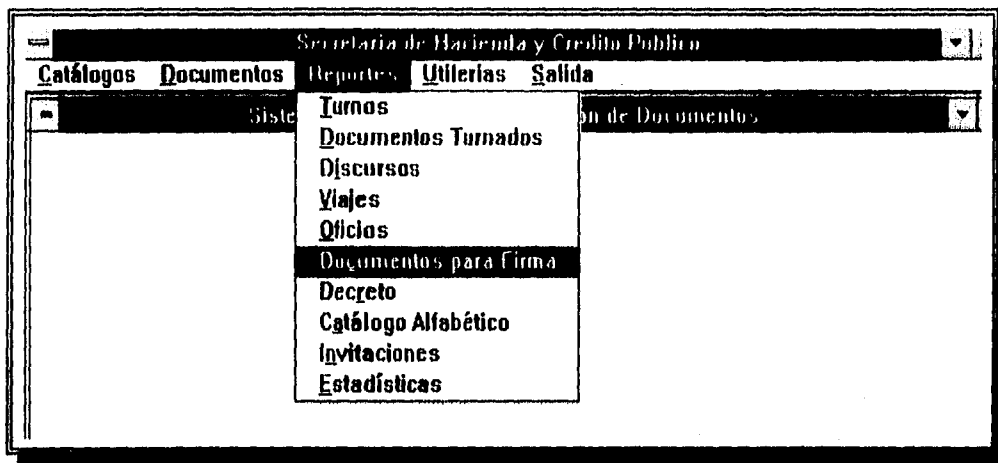






Figura 6.67 Menú de Reportes

Existen dos formas para poder acceder a cada una de estas opciones:

- Con las teclas de dirección   se localiza la opción deseada y se oprime , o
- Con la tecla de la letra que está subrayada dentro del menú.

En este menú, el usuario desarrollará las mismas tareas para todas las opciones, pues no hay variación en cuanto a la forma de trabajar.

El usuario eligirá la opción deseada y pulsará  para desplegar de una ventana solicitando los datos necesarios. Para imprimir el reporte requerido, es indispensable que el usuario otorgue todos los datos.

Se puede navegar entre los campos con las opciones del menú Acción que se muestra en la figura 6.68

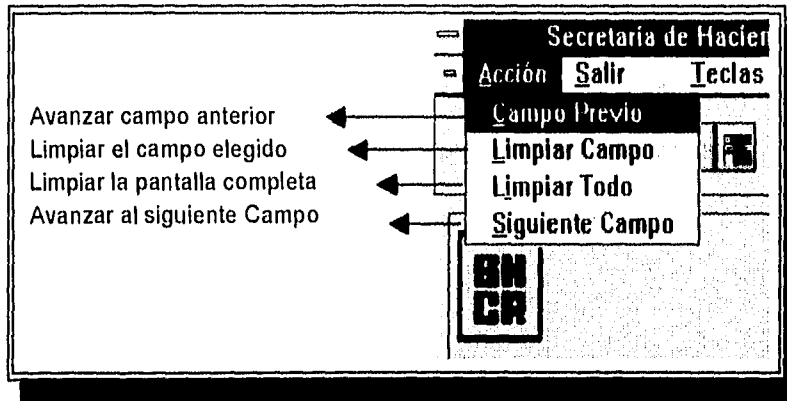


Figura 6.68 Navegación entre los Campos

Cada pantalla contiene íconos, éstos se muestran en la figura 6.69, permiten al usuario manejar distintas instrucciones con mayor facilidad; en la parte inferior de la pantalla, están presentes los mensajes principales ya sean de tipo informativo de las funciones de cada ícono, mensajes de las teclas principales de acuerdo al campo en donde se encuentre ubicado o desplegará el error si existe.

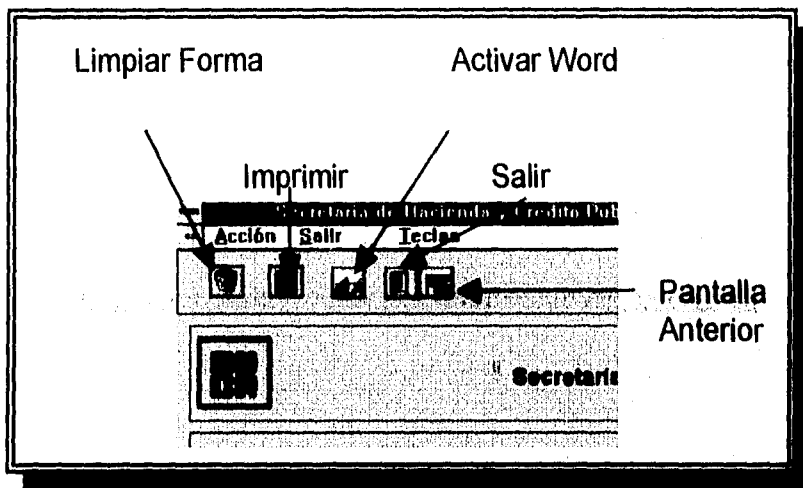


Figura 6.69 Funciones de los Íconos

Una vez seleccionado el reporte y otorgados los datos, aparecerá un mensaje preguntando si todos los datos están correctos, o bien, se puede hacer clic en el ícono de generación de reporte, si la respuesta es afirmativa se inicia la impresión, en el caso de que sea negativa, regresa a la ventana para poder corregir cualquier error.

A continuación presentamos las pantallas de los distintos reportes, que se manejan en esta opción, varían en el tipo de información que solicitan, pero el funcionamiento es el mismo.

Como se puede ver en la figura 6.70, la pantalla de impresión de Turnados muestra un menú, íconos y los campos que requiere un rango de documentos y distintos turnados.

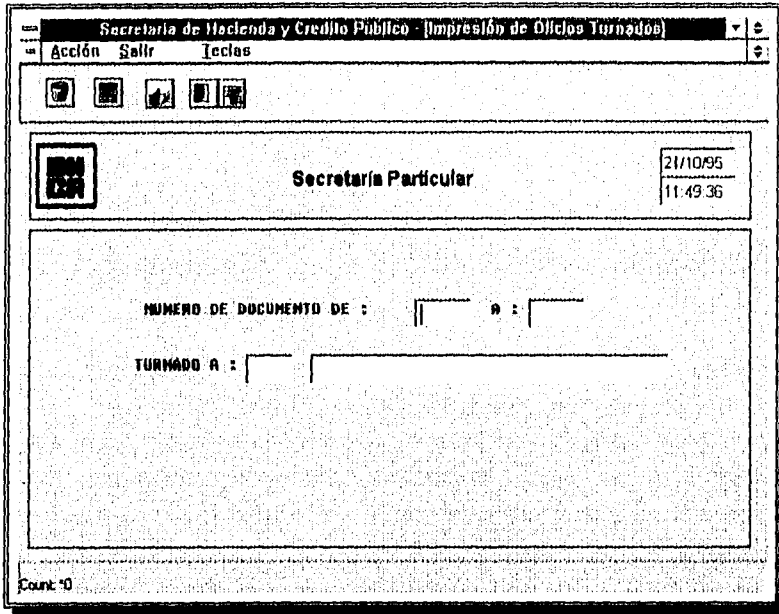


Figura 6.70 Pantalla de Impresión de Oficios Turnados.

Para el Reporte de Documentos Turnados es necesario proporcionar los datos que se solicitan en la pantalla mostrada en la figura 6.71, para que se generen cinco distintos tipos de reportes: documentos turnados sin y con resolución, invitaciones, audiencias y obsequios y regalos.

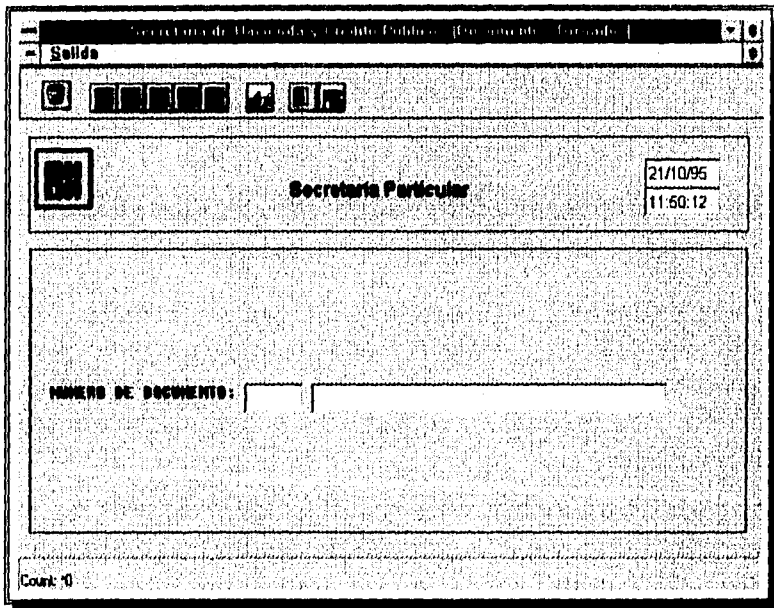


Figura 6.71 Documentos Turnados

La pantalla mostrada en la figura 6.72, presenta los datos que se requieren para generar el reporte de Relación de Discursos.

The screenshot shows a window titled 'Secretaría de Hacienda y Crédito Público [Relación de Discursos]'. The menu bar includes 'Acción', 'Salir', and 'Teclas'. The main header area contains the 'Secretaría Particular' logo and the date '21/10/95' with the time '11:50:53'. The form fields are as follows:

- NUMERO DE IDENTIFICADOR DE : [ ] A : [ ]
- FECHA DE RECEPCION DE : [ DD/MM/AA ] A : [ DD/MM/AA ]
- FECHA DE EVENTO DE : [ DD/MM/AA ] A : [ DD/MM/AA ]
- TIPO DE DOCUMENTO : [ ]
- TORNADO A : [ ]
- CLASIFICACION : [ ]
- INSTRUCCION : [ ]

Count: 0

Figura 6.72 Relación de Discursos

La pantalla mostrada en la figura 6.73, presenta los datos que se requieren para generar el reporte de Programas de Viajes.

The screenshot shows a window titled 'Secretaría de Hacienda y Crédito Público [Programa de Viajes]'. The menu bar includes 'Acción', 'Salir', and 'Teclas'. The main header area contains the 'Secretaría Particular' logo and the date '21/10/95' with the time '11:51:42'. The form fields are as follows:

- NUMERO DE IDENTIFICADOR DE : [ ] A : [ ]
- FECHA DE RECEPCION DE : [ DD/MM/AA ] A : [ DD/MM/AA ]
- FECHA DE EVENTO DE : [ DD/MM/AA ] A : [ DD/MM/AA ]
- TIPO DE DOCUMENTO : [ ]
- CLASIFICACION : [ ]

Count: 0

Figura 6.73 Programa de Viajes

La pantalla mostrada en la figura 6.74, presenta los datos que se requieren para generar el reporte de Relación de Oficios Despachados.

Figura 6.74 Relación de Oficios Despachados

La pantalla mostrada en la figura 6.75, presenta los datos que se requieren para generar el reporte de Documentos para Firma.

Figura 6.75 Documentos para Firma

La pantalla mostrada en la figura 6.76, presenta los datos que se requieren para generar el reporte de Relación de Proyectos de Decreto.

The screenshot shows a window titled 'Secretaría de Hacienda y Crédito Público - (Relación de Proyectos de Decreto)'. The menu bar includes 'Acción', 'Salir', and 'Teclas'. Below the menu is a toolbar with several icons. The main area contains a header with 'Secretaría Particular' and a date/time stamp '21/10/95 11:53:57'. The form fields are as follows:

NÚMERO DE DOCUMENTO DE :	<input type="text"/>	A:	<input type="text"/>
	DD/MM/AA		DD/MM/AA
FECHA DE RECEPCIÓN DE :	<input type="text"/>	A:	<input type="text"/>
TURNADO A :	<input type="text"/>		
CLASIFICACION :	<input type="text"/>		
INSTRUCCION :	<input type="text"/>		

At the bottom left, there is a 'Count: 0' label.

Figura 6.76 Relación de Proyectos de Decreto

La pantalla mostrada en la figura 6.77, presenta los datos que se requieren para generar el reporte de Catálogo Alfabético.

The screenshot shows a window titled 'Secretaría de Hacienda y Crédito Público - (Catálogo Alfabético)'. The menu bar includes 'Acción', 'Salir', and 'Teclas'. Below the menu is a toolbar with several icons. The main area contains a header with 'Secretaría Particular' and a date/time stamp '21/10/95 11:56:36'. The form fields are as follows:

NÚMERO DE IDENTIFICADOR DE :	<input type="text"/>	A:	<input type="text"/>
------------------------------	----------------------	----	----------------------

At the bottom left, there is a 'Count: 0' label.

Figura 6.77 Catálogo Alfabético

La pantalla mostrada en la figura 6.78, presenta los datos que se requieren para generar el reporte de Relación de Invitaciones.

Figura 6.78 Relación de Invitaciones

La pantalla mostrada en la figura 6.79, presenta los datos que se requieren para generar el reporte de relación de Estadísticas de Documentos.

Figura 6.79 Estadística de Documentos

Debe recordar que los reportes son impresos en impresora láser a excepción de impresión de Turnos.



## 6.6 Utilerías

La opción de Utilerías se activa presionando **Alt** y la letra inicial de Utilerías, es decir la letra **U**. Presentará el menú mostrado en la figura 6.80. Para poder activar cualquiera de estos elementos, es necesario oprimir la letra subrayada de cada uno de ellos.

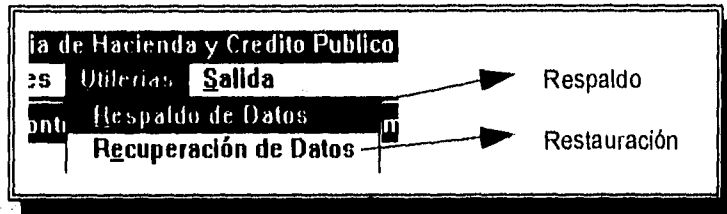


Figura 6.80 Menú de Utilerías.

## Respaldo de Datos

Para realizar un Respaldo de Datos es necesario oprimir **R**, esto presentará la pantalla que se muestra en la figura 6.81. Se solicitará la confirmación del administrador de la base de datos.

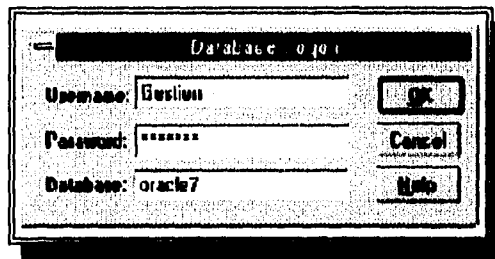


Figura 6.81 Pantalla de Confirmación del clave del administrador.

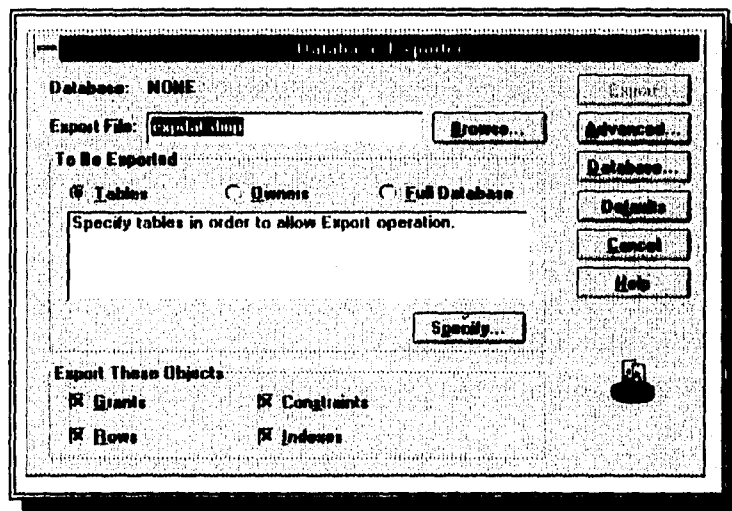


Figura 6.82. Pantalla principal de Respaldo de Datos

Posteriormente, se presentará la pantalla mostrada en la figura 6.82 en donde se apreciarán los siguientes requerimientos:

- **Export File<sup>1</sup>.**- Es el nombre del archivo, en donde se alojará el contenido de la base de datos.
- **To Be Export<sup>1</sup>.**- Es el tipo de respaldo que se va a realizar. Existen tres maneras distintas de hacer el respaldo: por tablas, usuarios o toda la base de datos. Dependiendo del tipo de respaldo, resaltará el ícono Specify, esto es concretamente para el caso de usuarios y tablas; aquí se puede elegir el tipo de usuario o tablas a respaldar, tal como se muestra en la figura 6.83

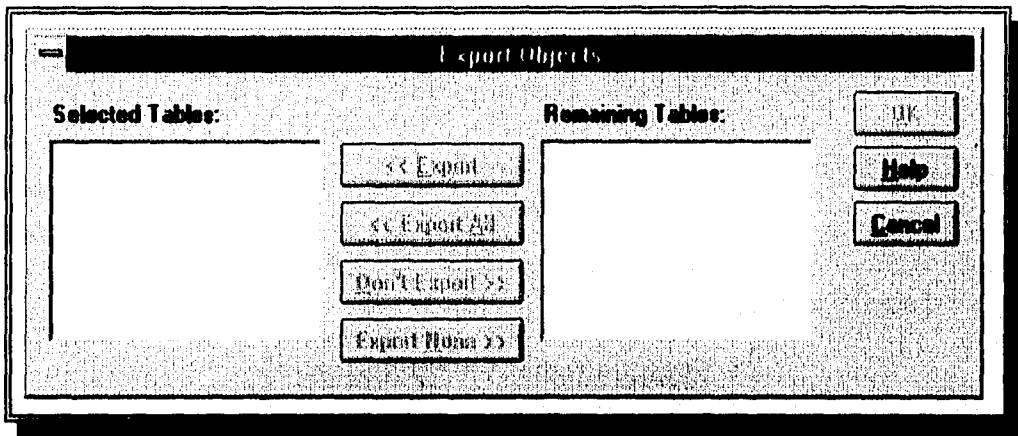


Figura 6.83 Pantalla de presentación de tablas o usuarios.

Aquí se elegirán las tablas o los usuarios a exportar, ya sean todos o solo algunos.

En la parte inferior se presenta el tipo de respaldo, por default están seleccionados todos y su significado es el siguiente:

- **Grants<sup>1</sup>.**- Algunos de los objetos que están contenidos en la base de datos tienen privilegios(GRANTS), estos privilegios se respaldarán.
- **Rows<sup>1</sup>.**- Respaldar los registros de cada una de las tablas seleccionadas, es decir, solo se puede respaldar la definición de las tablas seleccionadas con marcar esta opción.
- **Constraints<sup>1</sup>.**- Si se desean respaldar los parámetros de las restricciones (constraints) de la tabla y de columna, se marcará esta opción, en caso contrario no se guardarán dichos parámetros.
- **Indexes<sup>1</sup>.**- En esta opción se respaldarán cada uno de los índices creados para cada tabla.

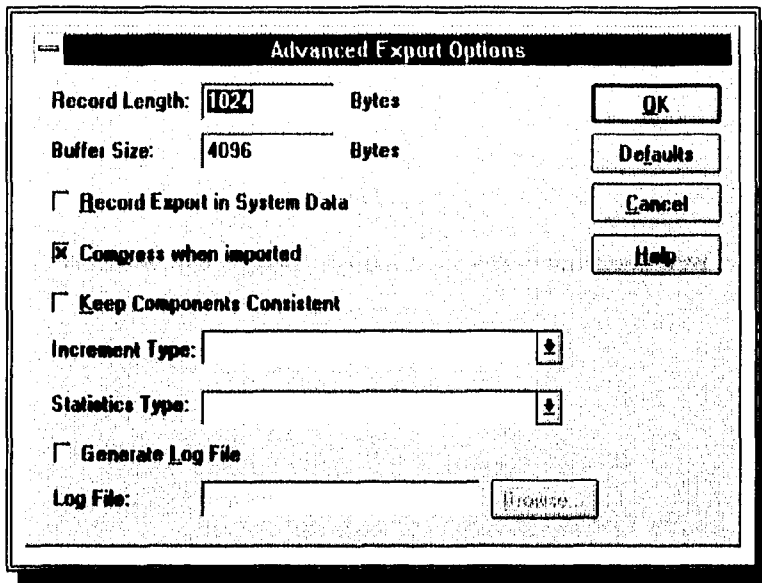


Figura 6.84 Pantalla de opciones avanzadas.

Al oprimir el ícono de Advanced se presentará la edición de cada uno de los parámetros avanzados; la pantalla correspondiente se presenta en la figura 6.84, y son los siguientes:


- Record Length<sup>1</sup>.- Escribe el valor en bytes para la longitud del archivo que contiene los registros de las tablas.
- Buffer Size<sup>1</sup>.- Se especifica el valor del tamaño del buffer.
- Record Export in System Data<sup>1</sup>.- Se selecciona si se desea que el proceso de respaldo se grabe en la base datos.
- Compress when imported<sup>1</sup>.- Se elige si se desea que la tabla de datos se compacte en un extent durante la restauración.
- Keep Components Consistent<sup>1</sup>.- Se selecciona si se desea leer la vista de la base al respaldar en forma consistente.
- Increment Type<sup>1</sup>.- Se selecciona el tipo de respaldo: Incremental, completo o acumulativo.
- Statistics Type<sup>1</sup>.- Se especifica el parámetro de estadística estimada y cálculo.
- Generate Log File<sup>1</sup>.- Se usa cuando se desea generar un archivo log durante el respaldo de la base.
- Log File<sup>1</sup>.- Es el nombre del archivo log.

El botón browse, sirve para buscar el path o ruta y el nombre del archivo log.

Para regresar a los valores estándar se oprime el botón de defaults.

Para activar el inicio del respaldo de la base de datos se oprime el botón EXPORT.

## Restauración de Datos

Para poder acceder a la Restauración de Datos es necesario oprimir , se presentará la pantalla que se muestra en la figura 6.81, aquí se requiere la confirmación del administrador de la base de datos. Para el caso de los usuarios esta pantalla no se presentará igual que en el caso anterior.

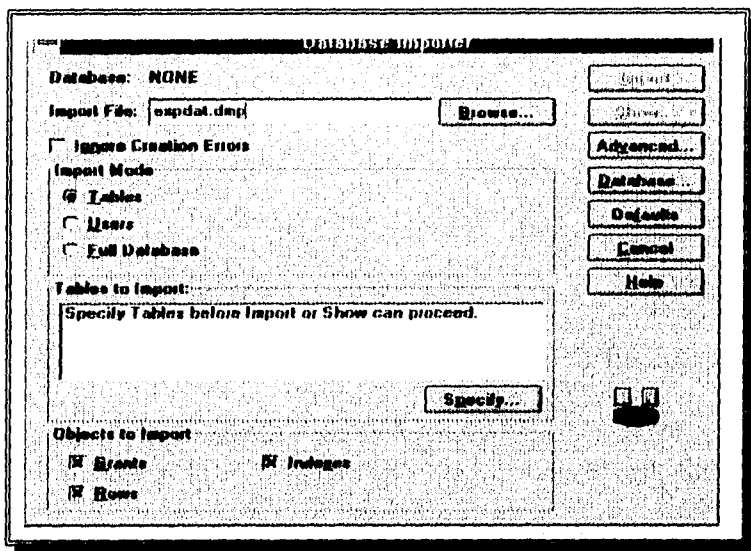


Figura 6.85 Pantalla principal de Restauración de Datos

Posteriormente, se presentará la pantalla mostrada en la figura 6.85 donde se tendrán las siguientes características:

- **Import File<sup>1</sup>.**- Es el nombre del archivo donde se aloja el respaldo de la base de datos.
- **Import Mode<sup>1</sup>.**- Es el modo de restauración que se va a realizar, es decir, de tablas, usuarios o base de datos completa.
- **Tables to import<sup>1</sup>.**- Son las tablas que se van a restaurar, ya sean todas o individualmente.

En la parte inferior presenta el tipo de restauración, por default están seleccionados todos, los cuales tienen el siguiente significado:

- **Grants<sup>1</sup>.**- Algunos de los objetos que están contenidos en la base de datos tienen privilegios(GRANTS), estos privilegios se respaldarán.
- **Rows<sup>1</sup>.**- Respalda los registros de cada una de las tablas seleccionadas, es decir solo se puede respaldar la definición de las tablas seleccionadas con marcar esta opción.
- **Indexes<sup>1</sup>.**- Con esta opción se respaldará cada uno de los índices creados para cada tabla.

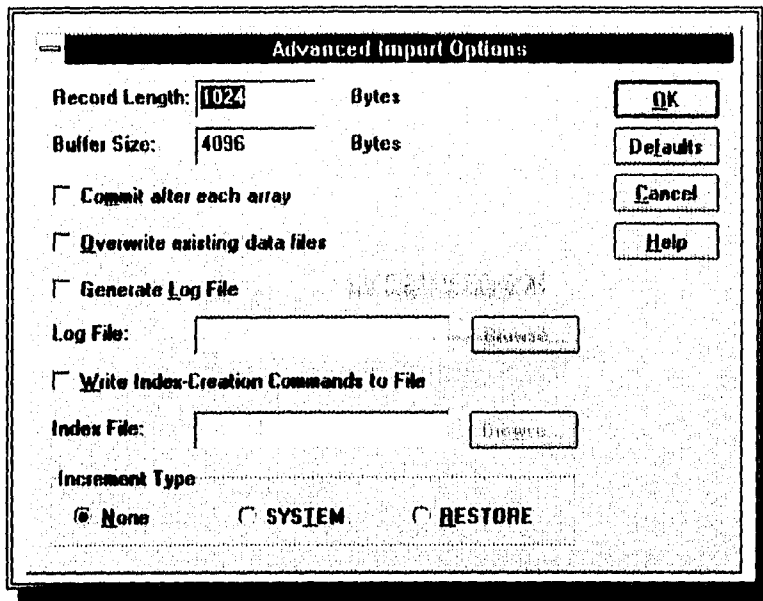


Figura 6.86 Pantalla de opciones avanzadas.

Al oprimir el ícono de Avanced, presentará la edición de cada uno de los parámetros avanzados, esta pantalla se presenta en la figura 6.86, y son los siguientes:

- Record Length<sup>1</sup>.- Escribe el valor en bytes para la longitud del archivo que contiene los registros de las tablas.
- Buffer Size<sup>1</sup>.- Se especifica el valor del tamaño del buffer.
- Commit After Each Array<sup>1</sup>.- Se elige si se desea que después de insertar un renglón se de aceptación (commit) de la operación.
- Overwrite Existing Database Files<sup>1</sup>.- Cuando el tipo de import es completo, esta opción se selecciona si se desea que se sobrescriba en uno ya existente al hacer la restauración.
- Generate Log File<sup>1</sup>.- Se genera un archivo log durante la restauración.
- Log File<sup>1</sup>.- El nombre del archivo log.  
El botón browse sirve para buscar el path o ruta y el nombre del archivo log.
- Write Index<sup>1</sup>.- Al seleccionarse se crea el archivo de comandos o sentencias SQL para la creación de índices.
- Index File<sup>1</sup>.- Nombre del archivo que contienen las sentencias de SQL para la creación de índices.
- Increment Type<sup>1</sup>.- Selecciona el tipo de import: SYSTEM, RESTORE, or None.

Cuando se termina la edición se oprime OK.

Para la ejecución de la restauración se oprime el botón de import.

## 6.7 Salir

La opción Salida se activa presionando **AK** y la letra inicial de Salida, es decir, con la letra **S**, se saldrá automáticamente del sistema. Esto se ilustra en la figura 6.87.



Figura 6.87 Salida del sistema

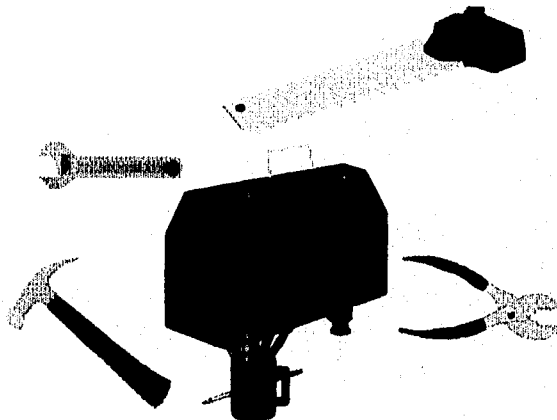
---

<sup>1</sup> La terminología no se ha traducido con el fin de no alterar el concepto de la palabra o frase.

# CAPÍTULO 7

## MANUAL DE INSTALACIÓN DEL SISTEMA DE CONTROL Y AUTOMATIZACIÓN DE DOCUMENTOS A GRAN ESCALA (SICADE)

---



El *Sistema de Control y Automatización de Documentos a gran Escala* (SICADE) permite las facilidades de clasificación, seguimiento y procesamiento dependiendo del carácter con que son recibidos.

Este manual nos ofrece los mecanismos de instalar e integrar SICADE a un ambiente operativo que se proporcionó para la implantación del mismo.

Debido a que SICADE requiere del almacenamiento masivo de información, además del control y la automatización, se requiere de una herramienta robusta, que permita la integridad, seguridad y fácil acceso a la información. Con los antecedentes antes descritos un Sistema de Administración de Base de Datos Relacionales permite obtener el mayor rendimiento y operatividad de SICADE, debido a los intereses de integración de información y la propia rentabilidad de dispositivos de almacenamiento, se requiere de un Manejador de Base de Datos que permita la portabilidad, transparencia y distribución de información desde un ambiente monousuario hasta una red de área amplia con datos distribuidos. Por otro lado debido al manejo de grandes volúmenes de información que se manejarán con SICADE, el mantenimiento de la base, el soporte y la atención a usuarios en México, debe cumplir con ser un producto sumamente comercial para futuros cambios y actualizaciones en relación directa con la funcionalidad de SICADE; la plataforma del manejador de base datos seleccionada es *Oracle*, ya que también permite su instalación y ejecución en la gran mayoría de los sistemas operativos comerciales y actuales.

Es importante recalcar cuando se requiera implementar SICADE en un ambiente de sistemas abiertos la información afectada por las transacciones es almacenada en una base de datos distribuida por lo general y puede ser accesado y/o cambiado en tiempo real por ejecución de estas transacciones. Cada ambiente computacional es referido como Sistema de Procesamiento de Transacciones en Línea (OLTP) lo que nos permite obtener un alto grado de seguridad e integridad de información.

Para fines prácticos se considera la instalación de SICADE en un ambiente monousuario, haciendo referencias, aclaraciones y cambios para los ambientes de sistemas abiertos.

## **7.1 Componentes y Arquitectura de Hardware**

Existen dos plataformas, en las cuales podemos instalar el sistema, una es enfocada a un ambiente de conectividad *en red*, y la otra es por medio de una estación de trabajo *monousuario*.

### **ESTACIONES DE TRABAJO (Implementación monousuario)**

Los requerimientos de hardware comprenden una computadora personal IBM, COMPAQ, u otra 100% compatible, con las siguientes características:



- Una unidad de proceso central (CPU) 80486 o Pentium.
- 16 MB en RAM.
- 500 MB en Disco Duro.
- Un manejador de Discos compactos, funcionando como un manejador lógico.
- Un manejador de Discos flexibles, de 3.5 pulgadas y alta densidad.
- Corriendo el Sistema operativo Windows 3.1 o Windows para Grupos de Trabajo
- Corriendo el Sistema Operativo DOS con Windows 3.1 o Windows para Trabajo de Grupos 3.11.

### **SERVIDORES (Implementación de red)**

Debido a los requerimientos de SICADE las plataformas y sistemas operativos descritos a continuación, podrán soportar sus aplicaciones:

- AViiON (corriendo DG/UX)
- Groupe Bull
- ICL
- Motorola
- NEC (Sun SPARC corriendo SUN OS y Solaris)
- IBM RS/6000 (corriendo AIX)
- DEC Alpha (corriendo OSF/1)
- DEC VAX y Micro VAX (corriendo VMS, UNIX y ULTRIX)
- HP (corriendo HP/UX)
- NCR (corriendo UNIX V.4)
- Data General corriendo SVR4: UNIX System V Release 4)
- Pyramid ES (corriendo CO62)
- Sequent (corriendo PTX y Dynix)
- Silicon Graphics Incorporated (corriendo IRIX)
- Stratus FTX (corriendo FTX)
- Procesadores Intel (corriendo SCO UNIX, Novell UnixWare, Novell NetWare, IBM OS/2 y Windows NT).

## **7.2 Arquitectura de Software**

El termino cliente/servidor originalmente es aplicado a la arquitectura de software que describe el procesamiento entre dos programas, es decir, una aplicación y un soporte de servicio. Al mismo tiempo, el programa del cliente y el programa del servidor no tendrán que estar físicamente separados; estos fueron llamados o invocados para un propósito en especial y llaman o invocan otros programas corriendo en la misma maquina. El concepto de estos programas será capaz de proveer servicios o administración de recursos en favor de un numero de otros programas que son aceptados ampliamente. El modelo computacional cliente/servidor representa una instancia especifica de procesamientos cooperativos distribuidos, donde la relación de clientes y servidores también involucra la relación entre los componentes de hardware y software. SICADE basa su funcionamiento en este modelo computacional.

El modelo computacional cliente/servidor cubre un amplio rango de funciones, servicios y otros aspectos de ambientes distribuidos. El uso relevante de este modelo se encuentra en topologías de redes de área local, redes de área amplia, datos distribuidos, procesamientos distribuidos (aplicación y presentación), procesamiento y administración de transacciones distribuidas y sistemas abiertos. Este modelo implica un procesamiento cooperativo de peticiones realizadas por un cliente, hacia un servidor; el cual procesara la petición y retornara los resultados al cliente.

El proceso cooperativo cliente/servidor es realmente una forma especial de procesamiento distribuido, en el cual los recursos (y tareas que afectan los recursos) son desplegadas a través de dos o mas sistemas computacionales discretamente. Mientras los sistemas distribuidos son un fenómeno relativamente nuevo, la operación de una distribución estratificada por niveles es mucho mas conocida, como ampliamente usada. Un ejemplo puede ser la función de distribución aritmética de entrada/salida en una unidad de proceso central (CPU) y un controlador de canal de entrada/salida.

Otros ejemplos incluyen la distribución del control de funciones de una red entre un host IBM corriendo su tradicional método de Acceso de Telecomunicaciones Virtual (VTAM) y un controlador de comunicaciones corriendo un proceso de control de redes (NCP), o la distribución de funciones del sistema operativo entre múltiples procesadores, como es el caso del IBM 3090/600 (quien tiene seis procesadores).

*Los sistemas distribuidos* abarcan desde el ambiente mas primitivo para soportar el procesamiento de una aplicación. Esto es basado en el ambiente de procesamiento de un Host ya que este no tendrá ninguna características de procesamiento de alguna aplicación distribuida.

El procesamiento de las aplicaciones basados en un Host, genera que únicamente el desempeño o rendimiento del procesamiento se realice únicamente en el Host, generando un sistema totalmente no distribuido, como podemos observar en la figura 7.1

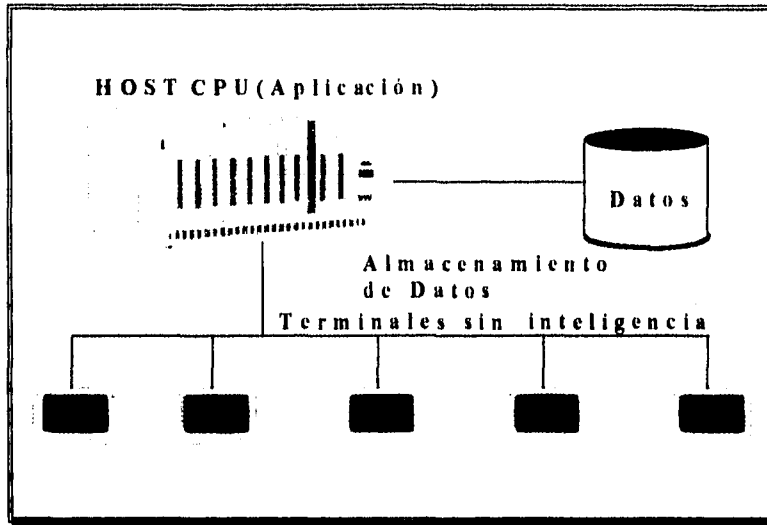


Figura 7.1 Ambiente de Procesamiento Basado en un Host.

El siguiente nivel superior de procesamiento para aplicaciones distribuidas es el *procesamiento maestro/esclavo*. Como su nombre lo especifica, las computadoras esclavas son asignadas hacia la computadora maestra y el rendimiento nuevamente recae sobre el Host. Esto se puede escenificar, con el ejemplo de los sistemas IBM 30XX, donde las computadoras esclavas, realmente son controladores de cluster o terminales inteligentes, con capacidad limitada de procesamiento, tal como la validación de campos en la pantalla, edición y algunas funciones asignadas para el procesamiento en su teclado, como se muestra en la figura 7.2

El modelo de *procesamiento de dispositivos compartidos* tuvo que emerger como un nivel superior sobre el modelo *maestro/esclavo*, típicamente se ejemplifica en las redes de área local (LAN).

En los ambientes de procesamiento de dispositivos compartidos donde se comparten recursos o dispositivos, las computadoras personales son asignadas al sistema y se representan como un recurso mas de la red. Dentro de la terminología de las redes locales, cada dispositivo compartido es llamado *servidor* (un servidor de archivos o un servidor de impresión, como ejemplo). El nombre de *servidor* es apropiado, desde que estos dispositivos compartidos son usados para recibir peticiones desde cualquier PC genérica, mediante funciones de bajo nivel. Un ejemplo de procesamiento con dispositivos compartidos puede ser *Novell's NetWare* o *Microsoft's LAN Manager*, estos contienen una red de área local que tendrá exclusivamente un sistema dedicado para servicios de archivos y/o impresiones, como podemos ver en la figura 7.3

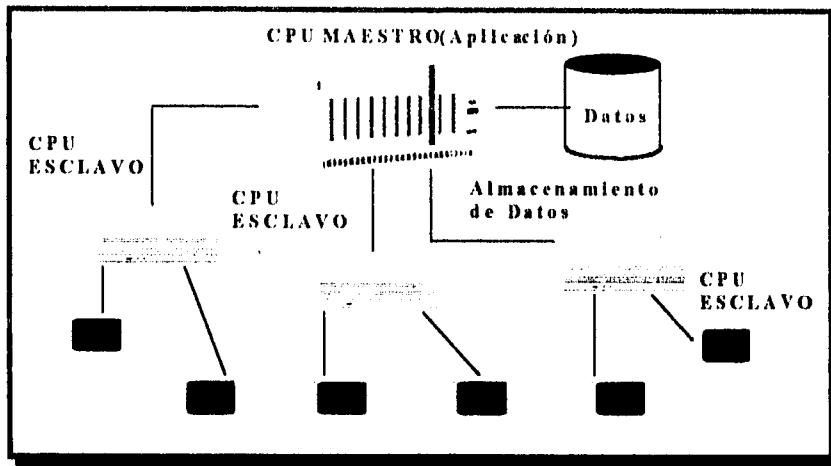


Figura 7.2 Ambiente de Procesamiento Maestro/esclavo.

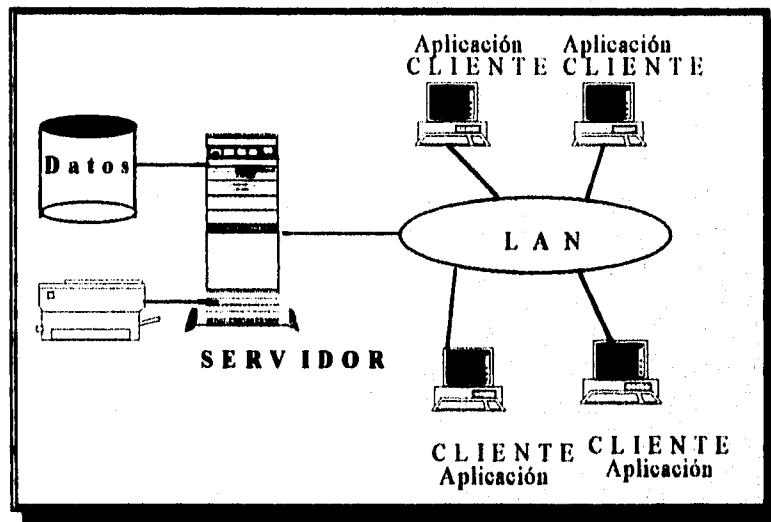


Figura 7.3 Ambiente de Procesamiento de dispositivos compartidos.

El modelo de procesamiento de *cliente/servidor* es una extensión natural del procesamiento de dispositivos compartidos, ya que cuando crece en el número de estaciones de trabajo también crecen los servidores de archivos e impresión en capacidad y poder. Llegando al punto en que las estaciones de trabajo cambian su comportamiento, hasta convertirse en clientes de los servidores.

Por lo que en el modelo de procesamiento cliente/servidor, la aplicación del procesamiento es dividida entre los clientes y los servidores. Un ejemplo muy ilustrativo es la búsqueda de un archivo, en los ambientes maestro/esclavo las terminales inteligentes generan los comandos necesarios para la localización del

archivo, pero en el ambiente cliente/servidor tendrá la característica, en donde el cliente podrá realizar la búsqueda sobre los recursos de el servidor y el servidor únicamente intervendrá para verificar rutas y accesos, como se muestra en la figura 7.4. Una ventaja que se puede observar será ilustrada por la comparación entre un servidor de impresión y un servidor de base de datos. Si una aplicación en una PC requiere unos registros en particular desde un archivo compartido, esta enviara una petición para leer el archivo entero desde el servidor de archivos, para que después sea enviado el archivo completo hacia la PC, lo que genera una sobrecarga para la red. En el caso del servidor de base de datos, en una aplicación que este corriendo en una PC enviara una petición de lectura de registro hacia el servidor, lo que el servidor realizara y ejecutara devolviendo únicamente los registros requeridos. Con esto podemos determinar que un ambiente de procesamiento cliente/servidor requiere:

- Comunicación relevantemente robusta entre los clientes y los servidores.
- Cooperación iterativa cliente/servidor que sea iniciada por el cliente.
- Distribución del procesamiento de la aplicación entre un cliente y cualquier servidor.
- Control total del servidor sobre los servicios y datos que pueda ser pedidos por los clientes.
- Arbitraje básico de los servidores en los conflictos de peticiones de los clientes.

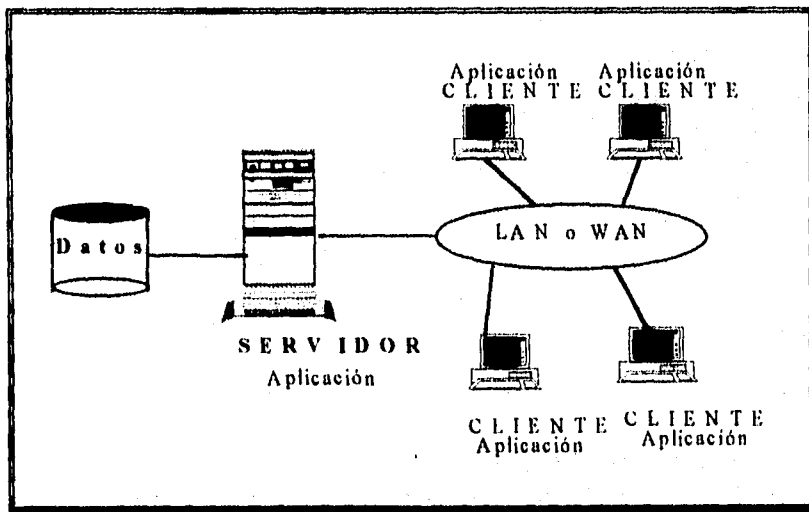


Figura 7.4 Ambiente de Procesamiento Cliente/servidor

El modelo de procesamiento *cliente/servidor* distingue entre los clientes que realizan una petición hacia los servidores y los servidores que prestan el servicio para atender las peticiones de los clientes. En cambio dentro del *procesamiento punto a punto* todo el sistema participa, es decir; cualquier elemento de la red puede realizar peticiones, así como prestar servicios simultáneamente con cualquier otro elemento de la red. Esta arquitectura es lo último en la distribución de procesamiento de aplicaciones. Idealmente, cada uno de los ambientes proveídos punto a punto, para el procesamiento cooperativo transparente entre posibles aplicaciones, residen entre una gama muy amplia de plataformas de Software y de Hardware, como se muestra en la figura 7.5. Una característica principal de los ambientes de procesamiento punto a punto es el soportar base de datos distribuidas en red. Cuando el Usuario del Sistema, Administrador de La Base de Datos (DBMS) realiza una consulta; realiza un movimiento marcadamente entre la base de datos heterogénea múltiple. El desarrollo actual de servidores de cooperación *cliente/servidor* y ambientes *punto a punto* no realizan una tarea trivial, para una satisfacción total de los requerimientos del procesamiento cooperativo entre servidores y clientes, además por extensión, entre nodos en *procesamiento punto a punto*, debemos saber con toda la certeza:

- Como se encuentran los clientes y los servidores a través de la red
- Desde cual cliente y servidor residen como sistemas separados (en espacios de dirección separada)
- Cuales de ellos comparten información, cuantos servidores y clientes pueden correr en diferentes plataformas heterogéneas por debajo de varios sistemas operativos abriendo sincronizando sus procesamientos a través de los diferentes protocolos de red

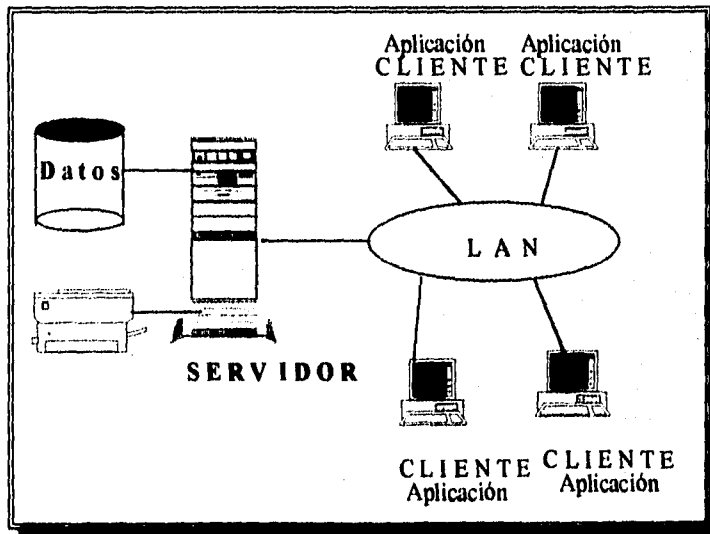


Figura 7.5 Ambiente de Procesamiento Punto a Punto.

Otra característica que no debemos descartar es la presentación de los datos, también podemos obtenerla de manera remota o local.

## EXPLICACIÓN BREVE DE LOS PRINCIPALES PROGRAMAS DE SICADE (Servidores y Clientes)

Debido a las condiciones de SICADE, podemos observar el comportamiento de los procesos; por lo que el servidor principal será la base de datos ocupada, ya sea desde una plataforma de red hasta una monousuaria.

Los clientes serán las estaciones de trabajo, en el caso de una plataforma de red, que contengan los programas y procesos para la ejecución de peticiones. Dentro del ambiente monousuario, esto es mas clara, ya que se requiere primeramente habilitar la base datos, para después ejecutar los programas y proceso de SICADE.

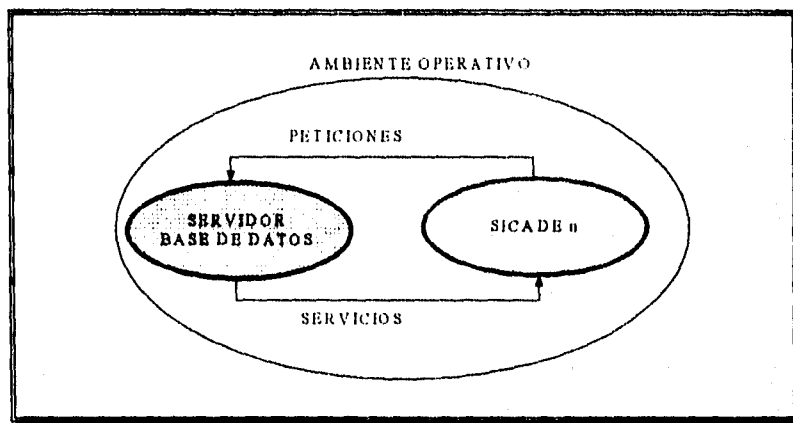


Figura 7.5 Ambiente Operativo Cliente/servidor de SICADE

## DATAGRAMAS DE PROCESOS DE INFORMACIÓN BASADOS EN LOS SERVIDORES

Como se puede apreciar en la figura 7.5, los procesos principales que generan al servidor, son los de la base de datos, así también, los procesos de SICADE serán los clientes, hablamos en plural debido a que si nos encontramos en un ambiente de red SICADE podrá ser ejecutado, simultáneamente varias veces (n), observando un ambiente cliente/servidor mas complejo.

### ESTRUCTURA DE DIRECTORIOS

La base de datos normalmente usa la estructura ilustrada en la figura 7.6. y los programas con los procesos de SICADE cuentan con la estructura de la figura 7.7:

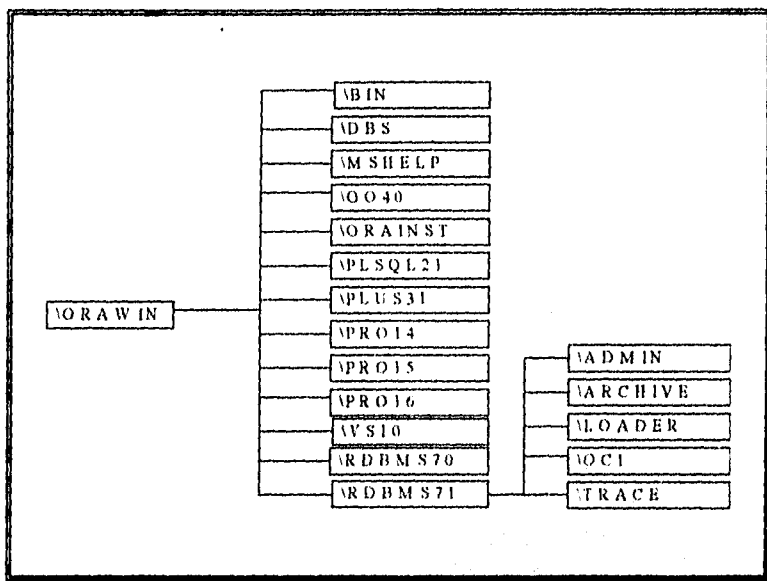


Figura 7.6 Estructuras de Directorios de la Base de Datos

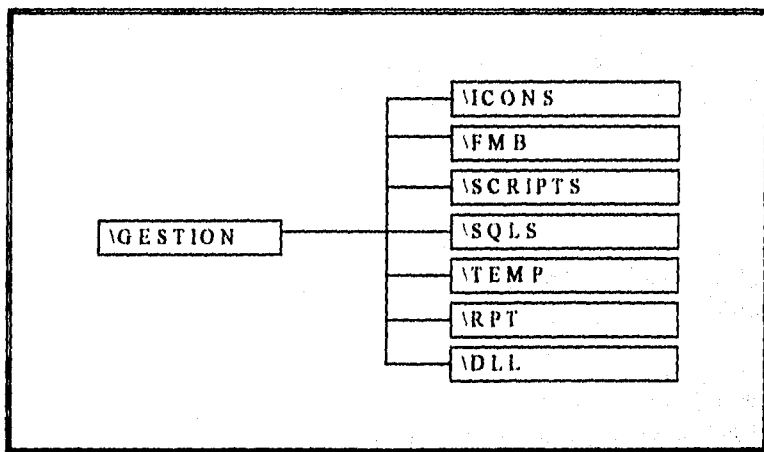


Figura 7.7 Estructuras de Directorios para SICADE

A continuación describimos el contenido de estas estructuras de directorios en las tablas 7.1 y 7.2.



<b>\ORAWIN</b>	
<b>Subdirectorio</b>	<b>Contenido</b>
\BIN	Todas las librerías de liga dinámica (DLL) y ejecutables
\DBS	Archivos de mensajes, archivos de base de datos y archivos de utilidades
\MSHELP	Archivos de ayuda
\OO4O	Archivos de Objetos ORACLE para OLE
\ORAINST	Archivos de instalación ORACLE
\PLSQL21	Archivos de mensajes para PL\SQL
\PLUS31	Archivos de SQL*PLUS
\PRO14	Archivos de mensajes, ejemplos y librerías de PRO*C
\PRO15	Archivos de mensajes, ejemplos y librerías de PRO*C
\PRO16	Archivos de mensajes, ejemplos y librerías de PRO*C
\VS10	Archivos de Herramientas para el Administrador de la Base de Datos
\RDBMS71	Mensajes de Personal Oracle 7, archivos de ayuda.
\RDBMS71\ADMIN	Todos los scripts de Personal Oracle 7, estos se requieren para la inicialización de la base de datos
\RDBMS71\ARCHIVE	Archivos de la Base de Datos para almacenamiento
\RDBMS71\LOADER	Archivos de control y archivos que serán usados en el case de ejecutar los ejemplos del LOADER
\RDBMS71\OCI	Programas de ejemplo para llamadas de interfaz a Oracle (OCI) y archivos de encabezado
\RDBMS71\TRACE	Archivos de información de procesos realizados pasos a pasos y archivos de alertas generadas por Personal Oracle 7

Tabla 7.1 Organización de los Archivos de la Base de Datos

<b>\SICADE</b>	
<b>Subdirectorio</b>	<b>Contenido</b>
\CONOS	Archivos de iconos para la manipulación de los programas de SICADE
\FMB	Archivos de formas de SICADE
\SCRIPTS	Archivos de scripts de SICADE
\SQLS	Archivos sqls de SICADE
\TEMP	Archivos temporales para la ejecución de los procesos de SICADE
\RPT	Archivos de reportes de SICADE
\DLL	Archivos de lenguaje de liga de datos , para ejecución de user exists (DLL)

Tabla 7.2 Organización de los Archivos de SICADE

## TABLAS UTILIZADAS EN LA BASE DE DATOS

<b>Gesexp(ID)</b>		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
ID	NUMBER(10)	Numero de Identificador
CVE EXP	VARCHAR2(15)	Clave del Expediente

Tabla 7.3 Gesexp(ID)

Maneja los expedientes de acuerdo a un identificador que es asignado cuando ingresa un nuevo documento.

<b>GesGral(ID)</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
TIPDOC	VARCHAR2(1)	Tipo de documento
NUMDOC	NUMBER(6)	Numero de documento
CLASIF	VARCHAR2(2)	Clasificación
REF	VARCHAR2(15)	Referencia del documento
FECDOC	DATE	Fecha del documento
FECREC	DATE	Fecha de recepción
FECDESP	DATE	Fecha de despacho
FECDESC	DATE	Fecha de descargo
FECEVEN	DATE	Fecha del evento
HOREVEN	VARCHAR2(5)	Hora del evento
FECARCH	DATE	Fecha de archivo
NOMDEP	VARCHAR2(50)	Clave de la Dependencia
ASUNTO	LONG	Asunto
DIGITALIZADO	VARCHAR2(1)	Digitalizado
ANT	NUMBER(10)	Antecedente
SUB	NUMBER(10)	Subsecuente

Tabla 7.4 GesGral(ID)

Almacena los datos necesarios para generar el seguimiento de los oficios o documentos a través del tiempo, además permite obtener el estatus de cualquier documento almacenado.

<b>GesRefAlf</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
REFALF	VARCHAR2(45)	Referencia alfabética

Tabla 7.5 GesRefAlf

Genera una correlación entre un documento y su referencia alfabética, no necesariamente debe ser el tema del documento, para obtener un rápido acceso y búsqueda entre títulos y temas de una manera mas efectiva y eficiente.

<b>GesRem</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
REMITENTE	VARCHAR2(45)	Remitente

Tabla 7.6 GesRem

Establece una referencia entre los remitentes que envían sus oficios o documentos, propiamente con ellos mismos.

<b>GesRes</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
RESOLUCIÓN	LONG	Resolución final del documento

Tabla 7.7 Gestes

Solución o determinación final hacia cualquier documento, generando un cambio de estatus del documento de vigente a histórico.

<b>GesTem</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
CVE_TEM	VARCHAR2(4)	Clave del Nombre del Tema

Tabla 7.8 GesTemp

Relación directa del documento con su identificador de tema, para una fácil búsqueda y orientación de la información.

<b>GesTurno</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
ID	NUMBER(10)	Numero de identificador
TURNO	VARCHAR2(4)	Turno
INSTRUC	VARCHAR2(1)	Instrucción
FDESPROV	DATE	Fecha de despacho
DESCARGO	DATE	Fecha de descargo

Tabla 7.9 GesTurno

Mantiene una relación directa entre un documento y los diferentes turnados, donde almacena las actividades o instrucciones tomadas para la solución del documento, asentando las fechas de recepción del documento como las de resolución.

<b>Clasif</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_CLA	VARCHAR2(2)	Clave de clasificación
NOMCLA	VARCHAR2(15)	Nombre de clasificación

Tabla 7.10 Clasif

Catálogo requerido para la clasificación de los oficios o documentos, dependiendo de la importancia de estos mismos.

<b>Tema</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_TEM	VARCHAR2(4)	Clave del Nombre del Tema
NOMTEM	VARCHAR2(45)	Nombre del tema

Tabla 7.11 Tema

Catálogo requerido para la agilización de la clasificación de temas para un documento.

<b>Dependencia</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
NOMDEP	VARCHAR2(50)	Nombre de la dependencia

Tabla 7.12 Dependencia

Catálogo requerido para la agilización de la clasificación de dependencias para un documento.

<b>Expediente</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_EXP	VARCHAR2(15)	Clave de expediente
NOMEXP	VARCHAR2(50)	Nombre de expediente

Tabla 7.13 Expediente

Catálogo requerido para la agilización de la clasificación de expedientes por dependencia.

<b>Documento</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
TIPDOC	VARCHAR2(1)	Tipo de documento
NUMDOC	NUMBER(6)	Numero de documento
DESCRIP	VARCHAR2(20)	Descripción

Tabla 7.14 Documento

Relación directa entre el numero de oficios o documentos del mismo tipo y su clasificación de estos.

<b>Turno</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_TUR	VARCHAR2(4)	Clave del turnado
NOMTUR	VARCHAR2(45)	Nombre del turnado
CARGO	VARCHAR2(45)	Cargo
AREA	VARCHAR2(10)	Área
CALLEDOM	VARCHAR2(100)	Calle del domicilio
COLDOM	VARCHAR2(50)	Colonia del domicilio
CPDOM	VARCHAR2(5)	Código postal del domicilio
TELDOM	VARCHAR2(50)	Teléfono del domicilio
FAXDOM	VARCHAR2(50)	Fax del domicilio
CALLEOFI	VARCHAR2(100)	Calle de la oficina
COLOFI	VARCHAR2(50)	Colonia de la oficina
CPOFI	VARCHAR2(5)	Código postal de la oficina
TELOFI	VARCHAR2(100)	Teléfono de la oficina
FAXOFI	VARCHAR2(100)	Fax de la oficina
CIUDADOM	VARCHAR2(25)	Ciudad del domicilio
CIUDAOFI	VARCHAR2(25)	Ciudad de la oficina
ESPOSA	VARCHAR2(45)	Esposa

Tabla 7.15 Turno

Almacenamiento de información detallada y clasificada de las personas, a las cuales se les reasigna el tratado del documento en proceso.

<b>Instruc</b>		
<i>Nombre</i>	<i>Tipo</i>	<i>Descripción</i>
CVE_INST	VARCHAR2(1)	Clave de instrucción
NOMINST	VARCHAR2(45)	Descripción de la instrucción

Tabla 7.16 Instruc

Catálogo de las diferentes instrucciones que debe realizar un turnado.

## MODELO ENTIDAD RELACIÓN DE LA BASE DE DATOS

Este modelo se presenta para que las personas interesadas en el uso o modificación del sistema tengan de inmediato una referencia o panorama del diseño del SICADE, por lo que es pertinente señalar que este diseño se realizó completamente enfocado a satisfacer los requerimientos del usuario en su totalidad.

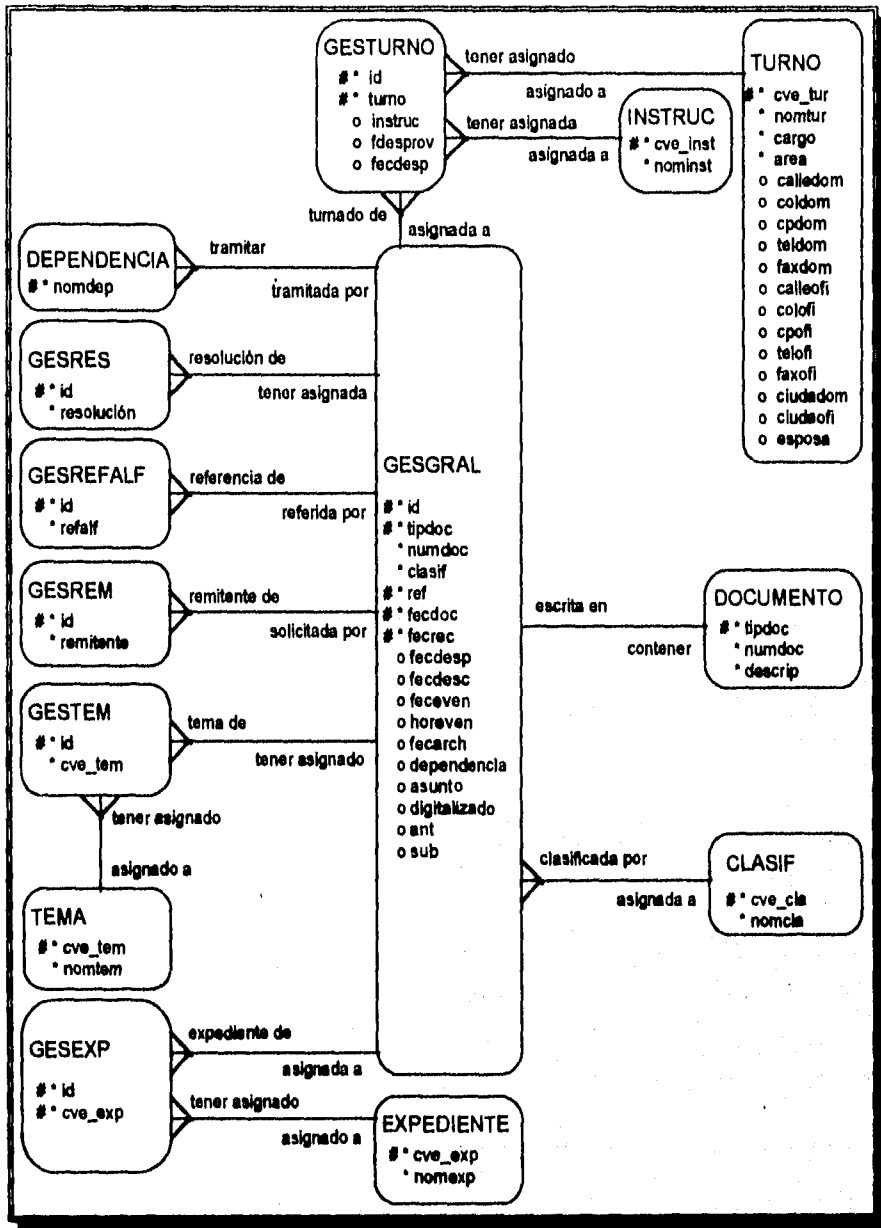


Figura 7.8 Modelo Entidad Relación

## DESCRIPCIÓN, DEFINICIÓN Y DECLARACIÓN DE LAS PRINCIPALES FUNCIONES Y PROCEDIMIENTOS DEL SISTEMA

En el esquema de la figura 7.9 se muestra la organización de las funciones y procedimientos, mostrando las rutas de seguimiento y las prioridades de acceso.

Con las siguientes tablas se describen detalladamente todos los procesos específicamente, resaltando los parámetros y campos mas importantes para las transacciones realizadas dentro de SICADE.

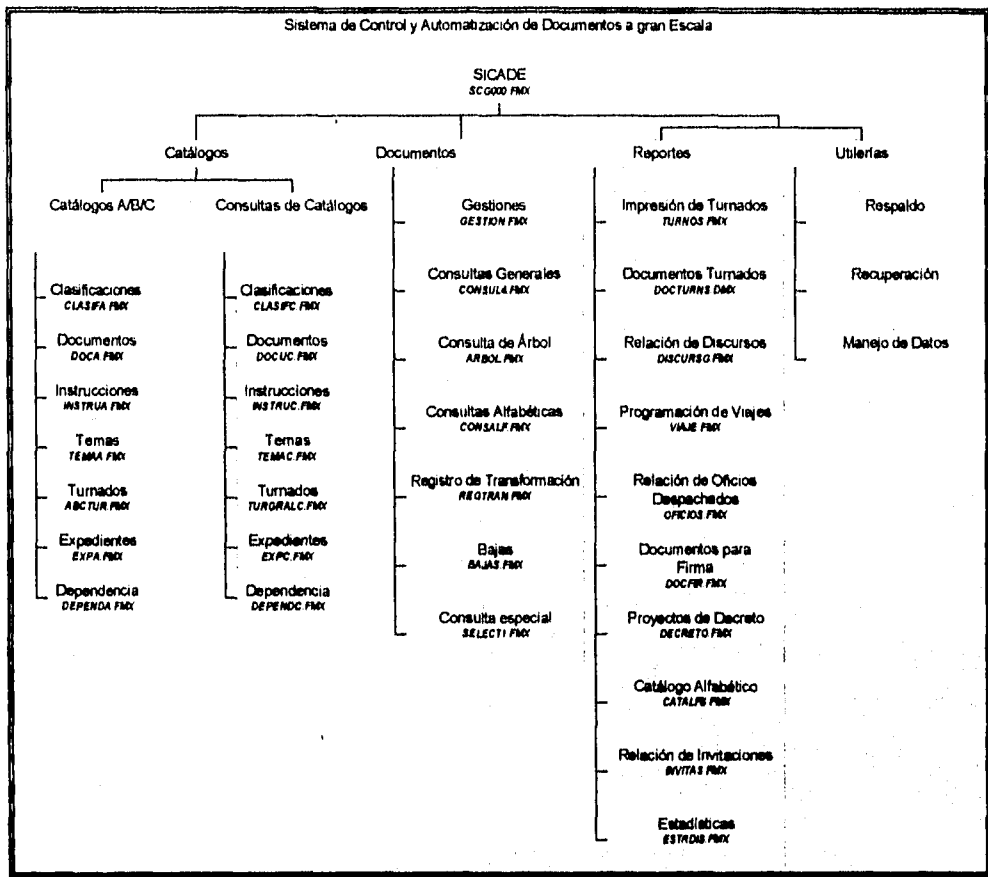


Figura 7.9 Distribución de Menús de SICADE

<b>SICADE</b>	
<b>Nombre</b>	<b>Descripción</b>
<b>Catálogos</b>	Información descriptiva de las características para los oficios o documentos
<b>Documentos</b>	Recopilación de información
<b>Reportes</b>	Obtención de información recopilada con características requeridas por el usuario.
<b>Utilerías</b>	Herramientas que ayudan a salvaguardar la información

Tabla 7.17 SICADE (Menú Principal)

<b>CATÁLOGOS</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Altas, Bajas y Cambios</i>	Menú de opciones para el tratamiento de oficios o documentos
<i>Consultas</i>	Menú de opciones para la consulta de documentos

Tabla 7.18 Catálogos

<b>Altas, Bajas y Cambios</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Clasificación</i>	Tipo de importancia que puede tomar un documento
<i>Documentos</i>	Tipo de Documentos
<i>Instrucciones</i>	Acciones que realiza el Turnado
<i>Turnados</i>	Personas disponibles, las cuales darán respuesta al oficio o documento
<i>Temas</i>	Título asignado a los asuntos de un oficio o documento
<i>Expedientes</i>	Es la clave que se le asigna a cada institución de acuerdo a su organización jerárquica
<i>Dependencias</i>	Nombres de las instituciones o empresas solicitantes

Tabla 7.19 Altas, Bajas y Cambios

<b>CONSULTAS</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Clasificación</i>	Tipo de Importancia que puede tomar un documento
<i>Documentos</i>	Tipo de Documentos
<i>Instrucciones</i>	Acciones que realiza el Turnado
<i>Turnados</i>	Personas disponibles, las cuales darán respuesta al oficio o documento
<i>Temas</i>	Título asignado a los asuntos de un oficio o documento
<i>Expedientes</i>	Es la clave que se asigna a cada institución de acuerdo a su organización jerárquica
<i>Dependencias</i>	Nombre de las Instituciones o Empresa solicitantes

Tabla 7.20 Consultas

<b>DOCUMENTOS</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Gestiones</i>	Registra y realiza movimientos de información de un documento.
<i>Consultas Generales</i>	Búsqueda de información
<i>Consultas de Árbol</i>	Búsqueda de información, en la cual se presentan datos antecedentes y consecuentes a la información citada
<i>Consultas Alfabéticas</i>	Búsqueda de información por palabra de remitencia, referencia alfabética y temas
<i>Registro de Transformación</i>	Genera oficios o documentos subsecuentes de otros
<i>Bajas</i>	Borra los registros de oficios o documentos
<i>Consultas Especiales</i>	Realiza las consultas por tipo de documento, rango de identificador, clasificación, instrucción y turnado

Tabla 7.21 Documentos



<b>REPORTE</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Impresión de Turnados</i>	Requiere de un rango de numero de documento para generar los oficios o documentos de turno.
<i>Documentos Turnados</i>	Reporte por numero de documento
<i>Relación de Discursos</i>	Requiere de un rango de Identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de oficios o documentos, distintos tipos de turnados, clasificaciones e instrucciones para generar una relación de discursos.
<i>Programación de Viajes</i>	Requiere un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de documentos, distintos tipos de clasificaciones para generar la programación de viajes
<i>Relación de Oficios Despachados</i>	Requiere un rango de documento, distintos tipos de turnados, distintas clasificaciones, y distintas instrucciones para generar una relación de oficios o documentos despachados.
<i>Documentos para Firma</i>	Requiere un rango de oficios o documentos, rango de fecha de recepción, distintos turnados, distintas clasificaciones, distintas instrucciones para generar una documentación de firma
<i>Proyectos de Decreto</i>	Requiere un rango de numero de oficios o documentos, rango de fecha de recepción, distintos turnados, clasificaciones e instrucciones para generar un proyecto de decreto
<i>Catalogo Alfabético</i>	Requiere un rango de numero de identificador para generar el catalogo o lista alfabética
<i>Relación de Invitaciones</i>	Requiere un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de turnados, clasificaciones e instrucciones, para generar una relación de invitaciones.
<i>Estadísticas</i>	Requiere un rango de identificador, tipo de documento y clasificación, para generar estadísticas

Tabla 7.22 Reporte

<b>UTILERÍAS</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>Respaldo</i>	Genera el respaldo de la base de datos, en forma total para generar un archivo plano en un directorio específico. Debe mencionarse que este proceso deberá ejecutarse periódicamente para tener un respaldo, reciente y no sufrir pérdidas considerables si llegase a sufrir algún daño la base de datos
<i>Recuperación</i>	Este proceso permite recargar la base de datos contra información de periodos anteriores, debido a una incorrecta acentuación de información o por pérdidas de integridad de la base de datos.
<i>Manejo de Datos</i>	Permite la integración e interrelación de toda la base de datos para poder generar un archivo con características determinadas por el usuario, para poder generar relaciones de reportes; explotadas por otras herramientas dedicadas a este objetivo, tales como hojas de calculo o sencillamente puede funcionar como retroalimentador de otro sistema si así se requiere periódicamente.

Tabla 7.23 Utilerías

<b>FORMAS</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>TURN.FMX</i>	Altas, Bajas y Cambios de Turnados	Manipulación de Personas disponibles, las cuales darán respuesta o seguimiento a un oficio o documento.
<i>ARBOL.FMX</i>	Consultas de Árbol	Búsqueda de información, en la cual se presentan datos tanto antecedentes como consecuentes
<i>BAJAS.FMX</i>	Bajas de Gestiones	Borra los registros de cualquier documento
<i>CATALFS.FMX</i>	Catálogo Alfabético	Requiere de un rango de numero de identificador para generar el catalogo o lista alfabética.
<i>CLASIFA.FMX</i>	Altas, Bajas y Cambios de Clasificaciones	Manipulación de las clasificaciones donde las clasificaciones son atributos de importancia que toma un oficio o documento
<i>CLASIFC.FMX</i>	Consultas de Clasificaciones	Consultas de las clasificaciones de un oficio o documento
<i>CONSALF.FMX</i>	Consultas Alfabéticas	Busque de información en oficios o documentos por palabras de remitencia , referencia alfabética y temas
<i>CONSUL4.FMX</i>	Consultas Generales	Búsqueda de información de interés general en los oficios o documentos
<i>DECRETO.FMX</i>	Proyectos de decreto	Requiere de un rango de numero de oficios o documentos, un rango de fecha de recepción, distintos turnados, clasificaciones e instrucciones para generar un proyecto de decreto
<i>DEPENDA.FMX</i>	Altas, Bajas y Cambios de Dependencia	Manipulación de los nombres de las instituciones o empresas solicitantes
<i>DEPENDC.FMX</i>	Consultas de Dependencias	Consultas de los nombres de las instituciones o empresas solicitantes.
<i>DISCURSO.FMX</i>	Relación de Discursos	Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de oficios o documentos, distintos tipos de turnados, clasificaciones e instrucciones para generar una relación de discursos
<i>DOCA.FMX</i>	Altas, Bajas y Cambios de Documentos	Manipulación de oficios o documentos

Tabla 7.24 Formas (continúa)

<b>FORMAS</b> (continuación)		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<b>DOCFIR.FMX</b>	Documentos para Firma	Requiere de un rango de oficios o documentos, rango de fecha de recepción, distintos turnados, distintas clasificaciones, distintas instrucciones para generar una documentación de firma
<b>DOCTURNS.FMX</b>	Documentos Turnados	Requiere de un numero de oficio o documento para generar un reporte referente al mismo documento
<b>DOCUC.FMX</b>	Consultas de Documentos	Realiza una descripción sobre los diferentes tipos de oficios o documentos
<b>ESTADIS.FMX.</b>	Estadísticas	Requiere de un rango de identificador, tipo de oficio o documento y clasificación para generar las estadísticas de los diferentes estados de tratamiento de un oficio o documento.
<b>EXPA.FMX</b>	Altas, Bajas y Cambios de Expedientes	Manipulación de las claves que se asignan a cada institución de acuerdo a su organización jerárquica
<b>EXPC.FMX</b>	Consultas de Expedientes	Consultas de las claves que se asignan a cada institución de acuerdo a su organización jerárquica
<b>GESTION.FMX</b>	Gestiones	Registra y realiza movimientos de información de un oficio o documento
<b>INSTRUA.FMX</b>	Altas, Bajas y Cambios de Instrucciones	Manipulación de las acciones que realizar un turnado
<b>INSTRUC.FMX</b>	Consultas de Instrucciones	Realiza consultas de las acciones que puede realizar un turnado
<b>INVITAS.FMX</b>	Relación de Invitaciones	Requiere de un rango de identificador , rango de fecha de recepción, rango de fecha de evento, distintos tipos de turnados, clasificaciones e instrucciones, para poder generar una relación de invitaciones
<b>OFICIOS.FMX</b>	Relación de Oficios Despachados	Requiere de un rango de oficios o documento, distintos tipos de turnados, distintas clasificaciones y distintas instrucciones para generar una relación de oficios despachados.
<b>REGTRAN.FMX</b>	Registro de Transformación	Realiza la generación de oficios o documentos subsecuentes de un oficio o documento primario
<b>SCG000.FMX</b>	Menú principal	Realiza el control y manipulación de las diferentes acciones que realiza SICADE

Tabla 7.24 Formas (continúa)

<b>FORMAS (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
SELECT1.FMX	Consultas Especiales	Realiza las consultas por tipos de oficio o documento, rango de identificador, clasificación, instrucción y turnado para la búsqueda de información contenida en un oficio o documento
TEMAA.FMX	Altas, Bajas y Cambios de Tema	Manipulación de los títulos asignados a los asuntos de un oficio o documento
TEMAC.FMX	Consultas de Temas	Consultas de los títulos asignados a los asuntos de un oficio o documento
TURGRALC.FMX	Consultas de Turnados	Consulta de las personas disponibles, las cuales darán respuesta al oficio o documento
TURNOS.FMX	Impresión de Turnados	Requiere de un rango del número de documentos para generar los documentos del turno
VIAJE.FMX	Programas de Viajes	Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de documentos, distintos tipos de turnados, distintos tipos de clasificaciones para generar la programación de viajes

Tabla 7.24 Formas

<b>REPORTES</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
CLASIFI.RDF	Clasificaciones	Generación de Reporte de Actividades
DEPENDI.RDF	Dependencias	Generación de Reporte de Dependencias
GRALTUR.RDF	Catalogo general de Turnados todos los campos	Generación de Reporte General de Turnados
CATTUR.RDF	Catalogo Parcial de Turnados (Clave, Norm, Cargo)	Generación del catalogo de Turnados
DOCUMENT.RDF	Documentos	Generación de Reporte de Documentos
EXPEDI.RDF	Expedientes	Generación de Reporte de Expedientes
ETIQUET.RDF	Etiquetas de turnados 4 etiquetas	Reporte para generar de Cuatro Etiquetas por cada Turnado seleccionado
INSTRUCI.RDF	Instrucciones	Generación del Reporte Del Catalogo de Instrucciones
CATALF.RDF	Catalogo Alfabético Titulo dependencia	Generación del Reporte del Catalogo Alfabético

Tabla 7.25 Reportes (Continúa)

<b>REPORTES (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>TEMAI.RDF</i>	Tema	Generación del Reporte del Catalogo de Temas
<i>CONSALF1.RDF</i>	Consultas Alfabéticas	Generación de Reportes asignados a las Consultas Alfabéticas
<i>GNORMALX.RDF</i>	Consultas Generales reporte Normal Láser.	Reporte Normal de Consultas Generales con salida a Impresora Láser.
<i>NORMAL3.RDF</i>	Consultas Alfabéticas reporte Normal Matriz	Reporte Normal de Consultas Alfabéticas con salida a Impresora de Matriz de Punto.
<i>CONSGRAL.RDF</i>	Consultas Generales por pantalla	Reporte de Consultas Generales con salida en monitor.
<i>NORMAL3X.RDF</i>	Consultas Alfabéticas reporte Normal Láser	Reporte de Consultas Alfabéticas con salida a Impresora Láser.
<i>COMPLET1.RDF</i>	Consultas Alfabéticas reporte Completo Matriz	Reporte Completo de Consultas Alfabéticas con salida a impresora de Matriz de punto
<i>COMPLETX.RDF</i>	Consultas Alfabéticas reporte Completo Láser	Reporte Completo de Consultas Alfabéticas con salida a impresora Láser.
<i>SEGUI1.RDF</i>	Consultas Alfabéticas reporte Seguimiento Matriz	Reporte de Seguimientos de Consultas Alfabéticas con salida a Impresora de matriz de punto
<i>SEGUI1X.RDF</i>	Consultas Alfabéticas reporte Seguimiento Láser	Reporte de Seguimientos de Consultas Alfabéticas con salida a impresora Láser
<i>DISCURSO.RDF</i>	Discurso	Generación de Reporte de Discursos
<i>RESUMI1.RDF</i>	Consultas Alfabéticas reporte Resumido Matriz	Generación de Reporte Resumido para Consultas Alfabéticas con salida de impresora de Matriz de Punto.
<i>RESUMI1X.RDF</i>	Consultas Alfabéticas reporte Resumido Láser	Generación de Reporte Resumido para Consultas Alfabéticas con salida de Impresora Láser
<i>VIAJE.RDF</i>	Viaje	Generación de Reporte de Viajes
<i>OFICIO.RDF</i>	Oficio	Generación de Reporte para los Oficios Tomados
<i>RELDIS.RDF</i>	Relación de Discursos	Generación de Reportes de la Relación de Discursos
<i>DOCFIR.RDF</i>	Documentos para firma	Generación de Reportes de los Documentos para Firma
<i>DECRETO.RDF</i>	Decreto	Generación de Reporte para los Decretos
<i>GSEGUI1X.RDF</i>	Consultas Generales reporte de Seguimiento Láser	Generación de Reporte de Seguimiento de Consultas Generales con salida a impresora láser

Tabla 7.25 Reportes (Continúa)

<b>REPORTES (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
DECRET.RDF	Decreto Especial	Generación de Reportes para la relación de decretos
GCOMPLEX.RDF	Consultas Generales Reporte Complemento Láser	Generación de Reportes Completos de Consultas Generales con salida de impresora láser
GRESUMIX.RDF	Consultas Generales Reporte resumido, Láser	Generación de Reportes Resumido de Consultas Generales con salida de impresora láser
INVITA.RDF	Invitaciones	Generación de Reporte de Invitaciones
GNORMAL.RDF	Consultas Generales Reporte Normal Matriz	Generación de Reportes Normales de Consultas Generales con salida a impresora de matriz de punto.
GCOMPLET.RDF	Consultas Generales Reporte Complemento Matriz	Generación de Reportes Completos de Consultas Generales con salida de impresora láser
CONSESP.RDF	Consultas Especiales	Generación de Reporte de Consultas Especiales
GRESUMI1.RDF	Consultas Generales reporte resumido Matriz	Generación de Reportes Resumidos de Consultas Generales con salida a impresora de matriz de punto
GSEGUI1.RDF	Consultas Generales seguimiento Matriz	Generación de Reportes de Consultas Generales con salida a impresora de matriz de punto
DOCTUR.RDF	Documentos Turnados	Generación de Reportes de los documentos asignados a los turnados
DOCTURES.RDF	Documentos Turnados con resolución	Generación de Reportes de los documentos asignados sin resolución
IMPTUR.RDF	Impresión de Turnados	Generación de Reportes de los documentos de turnados para impresión
DOCTURIN.RDF	Documentos Turnados Invitaciones	Generación de Reportes de los documentos turnados para invitaciones
DOCTURAU.RDF	Documentos Turnados Audiencias	Generación de Reportes de los Documentos Turnados para Audiencias
CATALF1.RDF	Catalogo Alfabético Titulo Remitente	Generación de los Reportes Asignados a la Relación del Catalogo Alfabético
ESTADIS.RDF	Estadísticas	Generación de los Reportes de Estadísticas
DOCTUROB.RDF	Documentos Turnados Obsequios	Generación de Reportes de los Documentos turnados de obsequios
ETIQUET6.RDF	Etiquetas de Turnados (6 etiquetas)	Generación de Reporte para la obtención de seis etiquetas para el envío de correspondencia

Tabla 7.25 Reportes (Continúa)

<b>REPORTES (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>CATALF2.RDF</i>	Catálogo alfabético Titulo tema	Generación de Reportes para el catálogo alfabético por el título del tema
<i>CATALF3.RDF</i>	Catálogo Alfabético Titulo referencia Alfabética	Generación de Reportes Alfabéticos para el catálogo alfabético por el título del tema
<i>CATALF4.RDF</i>	Catálogo Alfabético Titulo Turnado	Generación de Reportes para los catálogos Alfabéticos por el título del turnado

Tabla 7.25 Reportes

<b>USER EXIT</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>catalf</i>	Genera un SQL externo (catalf.sql), el cual se ejecutara por medio del Reporte de Viajes.	Requiere de un rango de identificador, rango de fecha de recepción, rango de fecha de evento, distintos tipos de documentos, distintos tipos de turnados, distintos tipos de clasificaciones para generar la programación de viajes
<i>conesp</i>	Genera un SQL externo (conesp.sql), el cual se ejecutara por medio de la forma de Consultas Especiales.	Realiza las consultas por tipos de oficio o documento, rango de identificador, clasificación, instrucción y turnado para la búsqueda de información contenida en un oficio o documento
<i>decret</i>	Genera un SQL externo (decret.sql), el cual se ejecutara por medio del Reporte de Decreto Especial.	Realiza la selección de los registros en la búsqueda de los decretos, por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones. Desplegando el identificador, tipo de documento, numero de documento, clasificación, referencia y asunto

Tabla 7.26 User Exit (Continúa)

<b>USER EXIT (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>decreto</i>	Genera un SQL externo (decreto.sql), el cual se ejecutara por medio del Reporte de Decreto.	Realiza la selección de los registros en la búsqueda de los decretos, por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones. Desplegando el identificador, tipo de documento, numero de documento, fecha de documento, fecha de recepción, dependencia, asunto, fecha de despacho, fecha de descargo, referencia, antecedente y subsecuente
<i>docfir</i>	Genera un SQL externo (docfir.sql), el cual se ejecutara por medio del Reporte de Documentos para Firma.	Genera la selección de los documentos para firma, por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.
<i>doctur</i>	Genera un SQL externo (doctur.sql), el cual se ejecutara por medio de la forma de Impresión de Documentos Turnados.	Genera la selección de los documentos que han sido turnados por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.
<i>estadis</i>	Genera SQL's externos (estadis1.sql, estadis2.sql, estadis3.sql), los cuales se ejecutan por medio del Reporte de Viajes.	Genera las estadísticas de documentos (turnados, no turnados, etc.), por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.
<i>gentem</i>	Genera un SQL externo (gesgral.sql), el cual se ejecutara por medio del Reporte de Consultas Generales.	Realiza las consultas para obtener los documentos del Reporte de Consulta Generales por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.

Tabla 7.26 User Exit (Continúa)



<b>USER EXIT (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>invita</i>	Genera un SQL externo (invita.sql), el cual se ejecutara por medio del Reporte de 'Invita.	Genera la selección de los documentos que han sido ingresados como invitaciones por medio de un rango de numero de documentos, un rango de fecha de recepción, distintas clasificaciones y distintas instrucciones.
<i>invtur</i>	Genera un SQL externo (invtur.sql), el cual se ejecutara por medio de la forma de Impresión de Doc. Turnados (Invitaciones).	Genera la selección de los documentos que han sido asignados a los turnados con motivo de invitación para saber el estatus de ellos por medio de de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.
<i>obstur</i>	Genera un SQL externo (obstur.sql), el cual se ejecutara por medio de la forma de Impresión de Doc. Turnados (Obsequios).	Genera la selección de los documentos que han sido asignados a los turnados con motivo de obsequios para saber el estatus de ellos por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.
<i>oficio</i>	Genera un SQL externo (oficio.sql), el cual se ejecutara por medio del Reporte de oficio.pc	Genera la relación de documentos que han sido recibidos con el carácter de Oficio por medio de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones.

Tabla 7.26 User Exit (Continúa)

<b>USER EXIT (continuación)</b>		
<b>NOMBRE</b>	<b>FUNCIÓN</b>	<b>DESCRIPCIÓN</b>
<i>reldis</i>	Genera un SQL externo (reldis.sql), el cual se ejecutara por medio de la forma de Consultas Especiales.	Genera la relación de los documentos que serán una consulta especial en donde primordialmente se requiere de un rango de numero de documentos, un rango de fecha de recepción, distintos turnados, distintas clasificaciones y distintas instrucciones, además de otros parámetros para la ejecución de esta consulta.
<i>reporte</i>	Genera un archivo GenRep.bat, para después ejecutarlo desde la forma.	Generación del reporte general, el cual será disparado por medio de una forma
<i>turno</i>	Genera un SQL externo (tumo.sql), el cual se ejecutara por medio de la forma de Impresión de Turnos.	Obtiene el reporte de turnados para después ser impresos
<i>viaje</i>	Genera un SQL externo (viaje.sql), el cual se ejecutara por medio del Reporte de Viajes.	Genera la relación de viajes a partir de los datos de los documentos clasificados por prioridad del documento y fechas de recepción y descargo

Tabla 7.26 User Exit

## **PROCESOS DE RESPALDO / RESTAURACIÓN DEL SISTEMA Y SU INFORMACIÓN**

Estas herramientas han sido elaboradas para poder obtener respaldos temporales o para depuración de datos, los cuales podrán ser almacenados en archivos planos y cuando se requiera estos datos serán reingresados o intercambiados al sistema.

El poder tener respaldos periódicos genera un alto grado de seguridad contra posibles problemas de perdidas de información o simplemente cuando en la base de datos existe información de carácter histórico y únicamente genera que los procesos sean mas lenta, por lo que el poder separar esta información, mantendrá al sistema en un máximo de rendimiento o desempeño.

## **RESPALDO DE INFORMACIÓN**

Esta herramienta esta disponible para diferentes sistemas operativos en distintas plataformas de hardware. Únicamente nos abocaremos a describirla en un ambiente monousuario (WINDOWS 3.1).

**Procedimiento.**

1. Dentro del Menú de Utilerías seleccionar "Respaldo de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Exporter".
5. Deberá ingresar o cambiar los siguientes parámetros o valores de las opciones señaladas:

<b>Campo</b>	<b>Valor</b>
Export File	c:\gestion\bck\sicade.dmp
To Be Exported	
Tables	Sin seleccionar
Owners	Sin seleccionar
Full Database	Seleccionada
Export These Objects	
Grants	Seleccionada
Constraints	Seleccionada
Rows	Seleccionada
Indexes	Seleccionada

Tabla 7.27 Valores para Respaldo de Información (continúa)

6. Seleccionar el botón de "Advanced"
7. Aparecerá un dialogo con el siguiente nombre "Advanced Export Options"
8. Deberá ingresar o cambiar los siguientes parámetros o valores de las opciones señaladas:

<b>Campo</b>	<b>Valor</b>
Record Length	1024 Bytes
Buffer Size	4096
Record Export in System Data	Sin seleccionar
Compress when imported	Seleccionada
Keep Components Consistent	Sin seleccionar
Increment Type	NONE
Statistics Type	NONE
Generate Log File	Seleccionada
Log File	c:\gestion\bck\sicade.log

Tabla 7.27 Valores para Respaldo de Información

9. Seleccionar el botón de "Export", si este botón no a cambiado a un estado deshabilitación, cancelar los diálogos "Advanced Export Options" y "Database Exporter", en este respectivo orden y comenzar desde el paso uno. Si esto no puede realizarse acudir a los manuales *Personal Oracle 7 Instalation and User's Guide* y *Oracle 7 Server \_Utilities User's Guide*.

**Descripción de los términos usados en los diálogos del respaldo de información**

<b>Elemento del Diálogo</b>	<b>Descripción</b>
<b>Export File</b>	Ingresar en este elemento el nombre de uso en el sistema operativo el cual contendrá los datos que desea exportar. (Se requiere un valor específico y correcto del nombre de un archivo, con respecto al sistema operativo).
<b>Browse</b>	Este elemento permite seleccionar un nombre de archivo provecto por el sistema operativo para que almacene los datos que se requieran respaldar.
<b>Tables</b>	Selección que generara el respaldo de Tablas o Entidades que se requieren.
<b>Owners</b>	Selección que generara el respaldo de los objetos que se requieren pertenecientes a un usuario en particular
<b>Full Database</b>	Selección que genera el respaldo total de la Base de Datos
<b>Specify...</b>	Si se a seleccionado Tables o Owners, al seleccionar Specify, se podrá especificar las tablas o entidades de un usuario o los usuarios a respaldar.
<b>Grants</b>	Seleccionara esta opción si usted requiere los permisos de los objetos a respaldar.
<b>Rows</b>	Esta opción permitirá respaldar los renglones de las tablas de datos, ya que si no se selecciona únicamente se respaldara las definiciones de las base de datos
<b>Constrains</b>	Esta opción permitirá respaldar las restricciones de las base de datos
<b>Indexes</b>	Esta opción permitirá respaldar los Indices de la base de datos.
<b>Advanced</b>	Habilitara opciones avanzadas del respaldo de la base de datos (Habilitará el dialogo de Advanced Export Options)
<b>Database</b>	Permite seleccionar la base de datos con la cual se requiere trabajar para realizar el respaldo

Tabla 7.28 Términos del Respaldo de Información (Continúa)

<b>Elemento del Diálogo</b>	<b>Descripción</b>
Defaults	Reincorpora las opciones por omisión definidas por el fabricante
Record Length	Permite intercambiar la longitud del registro de intercambio para el respaldo
Buffer Size	Permite intercambiar el tamaño del Buffer para realizar los procesos de Fetch asignados al respaldo
Record Export in System Data	Permite respaldar los procesos almacenados en la base de datos
Compress when imported	Permite realizar la compresión de datos mientras se realiza el respaldo
Keep Components Consistent	Permite exportar la consistencia de lectura desde una vista de la base de datos
Increment Type	Permite seleccionar el modo de respaldo incremental, con las siguientes opciones: NONE, INCREMENTAL, COMPLETE, y CUMULATIVE.
Statistics Type	Permite intercambiar el tipo de las estadísticas de optimización de la base de datos para generarse cuando los datos respaldados son restaurados dentro de la base de datos.
Generate Log File	Permite intercambiar la generación del archivo de log o salida de resultados en el respaldo.
Log File	Nombre del archivo de log o salida de resultados de los procesos de respaldo

Tabla 7.28 Términos del Respaldo de información (Continuación)

## RESTAURACIÓN DE INFORMACIÓN

Estas herramientas han sido elaboradas para poder obtener restauración de información, los cuales podrán ser tomados de archivos planos y cuando se requiera estos datos serán reingresados o intercambiados al sistema.

El poder tener respaldos periódicos genera un alto grado de seguridad contra posibles problemas de pérdidas de información o simplemente cuando en la base de datos existe información de carácter histórico y únicamente genera que los procesos sea mas lenta, por lo que el poder separa esta información, mantendrá al sistema en un máximo de rendimiento o desempeño.

### Procedimiento.

1. Dentro del Menú de Utilerias seleccionar "Recuperación de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Ingrese el usuario, clave y base de datos a restaurar.
4. Deberá ingresar o cambiar los siguientes parámetros o valores de las opciones señaladas:

<b>Campo</b>	<b>Valor</b>
Import File	\\gestion\bck\sicade.dmp
Ignore Creation Errors	Sin Seleccionar
Import Mode	
Tables	Sin seleccionar
Users	Sin seleccionar
Full Database	Seleccionada
Tables to Import	Specify Tables before Import or Show can proceed
Objects to Import	
Grants	Seleccionada
Indexes	Seleccionada
Rows	Seleccionada

Tabla 7.29 Valores para la Restauración de la Información (continúa)

5. Seleccionar el botón de "Advanced"
6. Aparecerá un dialogo con el siguiente nombre "Advanced Import Options"
7. Deberá ingresar o cambiar los siguientes parámetros o valores de las opciones señaladas:

<b>Campo</b>	<b>Valor</b>
Record Length	1024 Bytes
Buffer Size	4096
Commit after each array	Sin seleccionar
Overwrite existing data files	Sin seleccionar
Generate Log File	Seleccionada
Log File	c:\gestion\bck\sicadei.log
Write Index Creation Commands to File	Sin seleccionar
Index File	Sin seleccionar
Increment Type	
None	Seleccionada
System	Sin Seleccionar
Restore	Sin Seleccionar

Tabla 7.29 Valores para la Restauración de la Información

8. Seleccionar el botón de "Ok".
9. Seleccionar el botón de "Import", si este botón no a cambiado a un estado de deshabilitación, cancelar los diálogos "Advanced Import Options" y "Database Exporter", en este respectivo orden y comenzar desde el paso uno. Si esto no puede realizarse acudir a los manuales *Personal Oracle 7 Instalation and User's Guide* y *Oracle 7 Server Utilities User's Guide*.

## Descripción de los términos usados en los Diálogos de la Restauración de Información.

<b>Elemento del Diálogo</b>	<b>Descripción</b>
<i>Import File</i>	<i>Archivo del Sistema Operativo específicamente para restaurar la información hacia la base de datos. (Especifica el requerido del parámetro File)</i>
<i>Browse</i>	<i>Se usa esta opción para poder seleccionar del sistema operativo el archivo que se va a restaurar en la Base de Datos.</i>
<i>Ignore Creation Errors</i>	<i>Si se selecciona esta opción, el proceso de restauración no reportara el mensaje de "Object already exists" además de los errores cuando se creen los objetos en la base de datos.</i>
<i>Tables</i>	<i>Especifica al proceso el modo de restaurar algunas entidades o tablas</i>
<i>Users</i>	<i>Especifica al proceso el modo de restaurar las entidades o tablas de la Base de Datos, pertenecientes a los usuarios</i>
<i>Full Database</i>	<i>Especifica al proceso el modo total de la Base de Datos</i>
<i>Tables to Import</i>	<i>Si se ha seleccionado la opción del modo de "Tables", con esta opción podrá seleccionar las tablas que desea restaurar.</i>
<i>Users to Import</i>	<i>Si se ha seleccionado la opción del modo de "Users", con esta opción podrá seleccionar las tablas que corresponden al usuario que desea restaurar.</i>
<i>Specify</i>	<i>Si se ha seleccionado el modo de "Tables" o "Users", con esta opción se podrá seleccionar una lista de entidades o tablas o una lista de usuarios para restaurar.</i>
<i>Grants</i>	<i>Seleccionara esta opción si usted requiere que los permisos de los objetos sean restaurados.</i>
<i>Rows</i>	<i>Especifica al proceso que los renglones serán restaurados</i>
<i>Indexes</i>	<i>Especificara al proceso que los índices serán restaurados</i>
<i>Advanced</i>	<i>Permite habilitar la edición de parámetros avanzados del proceso de restaurar</i>
<i>Database</i>	<i>Permite intercambiar la base de datos con la cual se esta trabajando actualmente.</i>
<i>Defaults</i>	<i>Cuan es accionado pone todos los parámetros en su modo o estado de omisión.</i>

Tabla 7.30 Términos para la Restauración de Información (Continúa)

## **7.3 Instalación de Software**

### **ESTACIONES DE TRABAJO (Implementación en ambiente Monousuario).**

Para realizar la instalación del Sistema Manejador de Base de Datos, es necesario tener como requisito Windows 3.1 o Windows para Trabajo en Grupos 3.1, corriendo aplicaciones de 32 bits. Para poder verificar la facultad del ambiente operativo que corra aplicaciones de 32 bits existe una aplicación que puede verificar las instancias del sistema, esta aplicación se denomina *Win32s*. Es importante resaltar que si existen corriendo aplicaciones con uso de memoria cache solo de lectura, o haciendo uso del Microsoft SmartDrive u otro producto semejante que generen el uso de memoria cache solo de lectura, generara conflictos cuando se requiera hacer uso de las aplicaciones del RDBMS. No se deberá tener instalado ninguna clase de aplicaciones de compresión de disco, porque pueden generar problemas irreparables.

#### **Instalación de Win32s.**

1. Inicialize Windows en Modo Mejorado (Memoria Virtual, Enhanced).
2. Desde el Manejador de Programas (Program Manager) de Windows, elija el menú Archivo (File), después elija la opción de Ejecutar (Run) y aparecerá el dialogo de Ejecutar (Run).
3. Inserte el Disco Compacto de Personal Oracle 7 dentro del Manejador de Discos Compactos.
4. En la línea de comandos del dialogo de Ejecutar (Run), escriba : *g:\win32s\disk1\setup.exe*, donde *g* es la letra asignada al manejador lógico del Disco Compacto.
5. Seleccione Aceptar (OK).

#### **Instalación de Personal Oracle 7.**

1. Antes de comenzar su instalación cerciorarse de que corran las aplicaciones de 32 bits con Win32s.
2. Inserte su disco compacto de Personal Oracle 7 en su manejador de Discos Compactos. Verifique que realmente el mapeo de la letra asignada a su manejador de discos compactos pertenece a él.
3. Verifique el espacio en su disco duro, de acuerdo a las especificaciones de acuerdo a las especificaciones realizadas en el capítulo 2.
4. Inicialize Windows.



5. Verifique que realmente su Manejador de Discos Compactos se encuentra operando, haciendo uso del Manejador de Archivos (File Manager), en donde aparecerá el manejador de discos compactos, selecciones y compruebe que los direccionamientos son correctos.
6. Desde el Manejador de Programas (Program Manager) de Windows, elija el menú Archivo (File), después elija la opción de Ejecutar (Run) y aparecerá el dialogo de Ejecutar (Run).
7. En la línea de comandos del dialogo de Ejecutar (Run), escriba : *g:\windows\install\orainst.exe*, donde *g* es la letra asignada al manejador lógico del Disco Compacto.
8. Seleccione Aceptar (OK).
9. Seleccione el Idioma deseado.
10. Inserte Nombre de la Compañía y ubicación de la ruta de los directorios que se generaran para Oracle.
11. Adicione la ruta de los directorios para Oracle en su archivo *autoexec.bat*.
12. Seleccione el tipo de instalación " Complete".
13. Espere por el mensaje "Installation Complete", y seleccióné Aceptar (OK).  

Si sucede algún error o anomalía, siga las instrucciones proporcionadas por la ayuda de Windows o acuda a los manuales de instalación y guía de usuario de Personal Oracle 7 Ver. 7.1 para Windows
14. Desde el Manejador de Programas (Program Manager) de Windows, elija el menú Archivo (File), después elija la opción de Ejecutar (Run) y aparecerá el dialogo de Ejecutar (Run).
15. Inserte el Disco 1 de SICADE en el manejador de discos flexibles.
16. En la línea de comandos del dialogo de Ejecutar (Run), escriba : *a:\instalar.exe*, donde *a* es la letra asignada al manejador lógico del Discos Flexibles.
17. Seleccióné Aceptar (OK).
18. Cambie los discos del manejador de Discos Flexible conforme Instalar se lo indique.

Cuando se ha instalado la base de datos ORACLE y el sistema SICADE estos presentara cuatro tablespaces, cuatro archivos de base de datos, cuatro archivos de log y un diccionario de datos.

Los tablespace son System, User\_data, Temporary\_data y Rollback\_data.

*System*, almacena el contenido del diccionario de datos, incluyendo definiciones de tablas, vistas y procedimientos de almacenamiento necesarios para el Sistema de Mantenimiento de la Base de Datos Relacional (RDBMS). La información en esta área tiene mantenimiento automático.

*User\_data*, Cuando la base de datos es inicialmente instalada, este tablespace esta vacío. Por lo que hay que crear tablas e insertar datos en ellas, para generar la ocupación del espacio de este. Cuando se agregan usuarios en el sistema usando las herramientas del *User Manager*, utilizara automáticamente este tablespace.

*Temporary\_data*, La base de datos usa este tablespace para *scratchpad*, es decir datos de *scratch* creados por el Sistema de Mantenimiento de la Base de Datos Relacional (RDBMS), mientras se procesan ciertas opciones o características de los comandos SQL. Si se requiere incrementar el tamaño de este espacio, también crecerá el tablespace *User\_data* en tamaño.

*Rollback\_data*, las base de datos usara este tablespace para respaldar las transacciones que fallan por completo normalmente. Se hará lo necesario para expandir este tablespace si se tiene una gran ejecución o un volumen de datos transaccionados.

Los archivos son :*wbsys.ora*, *wbuser.ora*, *wdbtemp.ora* y *wdbrrs.ora* los cuales están asignados directamente a los tablespaces *System*, *User\_data*, *Temporary\_data* y *Rollback\_data*, respectivamente.

*Wdblog1.log* y *wdblog2.log* son tomados como salida en archivos de lectura del LOG de la base de datos. Además existen un archivo de *control ctl1.ora*, el cual contiene el nombre de la base de datos, los nombres y localizaciones de los archivos de datos, log y el tiempo de *stamp* de la creación de la base de datos..

Es importante recalcar los archivos de alerta de la base de datos *alert.log*, ya que estos archivos almacenan información importante acerca de los mensajes de error y excepciones que pueden ocurrir durante las operaciones de la base de datos, el archivo de alerta es la primera instancia en donde podremos verificar si ha ocurrido algún evento erróneo o incorrecto en la base de datos para determinar la causa que lo provocó. Este archivo es reinicializado cada vez que la base de datos es levantada.

Estos archivos son necesarios para iniciar (levantar) y correr el sistema de la base de datos.

## **SERVIDORES (Implementación en ambiente de Red)**

El RDBMS de ORACLE esta disponible prácticamente en todas las plataformas de hardware y sistemas operativos que existen hoy en día en el mercado. Iniciando con la versión 7, ORACLE provee la capacidad de procesamiento de consultas en paralelo y , en cada una de estas, es el primer RDBMS disponible para diversas plataformas SMP (Multiprocesamiento

Simétrico) y MPP (Procesamiento Paralelo Masivo). Además provee un juego completo de herramientas de integración de software, incluyendo un conjunto de herramientas I-CASE (Ingeniería de Software Asistido por Computadora Integrada). Con las siguientes características sobresalientes:

- Portabilidad en Hardware y Sistemas de Software.
- Un Amplio Soporte de Protocolos de Comunicación.
- Capacidad de Procesamiento Distribuido.
- Procesamiento Paralelo de Consultas.
- Diccionario de Datos Activo.
- Soporte SQL: Un lenguaje procedural (PL/SQL), Soporte de Precompiladores con SQL, Soporte de SQL ANSI y soporte de SQL extendido (SQL\*Plus).

Las herramientas de usuario final (Front-end) de ORACLE, incluye: las herramientas gráficas de SQL\*FORMS para el desarrollo de aplicaciones, SQL\*ReportWriter y los productos de SQL\*Menu. El RDMBS de ORACLE ofrece un sofisticado control de concurrencia, seguridad a nivel de registro, almacenamiento de consultas, disparadores asociados a eventos y entrada/salida asincrónica. Además, ORACLE soporta base de datos compartidas por medio del sistema de cache global, donde residen múltiples instancias en múltiples sistemas donde pueden compartir al mismo tiempo base de datos localizadas en discos compartidos. Las implementaciones de conectividad son características propias de los productos SQL\*Connect y SQL\*Net. Estos componentes permiten a ORACLE el soporte distribuido de sus consultas y actualizaciones, transparencia de la ubicación de los datos, autonomía de ubicación, acceso heterogéneo del DBMS e independencia de conectividad. SQL\*Net es una interface de conectividad heterogénea que soporta: DECnet, TCP/IP, SNA LU0, LU2 y APPC/LU6.2, IPX/SPX de Novell, Named Pipes, NetBIOS, X.25, OSI, Async y Banyan Vines, estos son los nombres de solo algunos protocolos de comunicación. SQL\*Connect soporta accesos con características de conectividad heterogénea en su DBMS tales como: DB2, SQL/DS y VAX RMS.

Uno de los mas importantes componentes del RDBMS de ORACLE es su diccionario de datos activo, el cual es un conjunto de tablas que contienen: índices, sinónimos, secuencias y agrupamientos (clusters). El diccionario de datos provee información acerca de nombres de usuarios, privilegios de usuario, table constraints, tipos de columnas, llaves primarias y foráneas, asignación de espacio para los objetos y datos auditables.

Dependiendo de su plataforma y sistema operativo, usted instalara su versión de Oracle 7, con sus respectivos programas de comunicaciones y protocolos (por ejemplo TCP/IP), entre el servidor y las terminales de la red, para poder generar un ambiente cliente/servidor. Debido al enfoque de este manual únicamente nos abocaremos a un ejemplo:

- Servidor: Plataforma Sequent.
  - Sistema Operativo *Dynix* (Modalidad de *UNIX*).
  - Programas de Conectividad y Comunicaciones : TCP/IP (*TCPCTL*) y Novell 4.0
  - Oracle 7.14
- Clientes: COMPAQ Intel 80486 / 16 MB de RAM / 50 Mhz / 500 MB en Disco Duro.
  - Windows 3.1 ó Windows para Trabajo en Grupos 3.1
  - Programas de Conectividad y Comunicaciones TCP/IP (Novell Lan WorkPlace)
  - Programas Run Times de :SQL\*Forms Ver. 2.5 y SQL\*Reports ver. 4.0

## **CONFIGURACIÓN DE IMPRESORAS Y TERMINALES**

Debido a la gran variedad de impresoras podemos determinar, que en nuestro ambiente cliente/servidor , las impresoras pueden adicionarse de dos maneras, como impresoras de uso compartido o de uso personal. Totalmente para el usuario es transparente ya que nuestro front end es proporcionado por Windows para Trabajos en Grupo 3.1 o Windows NT en un ambiente de red o Windows 3.1 y Windows para Trabajo en Grupo 3.1, esto origina que SICADE tome los DDL (Lenguaje de Definición de Datos) y DDE (Intercambio Dinámico de Datos) por omisión, es decir, las impresoras que estén definida y habilitadas en el icono de Impresoras (Printers) del Grupo de Programas (Program Group) Panel de Control (Control Panel) en Windows (cualquier opción) también serán reconocidas por SICADE.

Es necesario recordar el uso de dos tipos de impresoras diferentes que maneja SICADE, ya que es puede generar alguna alteración al buen funcionamiento del sistema cuando alguna de estas no se encuentra habilidad en el ambiente del cliente.

## **CONFIGURACIÓN DE ACCESOS**

La manipulación de la información es vital para el éxito de SICADE, por lo que , cuando se maneja erróneamente puede amenazar la integridad del sistema. SICADE proporciona unas características de seguridad basadas en las

herramientas que provee la herramienta de diseño, esto es con el propósito de obtener una seguridad extensa, sobre personas sin autorización, como de daños intencionales o inadvertidos.

Esta seguridad esta proporcionado por la concesión y supresión de privilegios sobre la base de datos, además de soportarse sobre la seguridad previa que posee el sistema operativo de la solución. SICADE posee dos tipos de usuarios agentes y supervisores, los cuales podrán únicamente actualizar el sistema, mientras los otros podrán realizar esto mismo y llevar la administración del sistema, respectivamente.(Para ingresar agentes usar gran conecta to user\_id identified by password\_id y para supervisores usar grant resource to user\_id identified by password\_id, para eliminar agentes o supervisores con revoke connect from user\_id, esta generación se realizara en una de las opciones de supervisión, que serán adicionales, a las de gestoria.)

## **7.4 Mantenimiento de Hardware y Software**

Plan detallado del mantenimiento/instalación de recursos, así como de periféricos para el sistema.(discos, memoria y dispositivos externos). Se recomienda que el mantenimiento del Hardware del Sistema reciba periódicamente atención preventiva y correctiva, con las siguientes características:

<b>Dispositivo</b>	<b>Ene</b>	<b>Feb</b>	<b>Mar</b>	<b>Abr</b>	<b>May</b>	<b>Jun</b>	<b>Jul</b>	<b>Ago</b>	<b>Sep</b>	<b>Oct</b>	<b>Nov</b>	<b>Dic</b>
<b>Discos</b>	A		BC		BC		A		BC		BC	
<b>Memoria</b>	A		BC		BC		A		BC		BC	
<b>Unidades de Respaldo</b>	A		BC		BC		A		BC		BC	
<b>Teclado</b>	A		BC		BC		A		BC		BC	
<b>Monitor</b>	A		BC		BC		A		BC		BC	
<b>Tarjetas Comunicaciones</b>	A		BC		BC		A		BC		BC	
<b>Fuentes de Poder y Cables</b>	A		BC		BC		A		BC		BC	

Tabla 7.31 Calendario de Mantenimiento

Eventos	Actividades
A	Mantenimiento Preventivo y Correctivo, con sustitución de dispositivos o piezas desgastadas, en base a los monitoreos y Reportes anteriores.
B	Monitoreo de la funcionalidad del Dispositivo
C	Reporte de la Funcionalidad del Dispositivo

Tabla 7.32 Descripción de Eventos para Calendario de Mantenimiento

Estas herramientas han sido elaboradas para poder obtener respaldos temporales o para depuración de datos, los cuales podrán ser almacenados en archivos planos y cuando se requiera estos datos serán reingresados o intercambiados al sistema.

El poder tener respaldos periódicos genera un alto grado de seguridad contra posibles problemas de pérdidas de información o simplemente cuando en la base de datos existe información de carácter histórico y únicamente genera que los procesos sean mas lentos, por lo que separar esta información, mantendrá al sistema en un máximo de rendimiento o desempeño.

### Respaldo de Información

Esta herramienta esta disponible para diferentes sistemas operativos en distintas plataformas de hardware. Únicamente nos abocaremos a describirla en un ambiente monousuario (WINDOWS 3.1).

### Procedimiento de Respaldo Tablas o Entidades

1. Dentro del Grupo de Utilerias de SICADE seleccionar "Respaldo de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Exporter".
5. Asegúrese de seleccionar la opción de "Tables".
6. Seleccione el botón de "Specify"
7. Aparecerá un dialogo con el siguiente nombre "Export Objects" y desplegara una lista de entidades o tablas que usted puede respaldar.

8. Seleccione las entidades o tablas que requiere respaldar desde la lista de la derecha y seleccione el botón de "Export". O si usted selecciona "Export All" respaldara la lista entera que se despliega.
9. Cuando finalice, seleccione el botón de "OK", y el dialogó de "Database Exporter" aparecerá.
10. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
11. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
12. Cuando este listo, seleccione el botón de "Export".
13. Fin del Proceso

### **Procedimiento de Respaldo Usuarios o Propietarios de Tablas.**

1. Dentro del Grupo de Utilerias de SICADE seleccionar "Respaldo de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Exporter".
5. Asegúrese de seleccionar la opción de "Owners"
6. Seleccione el botón de "Specify"
7. Aparecerá un dialogo con el siguiente nombre "Export Objects" y desplegara una lista de usuarios que usted puede respaldar.
8. Seleccione los usuarios que requiere respaldar desde la lista de la derecha y seleccione el botón de "Export". O si usted selecciona "Export All" respaldara la lista entera que se despliega.
9. Cuando finalice, seleccione el botón de "OK", y el dialogó de "Database Exporter" aparecerá.
10. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
11. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
12. Cuando este listo, seleccione el botón de "Export".
13. Fin del Proceso

### **Procedimiento de Respaldo Total de la Base de Datos.**

1. Dentro del Grupo de Utilerías de SICADE seleccionar "Respaldo de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Exporter".
5. Asegúrese de seleccionar la opción de "Full Database"
6. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
7. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
8. Cuando este listo, seleccione el botón de "Export".
9. Fin del Proceso

### **RESTAURACIÓN DE INFORMACIÓN**

#### **Procedimiento de Restauración de Tablas o Entidades.**

1. Dentro del Grupo de Utilerías de SICADE seleccionar "Restauración de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Importer".
5. Asegúrese de seleccionar la opción de "Tables"
6. Seleccione el botón de "Especify"
7. Aparecerá un dialogo con el siguiente nombre "Import Tables" y desplegara una lista de tablas que usted puede restaurar.
8. Seleccione las tablas que requiere restaurar desde la lista y seleccione el botón de "Import". O si usted selecciona "Import All" respaldara la lista entera que se despliega.
9. Cuando finalice, seleccione el botón de "OK", y el dialogó de "Database Importer" aparecerá.



10. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
11. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
12. Cuando este listo, seleccione el botón de "Import".
13. Fin del Proceso

### **Procedimiento de Restauración de los Usuarios.**

1. Dentro del Grupo de Utilerias de SICADE seleccionar "Restauración de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Importer".
5. Asegúrese de seleccionar la opción de "Users"
6. Seleccione el botón de "Especify"
7. Aparecerá un dialogo con el siguiente nombre "Import Users" y desplegara una lista de usuarios que usted puede restaurar.
8. Seleccione un nombre que requiere restaurar desde la lista.
9. Asumiendo que usted tiene privilegios de DBA, si usted requiere restaurar objetos de algún usuario dentro del esquema de otro usuario, seleccione el usuario desde el cual desea obtener los objetos.
10. Seleccione el botón de "Add"
11. Repita de los pasos 7 al 10 para agregar otros usuarios.
12. Cuando finalice, seleccione el botón de "Done", y el dialogó de "Database Importer" aparecerá.
13. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
14. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
15. Cuando este listo, seleccione el botón de "Import".
16. Fin del Proceso

### **Procedimiento de Restauración de Tablas o Entidades.**

1. Dentro del Grupo de Utilerías de SICADE seleccionar "Restauración de Datos"
2. Aparecerá un dialogo realizando una petición de acceso a la base de datos.
3. Insertar el password correcto, para el libre acceso hacia la base de datos.
4. Aparecerá un dialogo con el siguiente nombre " Database Importer".
5. Asegúrese de seleccionar la opción de "Full Database"
6. Seleccione el botón de "Especify"
7. Verifique todos los valores requeridos que aparecen en los objetos del dialogo.
8. Si es necesario, seleccione el botón de "Advanced" para posibles cambios en los valores.
9. Cuando este listo, seleccione el botón de "Import".
10. Fin del Proceso

### **Proceso de Arranque del Sistema**

Seguir procedimiento del manual, primeramente arrancando la base de datos y luego ejecutando la forma principal del sistema.  
Seguir procesos y métodos proporcionados por los manuales de la base de datos.

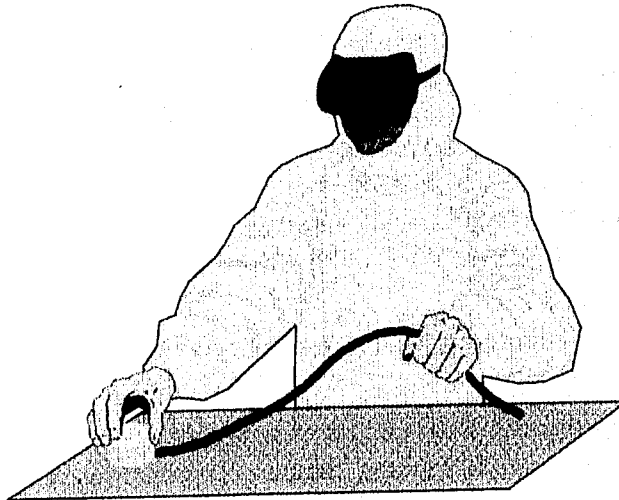
### **Respaldo / Restauración Temporal de la Base de Datos**

Realizar respaldos periódicos de la base de datos para obtener un respaldo metódico y programado para obtener satisfactoriamente el respaldo de la base de datos. Seguir procesos y métodos proporcionados por los manuales de la base de datos. Hacer uso de las herramientas Respaldo de Información y Restauración de Información, dentro del Grupo de Herramientas de SICADE.

# CAPÍTULO 8

## PLAN DE PRUEBAS

---



Aunque se haya hecho una labor perfecta de análisis, diseño y programación, se debe hacer algún esfuerzo para verificar que no existan errores. Generalmente se hace un trabajo "imperfecto" y es entonces cuando hay que realizar pruebas, las cuales se vuelven iterativas: la primera fase de ellas muestra la presencia de errores y las posteriores verifican si los programas corregidos funcionan correctamente.

El proceso de desarrollar casos de prueba de aceptación puede llevarse a cabo en paralelo con las actividades de implantación del diseño y de la programación, para que, cuando los programadores terminen de escribir sus programas y de realizar sus propias pruebas locales, el equipo del analista-usuario esté listo con sus propios casos de prueba.

### **8.1 Tipos de prueba**

Los dos tipos de pruebas más comunes se conocen como *prueba ascendente* y *descendente*.

El enfoque ascendente empieza por probar módulos individuales pequeños separadamente; esto a menudo se conoce como prueba de unidades, prueba de módulos, o prueba de programas. Luego, los módulos individuales se combinan para formar unidades cada vez más grandes que se probarán en masa; esto se conoce como prueba de subsistemas. Finalmente, todos los componentes del sistema se combinan para probarse; esto se conoce como prueba del sistema, y suele estar seguido de las pruebas de aceptación, donde se permite al usuario usar sus propios casos de prueba para verificar que el sistema esté trabajando de manera correcta.

El enfoque descendente empieza con un esqueleto del sistema; es decir, la estrategia de prueba supone que se han desarrollado los módulos ejecutivos de alto nivel del sistema, pero que los de bajo nivel existen sólo como módulos vacíos (sin hacer algo). Dado que muchas de las funciones detalladas del sistema no se han implantado, las pruebas iniciales están muy limitadas; el propósito es simplemente comenzar a ejecutar las interfases entre los subsistemas principales. Las pruebas siguientes abarcan y tratan aspectos cada vez más detallados del sistema.

También es importante estar familiarizado con los siguientes tipos de pruebas:

- *Prueba funcional*: Esta es la forma más común de prueba; su propósito es asegurar que el sistema realiza sus funciones normales de manera correcta. Así, los casos de prueba se desarrollan y se alimentan al sistema; las salidas (y los resultados de los archivos actualizados) se examinan para ver si son correctos.

- **Prueba de recuperación:** El propósito de este tipo de prueba es asegurar que el sistema pueda recuperarse adecuadamente de diversos tipos de fallas. Esto es de particular importancia en los sistemas en línea grandes, al igual que en varios tipos de sistemas de tiempo real que controlan dispositivos físicos y/o procesos de fabricación. Las pruebas de recuperación pueden requerir que el equipo que realiza el proyecto simule (o provoque) fallas de hardware, fallas de corriente, fallas en el sistema operativo; etc.
- **Prueba de desempeño:** El propósito de este tipo de prueba es asegurar que el sistema pueda manejar el volumen de datos y transacciones de entrada especificados en el modelo de implantación del usuario, además de asegurar que tenga el tiempo de respuesta requerido. Esto puede requerir que el equipo que realiza el proyecto simule una gran red de terminales en línea, de manera que se pueda "engañar" al sistema para que "crea" que está operando con una gran carga.

Un concepto que no se debe pasar por alto es el de la *prueba exhaustiva*. En el proyecto ideal, se generarían casos de prueba para cubrir cada entrada posible y cada combinación posible de situaciones que el sistema pudiera enfrentar alguna vez; luego, se probaría de manera exhaustiva para asegurar que su comportamiento sea perfecto. Sólo existe un problema con esto: no funciona. El número de casos de prueba para un sistema grande y complejo típico es tan increíblemente grande, a menudo del orden de 10'100 casos de prueba distintos o más, que aún si se pudiera realizar una prueba cada milésima de segundo tomaría más de la edad del universo terminar todas las pruebas.

Por lo tanto, se harán pruebas exhaustivas solo y únicamente de aquellas funciones que sean de vital importancia, reduciendo así el margen de posibilidades de casos de pruebas.

Entonces, un documento de plan de pruebas típico contendrá la siguiente información:

- **Propósito de la prueba:** cuál es el objetivo de la prueba, y qué parte del sistema se está probando
- **Localización y horario de la prueba:** dónde y cuándo se hará.
- **Descripciones de la prueba:** descripción de las entradas que se proporcionarán al sistema, y las salidas y resultados que se anticipan. Usualmente se darán descripciones de las entradas de prueba en el formato de diccionario de datos.
- **Procedimientos de prueba:** descripción de cómo se deben preparar y presentar los datos de prueba al sistema; cómo capturar los resultados de salida, cómo analizar los resultados de las pruebas, etc.

En este plan de pruebas, se realizarán "*pruebas de menús principales*" y "*pruebas funcionales*" a cada pantalla para verificar que trabajen correctamente.

En las "pruebas de menús principales" existe un aspecto en común, este es el *procedimiento de prueba*, el cual consiste en hacer un "click" en cada opción para comprobar que se despliega la pantalla esperada correspondiente.

En cuanto a las "pruebas funcionales", éstas consideran la alimentación de datos y el resultado esperado de su procesamiento.

## **8.2 Prueba de menús principales**

- *Tipo de prueba:* Descendente
- *Propósito de prueba:* Verificar que cada opción del menú principal despliega un submenú con las opciones correspondientes.
- *Descripción de la prueba:* En la tabla 8.1 se muestran los menús principales, el submenú esperado al seleccionar el menú y la aceptación de lo esperado.

<b>MENÚ PRINCIPALES</b>	<b>SUBMENÚ ESPERADO</b>
<b>CATÁLOGOS</b>	Altas, Bajas y Cambios Consultas a Catálogos
<b>DOCUMENTOS</b>	Altas y Cambios de Gestiones Consultas Generales Consultas por Árbol Consultas Alfabéticas Registro de Transformación Borrar Gestiones Consultas Especiales
<b>REPORTES</b>	Turnos Documentos Turnados Discursos Viajes Oficios Documentos para Firma Decreto Catálogo Alfabético Invitaciones Estadísticas
<b>UTILERÍAS</b>	Respaldo de Datos Recuperación de Datos

Tabla 8.1 Menús Principales

## PRUEBA DE SUBMENÚS

- *Tipo de prueba:* Descendente
- *Propósito de prueba:* Verificar que cada opción de cada submenú despliega a su vez otro submenú, en el cual se presentan los diferentes catálogos con los que el sistema trabaja.
- *Descripción de la prueba:* En la tabla 8.2 se muestran las opciones de submenú y los submenús esperados.

## OPCIÓN CATÁLOGOS

OPCIÓN DE SUBMENÚ	SUBMENÚ ESPERADO
Altas, Bajas y Cambios	Clasificación Documentos Instrucciones Temas Turnados Expedientes Dependencias
Consultas a Catálogos	Clasificación Documentos Instrucciones Temas Turnados Expedientes Dependencias

Tabla 8.2 Opciones de Submenús

## PRUEBA DE SUBMENÚS

- *Tipo de prueba:* Descendente
- *Propósito de prueba:* Verificar que cada opción de cada submenú despliega una pantalla de trabajo con las opciones correspondientes.
- *Descripción de la prueba:* En las tablas 8.3, 8.4, 8.5, 8.6 y 8.7 se muestran los submenús, la pantalla esperada y las opciones de trabajo correspondientes.

<b>SUBMENÚ</b>	<b>PANTALLA ESPERADA</b>
Clasificación	CATÁLOGO DE CLASIFICACIÓN
Documentos	CATÁLOGO DE DOCUMENTOS
Instrucciones	CATÁLOGO DE INSTRUCCIONES
Temas	CATÁLOGO DE TEMAS
Turnados	CATÁLOGO DE TURNADOS
Expedientes	CATÁLOGO DE EXPEDIENTES
Dependencias	CATÁLOGO DE DEPENDENCIAS

Tabla 8.3 Subopción altas,bajas y cambios de opción catálogos

<b>SUBMENÚ</b>	<b>PANTALLA ESPERADA</b>
Clasificación	CATÁLOGO DE CLASIFICACIÓN
Documentos	CATÁLOGO DE DOCUMENTOS
Instrucciones	CATÁLOGO DE INSTRUCCIONES
Temas	CATÁLOGO DE TEMAS
Turnados	CATÁLOGO DE TURNADOS
Expedientes	CATÁLOGO DE EXPEDIENTES
Dependencias	CATÁLOGO DE DEPENDENCIAS

Tabla 8.4 Sunopción consultas a catálogos de opción catálogos

## OPCIÓN DOCUMENTOS

<b>SUBMENÚ</b>	<b>PANTALLA ESPERADA</b>
Altas y Cambios de Gestiones	CONTROL DE GESTIONES
Consultas Generales	CONSULTAS GENERALES
Consultas por Árbol	CONSULTAS POR ARBOL
Consultas Alfabéticas	CONSULTAS ALFABÉTICAS
Registro de Transformación	REGISTRO DE TRANSFORMACIÓN
Borrar Gestiones	BORRAR GESTIONES
Consultas Especiales	CONSULTAS ESPECIALES

Tabla 8.5 Opción del Menú de Documentos



## OPCIÓN REPORTES

SUBMENÚ	PANTALLA ESPERADA
Turnos	IMPRESIÓN DE OFICIOS TURNADOS
Documentos Turnados	DOCUMENTOS TURNADOS
Discursos	RELACIÓN DE DISCURSOS
Viajes	PROGRAMA DE VIAJES
Oficios	RELACIÓN DE OFICIOS DESPACHADOS
Documentos para Firma	DOCUMENTOS PARA FIRMA
Decreto	RELACIÓN DE PROYECTOS DE DECRETO
Catálogo Alfabético	CATÁLOGO ALFABÉTICO
Invitaciones	RELACIÓN DE INVITACIONES
Estadísticas	ESTADÍSTICAS DE DOCUMENTOS

Tabla 8.6 Opción del submenú de reportes

## OPCIÓN UTILERÍAS

SUBMENÚ	PANTALLA ESPERADA
Respaldo de Datos	DATABASE EXPORTER
Recuperación de Datos	DATABASE IMPORTER

Tabla 8.7 Opción de submenú de utilerías

Al realizar las pruebas, se pudo constatar que al elegir cada una de las opciones, se despliega la pantalla de trabajo correspondiente.

### 8.3 Prueba de pantallas

- *Tipo de prueba:* Descendente
- *Propósito de prueba:* Verificar que cada pantalla cuente con las funciones correspondientes para el auxilio en la captura de los datos.
- *Descripción de la prueba:* En las tablas 8.8, 8.9, 8.10, 8.11, 8.12 se muestran todas las pantallas y cada una de las funciones esperadas, así como la aceptación de la prueba.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Clasificación	LIMPIAR FORMA LIMPIAR CAMPO CORTAR

Tabla 8.8 Pantalla catálogo de clasificación de subopción altas, bajas y cambios de catálogos

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
	COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.8 Pantalla catálogo de clasificación de subopción altas, bajas y cambios de catálogos  
(Continuación)

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Documentos	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.9 Pantalla catálogo de documentos de subopción altas, bajas y cambios de catálogos

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Instrucciones	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.10 Pantalla catálogo de instrucciones de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Temas	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.11 Pantalla catálogo de temas de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Turnados	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR SALVAR LIMPIAR BLOQUE LIMPIAR REGISTRO EDITAR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO DEFINIR CONSULTA EJECUTAR CONSULTA BÚSQUEDAS DUPLICAR CAMPO DUPLICAR REGISTRO

Tabla 8.12 Pantalla catálogo de turnados de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Expedientes	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.13 Pantalla catálogo de expedientes de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Dependencias	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR LISTAR WORD SALIDA A SISTEMA OPERATIVO BORRAR MENÚ ANTERIOR

Tabla 8.14 Pantalla catálogo de dependencias de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Clasificación	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE

Tabla 8.15 Pantalla catálogo de clasificación de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Documentos	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE

Tabla 8.16 Pantalla catálogo de documentos de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Instrucciones	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE

Tabla 8.17 Pantalla catálogo de instrucciones de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Temas	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE

Tabla 8.18 Pantalla catálogo de temas de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Turnados	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE CAMPO SIGUIENTE CAMPO PREVIO

Tabla 8.19 Pantalla catálogo de turnados de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Expedientes	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR GENERAR REPORTE

Tabla 8.20 Pantalla catálogo de expedientes de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo de Dependencias	LIMPIAR FORMA LISTAR WORD SALIDA A SISTEMA OPERATIVO

Tabla 8.21 Pantalla catálogo de dependencias de subopción altas, bajas y cambios de catálogos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Control de Gestiones	LIMPIAR FORMA LIMPIAR CAMPO CORTAR COPIAR PEGAR MENÚ ANTERIOR GENERAR REPORTE LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR SALVAR LIMPIAR BLOQUE LIMPIAR REGISTRO EDITAR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO DEFINIR CONSULTA EJECUTAR CONSULTA

Tabla 8.22 Pantalla control de gestiones de opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
	DUPLICAR CAMPO DUPLICAR REGISTRO BLOQUE PREVIO BLOQUE SIGUIENTE REGISTRO PREVIO REGISTRO SIGUIENTE PRIMER REGISTRO ÚLTIMO REGISTRO ACEPTAR

Tabla 8.22 Pantalla control de gestiones de opción documentos. (Continuación)

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Consultas Generales	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.23 Pantalla consultas generales de opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Consultas por Árbol	LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR

Tabla 8.24 Pantalla consultas por árbol de opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Consultas Alfabéticas	LIMPIAR FORMA WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR

Tabla 8.25 Pantalla consultas alfabéticas opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Registro de Transformación	LIMPIAR FORMA LIMPIAR CAMPO LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO

Tabla 8.26 Pantalla registro de transformación opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Borrar Gestiones	LIMPIAR FORMA WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR

Tabla 8.27 Pantalla borrar gestiones opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Consultas Especiales	LIMPIAR FORMA LIMPIAR CAMPO LISTAR WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.28 Pantalla consultas especiales opción documentos.



<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Impresión de Oficios Turnados	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENÚ ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.29 Pantalla impresión de oficios turnados opción documentos.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Documentos Turnados	LIMPIAR FORMA WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR DOCUMENTOS TURNADOS CON RESOLUCIÓN REPORTE VERIFICADOS REPORTE INVITACIONES REPORTE DE AUDIENCIAS REPORTE DE OBSEQUIOS

Tabla 8.30 Pantalla documentos turnados opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Relación de Discursos	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO REPORTE DE RELACIÓN DE DISCURSOS

Tabla 8.31 Pantalla relación de discursos turnados opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Programa de Viajes	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.32 Pantalla programa de viajes tomados opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Relación de Oficios Despachados	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.33 Pantalla relación de oficios despachados tomados opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Documentos para Firma	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.34 Pantalla documentos para firma opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Relación de Proyectos de Decreto	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO REPORTE DE DECRETO GENERAL REPORTE DE DECRETO ESPECIAL

Tabla 8.35 Pantalla relación de proyectos de decreto opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Catálogo Alfabético	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.36 Pantalla catálogo alfabético opción reportes.

<b>PANTALLA ESPERADA</b>	<b>FUNCIÓN</b>
Relación de Invitaciones	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.37 Pantalla relación de invitaciones opción reportes.

PANTALLA ESPERADA	FUNCIÓN
Estadísticas de Documentos	LIMPIAR FORMA LIMPIAR CAMPO WORD SALIDA A SISTEMA OPERATIVO MENU ANTERIOR LIMPIAR TODO CAMPO SIGUIENTE CAMPO PREVIO GENERAR REPORTE

Tabla 8.38 Pantalla estadística de documentos opción reportes.

PANTALLA ESPERADA	FUNCIÓN
Database Exporter	DATABASE LOGON Username Password Database EXPORT FILE Browse TO BE EXPORTED Tables Owners Full Database Specify EXPORT THESE OBJECTS Grants Rows Constraints Indexes ADVANCED Record Length (Bytes) Buffer Size (Bytes) Record Export in System Data Compress when imported Keep Components Consistent Increment Type Statistics Type

Tabla 8.39 Pantalla database exporter de opción de utilerías.

PANTALLA ESPERADA	FUNCIÓN
	Generate Log File Browse ADVANCED (continuación...) Defaults Cancel Help OK DATABASE Username Password Database DEFAULTS CANCEL HELP EXPORT

Tabla 8.39 Pantalla database exporter de opción de utilerías. (Continuación)

PANTALLA ESPERADA	FUNCIÓN
Database Importer	DATABASE LOGON Username Password Database IMPORT FILE Browse IGNORE CREACIÓN ERRORS IMPORT MODE Tables Owners Full Database TABLES TO IMPORT Specify OBJECTS TO IMPORT Grants Rows Indexes ADVANCED Record Length (Bytes)

Tabla 8.40 Pantalla database importer de opción de utilerías.

PANTALLA ESPERADA	FUNCIÓN
Database Importer	Buffer Size (Bytes) Commit after each array Overwrite existing data files Generate Log File Browse Write Index-Creation Commands to File Browse Increment Type None System Restore Defaults Cancel Help OK DATABASE Username Password Database DEFAULTS CLOSE HELP IMPORT

Tabla 8.40 Pantalla database importer de opción de utilerías. (Continuación)

Las pruebas de menús y submenús se realizaron satisfactoriamente, comprobando así que el sistema tiene buen rendimiento y esta cumpliendo con las expectativas para lo que fue creado.

## **8.4 Pruebas Funcionales**

Con fines de ilustración, se presentan únicamente las pruebas realizadas en los submenús de *Documentos* y *Reportes*, ya que éstos contienen las funciones de mayor peso dedicadas al control de la documentación.

- *Tipo de prueba:* Descendente, funcional.
- *Propósito de la prueba:* Verificar que el sistema genera la información ya detallada en el manual del usuario y emita los reportes ya especificados.

- *Descripción de la prueba:* Consiste en proporcionar los datos propuestos por los usuarios en los campos correspondientes de cada pantalla del submenú, con el fin de obtener los reportes y/o resultados que cumplan con los requerimientos del cliente.

### Submenú Documentos

**OPCIÓN:** Altas y Cambios de Gestiones.

**PANTALLA:** Control de Gestiones.

**PRUEBA:** Alta de Documentos

Para dar de alta un documento es necesario escribir los siguientes datos:

<i>Campo</i>	<i>Valor</i>
Tipo	G
Número	1234
Clasificación	N+
Documento Externo	Escrito
Fecha Documento	19/01/96
Fecha Recepción	19/01/96
Fecha Despacho	<nulo> Se asigna hasta que se turne
Fecha Descargo	<nulo> Se asigna hasta que el turnado conteste
Fecha Evento	<nulo> Se asigna cuando sea evento
Hora	<nulo> Si es evento se asigna
Fecha Archivo	<nulo> Se asigna hasta que se archiva el documento
Dependencia	Coordinación General de Transporte
Digitalizar	N
Asunto	Se implementará el doble no circula a partir del día 20 enero
Remitente	Carlos Ordoñez Mondragón
Referencia Alfabética	Problema Ambiental
Tema	Implementación del Doble No Circula
Expediente	0

*Resultado Esperado:* Documento dado de alta.

**OPCIÓN:** Altas y Cambios de Gestiones.

**PANTALLA:** Control de Gestiones.

**PRUEBA:** Consulta de Documentos

Para realizar la Consulta de un documento es necesario escribir los siguientes datos:

Campo	Valor
Tipo	G
Número	1234
Clasificación	N+
Documento Externo	Escrito

*Resultado Esperado:* Desplegó los datos de este registro al ejecutar la búsqueda.

**OPCIÓN:** Consultas Generales.

**PANTALLA:** Consultas Generales.

**PRUEBA:** Consulta de Documentos

Para realizar la Consulta de un documento es necesario escribir los siguientes datos:

Campo	Valor
Identificador	1 al 10
Tipo	G
Clasificación	N+,I+
Turnado A	PN

*Resultado Esperado:* Desplegó los datos de los registros que se pidieron con las características anteriores. Cabe mencionar que la búsqueda se puede realizar por los siguientes campos: rango de número de documento, documento externo, rango de fechas de documento, recepción, despacho, descargo, evento, archivo, hora, dependencia, remitente, referencia alfabética, tema, expediente e instrucciones.

**OPCIÓN:** Consultas de Árbol.

**PANTALLA:** Consultas de Árbol.

**PRUEBA:** Consulta de Antecedentes y Subsecuentes

Para realizar la Consulta del antecedente y subsecuente de un documento es necesario escribir el siguiente dato:

Campo	Valor
Identificador	1490

*Resultado Esperado:* Desplegado del antecedente del documento 1490.



**OPCIÓN:** Consultas de Árbol.

**PANTALLA:** Consultas de Árbol.

**PRUEBA:** Consulta de Antecedentes y Subsecuentes

Para realizar la Consulta del antecedente y subsecuente de un documento es necesario escribir el siguiente dato:

Campo	Valor
Identificador	1490

*Resultado Esperado:* Despliega el antecedente del documento 1490.

**OPCIÓN:** Consultas Alfabéticas.

**PANTALLA:** Consultas Alfabéticas

**PRUEBA:** Consulta Alfabéticas

Para realizar la Consulta Alfabética de un documento es necesario escribir el siguiente dato:

Campo	Valor
Palabra a buscar	Ingreso

*Resultado Esperado:* Despliega los registros que en dependencia o referencia alfabética o tema contienen la palabra ingreso.

**OPCIÓN:** Registro de Transformación.

**PANTALLA:** Registro de Transformación

**PRUEBA:** Registro de Transformación

Para realizar la generación de un antecedente y subsecuente, es necesario teclear los siguientes datos:

Campo	Valor
Identificador	123
tipo de Documento	G
Número de Documento	555

*Resultado Esperado:* Despliega la confirmación y la realización del Subsecuente Generado.

**OPCIÓN:** Bajas de Gestiones.

**PANTALLA:** Módulo de Bajas

**PRUEBA:** Dar de baja un registro

Para dar de baja un registro es necesario teclear y confirmar el siguiente dato:

Campo	Valor
Identificador	123

*Resultado Esperado:* Despliega los datos del documento cuyo identificador es 123 y pide la confirmación para ser borrado éste registro.

**OPCIÓN:** Consultas Especiales.

**PANTALLA:** Consultas Especiales

**PRUEBA:** Consultas Especiales

Para la generación del reporte de consultas especiales es necesario proporcionar los siguientes datos:

Campo	Valor
Rango de Identificador	1 al 23
Tipo de Documento	G
Clasificaciones	N+ ,I+
Turnos	PN,JSD,GOM
Instrucción	A,B,C,

*Resultado Esperado:* Generación del reporte de Consultas Especiales

**OPCIÓN: Reportes.**

**PRUEBA: Generación de Reportes**

Para la generación del reportes es necesario proporcionar los siguientes datos:

Campo	Valor
Reporte de Turnos	
No. de Documento	1 al 23
Turnado	PN,JSD,GOM
Documento Turnados	
Número de Documentos	1,2,3,4,5,6,7,8,9,10
Relación de Discursos	
Rango de Identificador	1 al 23
Fecha de Recepción	01/01/96 al 27/01/96
Fecha de Evento	01/01/96 al 27/01/96
Tipo de Documento	G,O,D
Turnado	PN,JSD,GOM
Clasificación	N+,I+
Instrucción	A,B,C

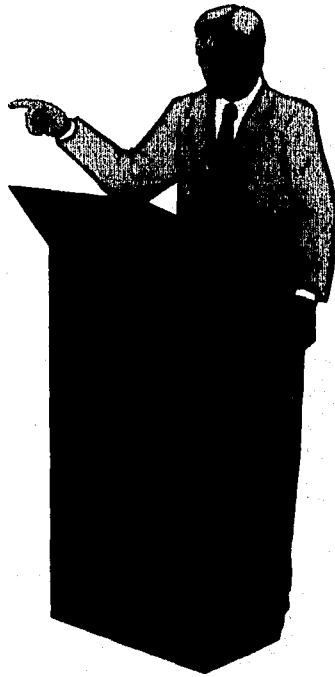
En general para todos los reportes pide los siguientes datos: rango de identificador, número de documento, fecha de recepción, evento, turnado, clasificación, tipo de documento e instrucción.

**Resultado Esperado:** Generación del reporte de Consultas Especiales

# CAPÍTULO 9

CONCLUSIONES

---



Para un estudiante de Ingeniería en Computación, la posibilidad de participar en la construcción de un sistema para una empresa reviste la gran oportunidad de poner en práctica los conocimientos obtenidos durante la carrera, desarrollarse como profesionista, engrandecer la institución escolar de donde es egresado, generar soluciones o herramientas útiles a la sociedad, difundir la cultura o simplemente crecer como persona. El Sistema de Control y Automatización de Documentos a gran Escala representó para nosotros esa gran oportunidad.

En la Coordinación Administrativa de la Secretaría de Hacienda y Crédito Público que controla y administra la documentación que llega a dicha entidad gubernamental, se detectó la problemática ya comentada en el primer capítulo. Esto nos motivó a pensar en un sistema que auxiliara al personal de la citada área.

El inicio de este trabajo no fue fácil, ya que actualmente todavía existen personas que no confían en los sistemas y hay un temor o rechazo implícito ante la idea de utilizar una computadora.

Sin embargo, esto fue una motivación extra porque dedicamos mayor esfuerzo y atención en el desarrollo de este trabajo para convencer a los usuarios de que el sistema ayudaría a hacer más eficiente su labor, a tener más control sobre los documentos y contar con información estadística veraz y oportuna para tomar decisiones, lo cual es fundamental en la dirección de una empresa.

La etapa de entrevistas a los usuarios fue complicada por el hecho de que ellos no estaban interesados en tener un sistema que en cierto momento los "sustituyera". O bien, por revelar ante personas desconocidas sus métodos y forma de trabajar.

Una vez que se logró entrevistar a las personas involucradas con la problemática se distinguió que el sistema a construir era una aplicación que podía ser usada en cualquier empresa en donde el flujo de la documentación es constante y voluminosa.

Los principales requerimientos de los usuarios son: administración, control, seguimiento, pero sobre todo disminuir el riesgo de perder información. Así como disminuir el gasto en el personal necesario para el seguimiento de la documentación dentro de la empresa.

El director personalmente puede cubrir sus necesidades de información controlando la documentación que llega a la empresa dirigida hacia él. Además, no será necesario que él la conteste, ya que podrá asignar a cualquier subordinado la resolución de la gestión tratada en un documento.

Una vez identificados los requerimientos, se procedió a elaborar los primeros diagramas de flujo de datos para esquematizar cómo se procesa la información y así provocar la participación de los usuarios. Ellos son los expertos en la problemática y el sistema a crear debe considerar todas los casos y situaciones que se presenten durante la administración manual de la documentación.

Después de tener exhaustivas revisiones a los diagramas de flujo de datos, en las que se resaltaron deficiencias, adversidades, duplicación de funciones, actividades redundantes, etc, se fue conformando finalmente la nueva estructura de los procedimientos, delimitando las acciones correspondientes al usuario y al propio sistema. El resultado de todo esto es el diagrama de flujo de datos definitivo que daría paso posteriormente al diseño del sistema.

A pesar de tener un esquema definitivo del sistema, el diseño no dejó de ser una actividad difícil, ya que se fueron conformando los módulos correspondientes pensando en la plataforma de programación, es decir, en los productos de ORACLE.

Estas herramientas, como se mencionó en su momento, era un recurso a ser aprovechado y afortunadamente cumplió con los requerimientos propios del sistema, de otra forma, se hubiera planteado la posibilidad de utilizar otra plataforma de programación.

Consideramos que es muy importante encontrar el equilibrio entre los recursos que se tienen y el impacto provocado al utilizarlos, es decir, si el aprovechar el lenguaje de programación existente implica que el sistema tenga un bajo rendimiento, se eleve el costo y no se cumplan los objetivos iniciales, es preferente adquirir el software de desarrollo adecuado. Sin embargo, la naturaleza del SICADE permitió utilizar los productos de ORACLE.

La etapa de programación fue simple en relación a las etapas ya explicadas. Únicamente se trasladaron las miniespecificaciones al código de programación de ORACLE.

Conjuntamente con la programación se fueron realizando las pruebas al sistema para detectar las deficiencias u omisiones de los acuerdos ya tomados. O bien, para tomar en cuenta y programar las disposiciones o estrategias de último momento. Se observó que funcionalmente se obtuvieron los submenús, pantallas, funciones y reportes esperados.

De esta forma el sistema obtenido cumple los objetivos planteados inicialmente, entre ellos el de simplificar los procedimientos efectuados en forma manual. Las ventajas obtenidas son control de la documentación, seguimiento de un documento en particular, simplicidad de funciones, eficiencia y rapidez en la generación de reportes, manejo de alto volumen de datos y veracidad en la información; además de disminuir costos a corto y mediano plazo.

Actualmente el buen manejo de la información es primordial para el desempeño de cualquier empresa o institución. El tener acceso inmediato a los datos permite tomar decisiones que pueden repercutir en el éxito o el fracaso. Por esto, SICADE se perfila como una gran herramienta para la administración y control de documentos, ofreciendo una interface sencilla y agradable para su manejo.

Lo anterior está respaldado por los manuales del usuario, de capacitación y de instalación. Estos documentos representan un gran apoyo para los usuarios que se inicien en el manejo total del SICADE.

La experiencia obtenida en este trabajo es invaluable porque se asumió la responsabilidad del sistema en forma total. Lo que al inicio parecía una idea poco útil e intrascendente, se fue convirtiendo poco a poco en un proyecto de gran importancia. Nos dimos cuenta que las relaciones humanas son indispensables para hacer un buen análisis, ya que los usuarios, de rechazarlos a hacer el sistema de su propiedad, significó una de las más grandes satisfacciones en el transcurso de este trabajo.

Además se pusieron en práctica los conocimientos obtenidos en forma académica y verlos cristalizados en una aplicación que es útil y que está en producción, reviste un logro importantísimo.

Durante el desarrollo de este trabajo, observamos que todavía hace falta crear una cultura en la sociedad, enfocada a considerar a los sistemas como herramientas actuales, útiles, provechosas y erradicar el temor y el rechazo sin fundamento a las computadoras.

Asimismo, consideramos que al realizar este sistema cumplimos con nuestros propios objetivos, ya que hemos obtenido reconocimiento y se nos ha ubicado como unos profesionistas serios, capaces y comprometidos con la sociedad y nuestro presente. Así como confirmar que la Facultad de Ingeniería y la Universidad Nacional Autónoma de México siguen siendo instituciones que le dan a nuestro país personas útiles.

**"Por mi raza hablará el espíritu."**

**Martha Laura López Coquis**

**Juan Vladimir Padilla Cerón**

**Raúl Ignacio Reséndiz Garcés**

**Guillermo Soto Tinoco**

# APÉNDICE A

## LISTADO PARCIAL DEL SISTEMA

---



El código fuente del sistema esta formado por: código fuente de formas, código fuente de User-Exit que son programas en ProC y código de reportes. Todo este código generado por las herramientas de Oracle se integran para formar toda la aplicación.

La figura A-1 muestra la estructura general del sistema, abajo del nombre de los elementos, en letra *cursiva* se indica el nombre de la forma que se ejecuta.

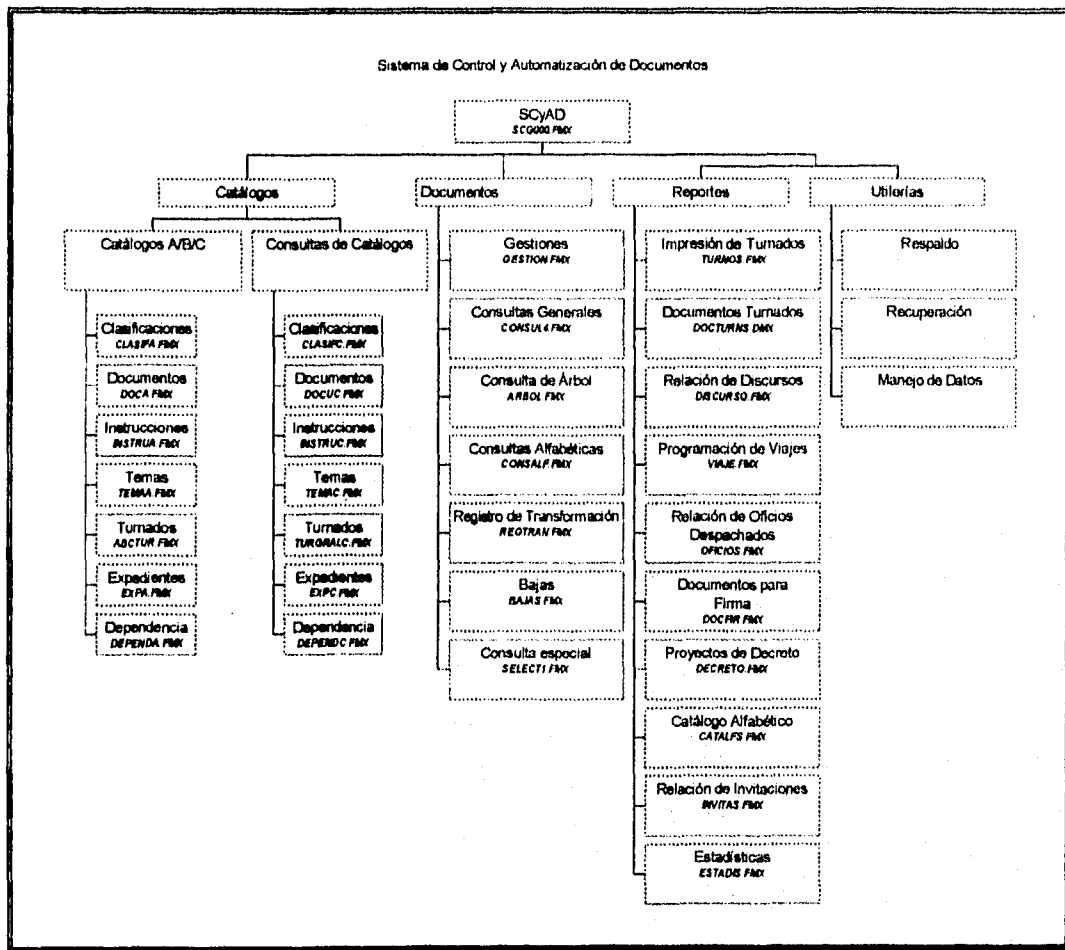


Figura A-1 Estructura General del Sistema.

Dadas las dimensiones del sistema, el código desarrollado y generado es muy extenso, razón por la cual solo se muestran algunos listados.

## **A.1 Listado de Formas**

El sistema esta formado por 32 formas de las cuales sólo se muestra el listado de una de ellas (se muestra la forma de TURGRALC.FMX, al principio del listado se explica el por que se eligió esta forma como ejemplo).

Debido a que Oracle genera demasiado código fuente para las formas, no es posible mostrar el código de todas las formas ya que estamos hablando de aproximadamente 1000 hojas de código, por ello solo se incluye como ejemplo, el código de una sola forma.

La tabla A-T1 muestra las formas creadas, su nombre y su función dentro del sistema:

<b>FORMA</b>	<b>FUNCIÓN</b>
ABCTUR.FMX	ABC de Turnados
ARBOL.FMX	Consultas de Arbol
BAJAS.FMX	Bajas de Gestiones
CATALFS.FMX	Catálogo Alfabético
CLASIFA.FMX	ABC de Clasificaciones
CLASIFC.FMX	Consultas de Clasificaciones
CONSALF.FMX	Consultas Alfabéticas
CONSUL4.FMX	Consultas Generales
DECRETO.FMX	Proyectos de decreto
DEPENDA.FMX	ABC de Dependencia
DEPENDC.FMX	Consultas de Dependencias
DISCURSO.FMX	Discursos
DOCA.FMX	ABC de Documentos
DOCFIR.FMX	Documentos para Firma
DOCTURNS.FMX	Documentos Turnados
DOCUC.FMX	Consultas de Documentos
ESTADIS.FMX	Estadísticas
EXPA.FMX	ABC de Expedientes
EXPC.FMX	Consultas de Expedientes
GESTION.FMX	AC de Gestión
INSTRUA.FMX	ABC de Instrucciones
INSTRUC.FMX	Consultas de Instrucciones
INVITAS.FMX	Relación de Invitaciones
OFICIOS.FMX	Relación de Oficios
REGTRAN.FMX	Registro de Transformación
SCG000.FMX	Menú principal
SELECT1.FMX	Consultas Especiales
TEMAA.FMX	ABC de Tema
TEMAC.FMX	Consultas de Temas
TURGRALC.FMX	Consultas de Turnados
TURNOS.FMX	Impresión de Turnos
VIAJE.FMX	Programas de Viajes

Tabla A-T1 Formas.

## **A.2 Listado de User Exit**

Un User-Exit es un programa en C o en ProC que al ser compilado con librerías de Oracle, se usa como una función, que puede ser llamada desde una forma creada en SqlForms.

Se utilizaron User-Exit debido a que en SqlForms no se pueden hacer consultas muy complicadas o variables, en cambio con la utilización de User-Exit se elaboran la sentencia de los Select's en forma variable permitiendo mayor flexibilidad y grado de complejidad.

Como los User-Exit que se hicieron son pequeños si es posible mostrar el listado de cada uno al final de este apéndice.

La tabla A-T2 muestra los User-Exit creados, su nombre y si función:

<b>USER EXIT</b>	<b>FUNCIÓN</b>
catalf	Genera un SQL externo (catalf.sql), el cual se ejecutara por medio del Reporte de Viajes.
conseps	Genera un SQL externo (conseps.sql), el cual se ejecutara por medio de la forma de Consultas Especiales.
decret	Genera un SQL externo (decret.sql), el cual se ejecutara por medio del Reporte de Decreto Especial.
decreto	Genera un SQL externo (decreto.sql), el cual se ejecutara por medio del Reporte de Decreto.
docfir	Genera un SQL externo (docfir.sql), el cual se ejecutara por medio del Reporte de Documentos para Firma.
doctur	Genera un SQL externo (doctur.sql), el cual se ejecutara por medio de la forma de Impresión de Documentos Tumados.
estadis	Genera SQL's externos (estadis1.sql, estadis2.sql, estadis3.sql), los cuales se ejecutan por medio del Reporte de Viajes.
gentem	Genera un SQL externo (gesgral.sql), el cual se ejecutara por medio del Reporte de Consultas Generales.
invita	Genera un SQL externo (invita.sql), el cual se ejecutara por medio del Reporte de Invita.
invtur	Genera un SQL externo (invtur.sql), el cual se ejecutara con la forma de Impresión de Doc. Tumados (Invitaciones).
obstur	Genera un SQL externo (obstur.sql), el cual se ejecutara por medio de la forma de Impresión de Doc. Tumados (Obsequios).
oficio	Genera un SQL externo (oficio.sql), el cual se ejecutara por medio del Reporte de oficio.pc
reldis	Genera un SQL externo (reldis.sql), el cual se ejecutara por medio de la forma de Consultas Especiales.
reporte	Genera un archivo GenRep.bat, para después ejecutarlo desde la forma.
tumo	Genera un SQL externo (tumo.sql), el cual se ejecutara por medio de la forma de Impresión de Tumos.
viaje	Genera un SQL externo (viaje.sql), el cual se ejecutara por medio del Reporte de Viajes.

Tabla A-T2 User-Exit.

### A.3 Listado de Reportes

SqlReport es una herramienta de Oracle que nos permite generar reportes de una manera sencilla. Debido a que Oracle no genera un programa fuente sino sólo el ejecutable, no es posible mostrar los listados.

La tabla A-T3 muestra los reportes creados, su nombre y su función:

<b>REPORTE</b>	<b>FUNCIÓN</b>
CATALF.RDF	Catálogo Alfabético Titulo dependencia
CATALF1.RDF	Catálogo Alfabético Titulo Remitente
CATALF2.RDF	Catálogo alfabético Titulo tema
CATALF3.RDF	Catálogo Alfabético Titulo referencia Alfabética
CATALF4.RDF	Catálogo Alfabético Titulo Turnado
CATTUR.RDF	Cat. Parcial de Turnados (Clave, Nom, Cargo)
CLASIFI.RDF	Clasificaciones
COMPLET1.RDF	Consultas Alfabéticas reporte Completo Matriz
COMPLETX.RDF	Consultas Alfabéticas reporte Completo Láser
CONSALF1.RDF	Consultas Alfabéticas
CONSESP.RDF	Consultas Especiales
CONSGRAL.RDF	Consultas Generales por pantalla
DECRET.RDF	Decreto Especial
DECRETO.RDF	Decreto
DEPENDI.RDF	Dependencias
DISCURSO.RDF	Discurso
DOCFIR.RDF	Documentos para firma
DOCTUR.RDF	Documentos Turnados
DOCTURAU.RDF	Documentos Turnados Audiencias
DOCTURES.RDF	Documentos Turnados con resolución
DOCTURIN.RDF	Documentos Turnados Invitaciones
DOCTUROB.RDF	Documentos Turnados Obsequios
DOCUMENT.RDF	Documentos
ESTADIS.RDF	Estadísticas
ETIQUET.RDF	Etiquetas de turnados 4 etiquetas
ETIQUET6.RDF	Etiquetas de Turnados (6 etiquetas)
EXPEDI.RDF	Expedientes
GCOMPLET.RDF	Consultas Generales Reporte Complemento Matriz
GCOMPLEX.RDF	Consultas Generales Reporte Complemento Láser
GNORMAL.RDF	Consultas Generales Reporte Normal Matriz
GNORMALX.RDF	Consultas Generales reporte Normal Láser.
GRALTUR.RDF	Catálogo general de Turnados todos los campos
GRESUMI1.RDF	Consultas Generales reporte resumido Matriz

Continúa...

REPORTE	FUNCIÓN
GRESUMIX.RDF	Consultas Generales Reporte resumido Láser
GSEGUI1.RDF	Consultas Generales seguimiento Matriz
GSEGUI1X.RDF	Consultas Generales reporte de Seguimiento Láser
IMPTUR.RDF	Impresión de Turnados
INSTRUCI.RDF	Instrucciones
INVITA.RDF	Invitaciones
NORMAL3.RDF	Consultas Alfabéticas reporte Normal Matriz
NORMAL3X.RDF	Consultas Alfabéticas reporte Normal Láser
OFICIO.RDF	Oficio
RELDIS.RDF	Relación de Discursos
RESUM11.RDF	Consultas Alfabéticas reporte Resumido Matriz
RESUM1X.RDF	Consultas Alfabéticas reporte Resumido Láser
SEGUI1.RDF	Consultas Alfabéticas reporte Seguimiento Matriz
SEGUI1X.RDF	Consultas Alfabéticas reporte Seguimiento Láser
TEMAI.RDF	Tema
VIAJE.RDF	Viaje

Tabla A-T3 Reportes.

### A.4 Ejemplo de una Forma

El siguiente listado corresponde únicamente a la forma TURGRALC.FMX que se ejecuta cuando se hace la consulta de catálogos de turnados. Se eligió esta forma, por que es una forma que se maneja diferente a los demás catálogos, debido a que las altas, bajas y cambios se manejan de manera más compleja.

Como es muy extenso el listado de esta forma, se edito, y solo se dejo las definiciones más importantes como nombre de la forma, nombre de los bloques, nombre de los campos, los triggers asociados a cada elemento y su código, y se le quito gran parte del código que genera Oracle que no es de mucho interés como atributos, características de despliegue, tipos de fonts, posición sobre la pantalla, etc.

```

*****
*****
***** TURGRAL.FMX : realiza las consultas de Turnados. Turnado es a *****
***** la persona a la cual se le asigna la resolución de un documento. *****
*****                               Genera 3 tipos de reportes: *****
*****                               -Reporte General. *****
*****                               -Reporte de Catálogos de turnados *****
*****                               -Reporte de Etiquetas. *****
*****
*****

```

Name TURGRALC

Triggers  
Name WHEN-TIMER-EXPIRED

```

Trigger Text
HINT.ShowButtonHelpHandler;
EXCEPTION
when zero_divide then
Message (ERROR_TYPE||' '||to_char(DBMS_ERROR_CODE)||'DBMS_ERROR_T
raise form_trigger_failure;

```

Name WHEN-MOUSE-UP

```

Trigger Text
HINT.HideButtonHelp;

```

Name WHEN-MOUSE-ENTER

```

Trigger Text
HINT.ShowButtonHelp;

```

Name WHEN-NEW-FORM-INSTANCE

```

Trigger Text
Set_Window_Property(FORMS_MDI_WINDOW,WINDOW_STATE,MAXIMIZE);
Set_Window_Property('Root_window',WINDOW_STATE,MAXIMIZE);

```

#### Blocks

Name DATOS

```

Name FECHA
Data Type Date
Maximum Length 9
Format Mask dd/mm/yy
Default Value $$DATE$$

```

```

Name HORA
Data Type Datetime
Maximum Length 7
Format Mask HH24:MI:SS
Default Value $$TIME$$

```

```

Name CVE_TUR
Item Type Text Item
Data Type Char
Maximum Length 45

```

```

Name NOMTUR
Item Type Text Item
Data Type Char
Maximum Length 45

```

```

Name CARGO
Item Type Text Item
Data Type Char
Maximum Length 45

```

```

Name AREA
Item Type Text Item
Data Type Char
Maximum Length 10

```

Triggers

Ejemplo de una forma

---

```

Name                                KEY-NEXT-ITEM
Trigger Text
DECLARE
i number;
cad1 char(80);
comando char(200);
BEGIN
IF :DATOS.NOMTUR IS Not NULL THEN
  cad1 := '%';
  for i in 1..length(:datos.nomtur) loop
    if substr(:datos.nomtur,i,1) <> ' ' then
      cad1:=cad1||substr(:datos.nomtur,i,1);
    else
      cad1:=cad1||'%';
    end if;
  end loop;
  :datos.nomtur := cad1||'%';
END IF;
IF :DATOS.CARGO IS Not NULL THEN
  cad1 := '%';
  for i in 1..length(:datos.cargo) loop
    if substr(:datos.cargo,i,1) <> ' ' then
      cad1:=cad1||substr(:datos.cargo,i,1);
    else
      cad1:=cad1||'%';
    end if;
  end loop;
  :datos.cargo := cad1||'%';
END IF;
message(:datos.cargo);
if :datos.nomtur is null then :datos.nomtur := '%';end if;
if :datos.cargo is null then :datos.cargo := '%';end if;
if :datos.area is null then :datos.area := '%';end if;
if :datos.cve_tur is null then :datos.cve_tur := '%';end if;
:datos.comando:=:DATOS.CVE_TUR||'|'
:DATOS.NOMTUR||'|'
:DATOS.CARGO||'|'
:DATOS.AREA;
user_exit('Reporte '|:datos.comando);
GO_FIELD('TIPO');
END;

```

Name	COMANDO
Item Type	Text Item
Data Type	Char
Maximum Length	200

Name	REPORTE
Item Type	Text Item
Data Type	Char
Maximum Length	20

Name	TIPO
Item Type	Text Item
Data Type	Char
Maximum Length	1

## Triggers

```

Name                                KEY-NEXT-ITEM
Trigger Text
DECLARE
nom1  char(80);
carg1 char(80);
list_id  ParamList;
Begin
IF :REPORTE.TIPO IN ('1','2','3') THEN
  IF :REPORTE.TIPO = '3' THEN
    GO_BLOCK('ETIQUET');
    CLEAR_BLOCK;
  END IF;
  IF :REPORTE.TIPO = '1' THEN
    GO_BLOCK('TITULO');
    CLEAR_BLOCK;
  END IF;
  IF :REPORTE.TIPO = '2' THEN
    list_id:=Get_Parameter_List('input_param');
    IF not id_null(list_id) THEN
      Destroy_Parameter_List(list_id);
    END IF;
    list_id := Create_Parameter_List('input_param');
    Add_Parameter(list_id,'PARAMFORM',TEXT_PARAMETER,'NO');
    run_product(reports,'f:gestion\vp\cattur',asynchronous,run
    GO_BLOCK('DATOS');
    CLEAR_BLOCK;
  END IF;
ELSE
  message('Elegir opcion 1, 2 o 3');
END IF;
END;

```

Name	ETIQUET
Name	NUMERO
Item Type	Text Item
Data Type	Char
Maximum Length	1

## Tiggers

```

Name                                KEY-NEXT-ITEM
Trigger Text
DECLARE
list_id ParamList;
BEGIN
IF :ETIQUET.NUMERO IN ('4','6') THEN
  list_id:=Get_Parameter_List('input_param');
  IF not id_null(list_id) THEN
    Destroy_Parameter_List(list_id);
  END IF;
  list_id := Create_Parameter_List('input_param');
  Add_Parameter(list_id,'PARAMFORM',TEXT_PARAMETER,'NO');
  IF :ETIQUET.NUMERO = '4' THEN
    run_product(reports,'f:gestion\vp\etiquet',asynchronous,r

```



```

ELSE
  run_product(reports,'f:\gestion\vpt\etiquet0',asynchronous,
END IF;
GO_BLOCK('DATOS');
CLEAR_BLOCK;
ELSE
  MESSAGE('Se debe de escoger 4 o 6');
END IF;
END;

```

Name TITULO

Name TITULO1  
Item Type Text Item  
Data Type Char  
Maximum Length 80

## Triggers

Name KEY-NEXT-ITEM

Trigger Text

DECLARE

i number;

cad1 char(80);

comando char(200);

list\_id ParamList;

BEGIN

IF :TITULO.TITULO1 IS NULL THEN

:TITULO.TITULO1:='Catalogo'||'%'||'General'||'%'||'de'||'%'

ELSE

FOR i in 1..length(:titulo.titulo1) LOOP

IF substr(:titulo.titulo1,i,1) <> '' THEN

cad1:=cad1||substr(:titulo.titulo1,i,1);

ELSE

cad1:=cad1||'%';

END IF;

END LOOP;

:TITULO.TITULO1 := cad1;

END IF;

list\_id:=Get\_Parameter\_List('input\_param');

IF not id\_null(list\_id) THEN

Destroy\_Parameter\_List(list\_id);

END IF;

list\_id := Create\_Parameter\_List('input\_param');

Add\_Parameter(list\_id,'TITULO',TEXT\_PARAMETER,:TITULO.TITU

Add\_Parameter(list\_id,'PARAMFORM',TEXT\_PARAMETER,'NO');

run\_product(reports,'f:\gestion\vpt\grafur',asynchronous,

END;

GO\_BLOCK('DATOS');

CLEAR\_BLOCK;

Name ICONOS

Name LIMPIAR\_FORMA  
Item Type Button  
Icon Name Clear

Default Button	False
Hint	Limpia todos los campos de
Triggers	
Name	WHEN-MOUSE-CLICK
Trigger Text	do_key('Clear_Form');
Name	SALIDA_TEMPORAL
Item Type	Button
Icon Name	Exit
Default Button	False
Hint	Salida Temporal MS-DOS
Triggers	
Name	WHEN-MOUSE-CLICK
Trigger Text	Host('C:\dos\command.com');
Name	SALIDA
Item Type	Button
Icon Name	menu
Default Button	False
Hint	Menú Anterior
Triggers	
Name	WHEN-MOUSE-CLICK
Trigger Text	do_key('Exit_Form');
Name	WORD
Item Type	Button
Icon Name	chkbook
Default Button	False
Hint	Activa Word For Windows
Triggers	
Name	WHEN-MOUSE-CLICK
Trigger Text	DECLARE IArgs OLE2.LIST_Type; BEGIN word.hApp := OLE2.CREATE_OBJ('Word.Basic'); IArgs := OLE2.CREATE_ARGLIST; OLE2.ADD_ARG(IArgs,'Doc1.doc'); OLE2.INVOKE(Word.hApp,'fileopen',IArgs); OLE2.DESTROY_ARGLIST(IArgs); END;
Relations	
-Views	
Name	PAGE_1
Window	PAGE_1
Name	PAGE_3

Ejemplo de una forma

---

Window	PAGE_3
Name	PAGE_4
Window	PAGE_4
Name	PAGE_5
Window	PAGE_5
Name	PAGE_6
Window	ROOT_WINDOW
Name	PAGE_7
Window	PAGE_7
Name	PAGE_8
Window	ROOT_WINDOW

Program Units  
WORD (Package Spec)  
PACKAGE word IS  
hApp Ole2.Obj\_Type;  
END;

Properties  
Record Groups

Name	IMPRESION_NAME_LOV0
Record Group Query	select name,descrip from imp
Column Specification	
NAME	Char 10
DESCRIP	Char 30

Visual Attributes

Windows

Name	PAGE_2
Window Style	Dialog
Name	PAGE_3
Window Style	Dialog
Name	PAGE_4
Window Style	Dialog

Name	PAGE_5
Window Style	Dialog
Name	PAGE_7
Window Style	Dialog
Name	ROOT_WINDOW
Title	Catálogo de Turnos
Window Style	Document

## A.5 Listado de 16 programas user exit

```

.....
.....
***** NAME *****
***** catalf.pc - User Exit GENERACION de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL(temalf), para despues generar un archivo *****
***** catalf.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio del reporte de viajes. *****
.....
...../

#ifdef RCSID
    static char *RCSid =
        "$Header: /forms41/prd/RCS/catalf.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro'C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int catalf(); /* User exit */
FILE *fp;

/** viajesql -- Funcion que genera un archivo query externo */
int catalfaq(void)
int mape[2], fig_and;
char aster[5] = " ";
char cadenot[60], cadenit[60];
char cade1[60], cade2[60], cade3[60];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if(((fp = fopen("\\GESTION\\TEMP\\temalf", "r"))==NULL)
    return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';

```

```

fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 2; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 2){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1, cadenit); break;
        case 1: strcpy(cade2, cadenit); break;
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\catalif.sql", "w")) == NULL)
    return(FAIL);
fprintf(fp, "select id, numdoc, clasif, ref, fecdoc, fecrec, fecdesp, \n");
fprintf(fp, "fecdesc, feceven, fecarch, dependencia, asunto, horaeven from gesgral \n");
fprintf(fp, "where tipdoc='G' \n");
fig_and=0;
if(i > 0)
    fprintf(fp, "and \n");
for(i = 0; i < 2; i++){
    if(mapa[i] == 1){
        switch(i){
            case 0:
                fprintf(fp, "id between %s and %s ", cade1, cade2);
                fig_and = 1;
            case 1: break;
        }
    }
}
fprintf(fp, "\n order by id ");
fclose(fp);
return(OK);
}
/**----- viaje -----**
** catalif -- User eXit Genera un archivo temporal y general un Query Extern
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int catalif()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral, 0, sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\wmat", "w")) == NULL)
        return(FAIL);
    strcpy(fld_gral, "ofgest.COMANDO");
    fld_gral.len = strlen("ofgest.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i] != '\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp, "%s\n", fld_buf_gral);
    fclose(fp);
    if (catalifsql() == 0) return(FAIL); /* Genera el Query externo para el reporte */
    return (OK);
}

```

```

/.....
***** NAME *****
***** ConsEsp.pc - User Exit de CONSULTA ESPECIAL archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL, para despues generar un archivo *****
***** CONESP.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Consultas Especiales. *****
/.....
/.....
#ifdef RCSID
static char *RCSid =
"$Header: //forms41/prd/RCS/ConsEsp.pc,v 1.3 1995/05/18 20:30:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro*C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.any[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int consesp();
FILE *fp;

/** GENSQLUNO -- Funcion que genera un archivo query externo */
int gensqluno(void){
int mapa[27],fig_and;
char cadenot[800],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if(((fp = fopen("\\GESTION\\TEMP\\TempCEsp","r"))==NULL)
return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 7; i++)
mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 6){
p2 = cadenit;
while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strlen(cadenit,"-1")!=0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1,cadenit); break;
case 1: strcpy(cade2,cadenit); break;
case 2: strcpy(cade3,cadenit); break;
case 3: strcpy(cade4,cadenit); break;
case 4: strcpy(cade5,cadenit); break;
case 5: strcpy(cade6,cadenit); break;
}
i++;
}
}

```

```

}
if((fp = fopen("\\GESTION\\SQL\\CONSESP.sql","w"))==NULL)
    return(FAIL);
fprintf(fp,"select id,tipdoc,numdoc,dependencia,clasif,ref,feccdoc,\n");
fprintf(fp,"fecrec,fecdesp,fecdesc,feceven,fecarch,horeven,asunto,\n");
fprintf(fp,"ant,sub,digitalizado from gesgral \n");
fig_and=0;
if(i > 0)
    fprintf(fp,"where \n");
for(i = 0; i < 6; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                if(fig_and == 1) fprintf(fp," \n and ");
                fprintf(fp,"tipdoc = '%s' ",cade1);
                fig_and = 1;
                break;
            case 1:
                if(fig_and == 1) fprintf(fp," \n and ");
                fprintf(fp,"id between %s and %s ",cade2,cade3);
                fig_and = 1;
                break;
            case 2: break;
            case 3:
                if(fig_and == 1) fprintf(fp," \n and ");
                if(strchr(cade4,',')!=NULL)
                    cade4[strlen(cade4)-1] = '\0';
                p = cade4; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"clasif in ('%s') ",cadenit);
                fig_and = 1;
                break;
            case 4: if(fig_and == 1) fprintf(fp," \n and ");
                if (strchr(cade5,',')!=NULL)
                    cade5[strlen(cade5)-1] = '\0';
                p = cade5; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"id in(select id from gesturno where upper(turno) in ('%s')",cadenit);
                if (mapa[5] != 1)
                    fprintf(fp,"");
                else fig_and = 1;
                break;
            case 5: if(fig_and == 1) fprintf(fp," \n and ");
                if (strchr(cade6,',')!=NULL)
                    cade6[strlen(cade6)-1] = '\0';
                p = cade6; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                if (strchr(cade6,'')==NULL)

```

```

        fprintf(fp," upper(instruc) in (%s\%) " ,cadena);
    else
        fprintf(fp," upper(instruc) <> 'L' ");
    break;
    }
}
fprintf(fp,"n order by numdoc");
fclose(fp);
return(OK);
}

```

```

/*----- CONSESP -----*/

```

```

** CONSESP - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise. **/
int consesp( )
{

```

```

    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /* ** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\TempCEsp", "w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,arr,"FECHA.COMANDO");
    fld_gral.len = strlen("FECHA.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    l=499;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"====s====n",fld_buf_gral);
    fclose(fp);
    if (gensqluno()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

.....
***** NAME *****
***** decret.pc - User Exit GENERacion de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL (temdec2), para despues generar un archivo *****
***** SQL (decret.sql), generando asi un SQL externo, el cual se ejecutara *****
***** por medio del reporte de decreto especial *****
.....
.....

```

```

#ifdef RCSID
    static char *RCSId =
        "$Header: //forms41/prd/RCS/decret.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro*C defs **/

```



## A.5 Listado de 16 programas user exit

```

#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int decret(); /* User exit*/
FILE *fp;

/** dec2sql -- Funcion que genera un archivo query externo ***/
int dec2sql(void)
int mapa[7],fig_and;
char aster[6] = " ";
char cadenot[600],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80],cade7[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\temdec2","r"))==NULL)
    return(FAIL);
fgets(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 6; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 6){
    p2 = cadenit;
    while(*p != '|' && *p *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit,"-1")!=0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1,cadenit); break;
        case 1: strcpy(cade2,cadenit); break;
        case 2: strcpy(cade3,cadenit); break;
        case 3: strcpy(cade4,cadenit); break;
        case 4: strcpy(cade5,cadenit); break;
        case 5: strcpy(cade6,cadenit); break;
        case 6: strcpy(cade7,cadenit); break;
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\decret.sql","w"))==NULL)
    return(FAIL);
fprintf(fp,"select id,tipdoc,numdoc,clasif,ref, \n");
fprintf(fp,"asunto from gesgral where tipdoc='D' \n");
fig_and=1;
for(i = 0; i < 7; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                if(fig_and == 1) fprintf(fp," \n and ");
                fprintf(fp,"numdoc between %s and %s ",cade1,cade2);
                fig_and = 1;
            case 1: break;
            case 2:
                if(fig_and == 1) fprintf(fp," \n and ");
                fprintf(fp,"fecrec >= to_date('%s','\dd-mm-yy') and fecrec <= to_date('%s','\dd-mm-yy')
",cade3,cade4);
                fig_and = 1;
            case 3: break;
            case 4:
                if(fig_and == 1) fprintf(fp," \n and ");
                if(strchr(cade5,',')!=NULL)
                    cade5[strlen(cade5)-1] = '\0';
                p = cade5; p2 = cadenit;

```

```

        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ' ';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        fprintf(fp, "id in (select id from gesturno where upper(turno) in ('%s'))", cadentit);
        flg_and = 1;
        break;

    case 5:
        if((flg_and == 1) fprintf(fp, "\n and ");
        if(strchr(cade6, ',') != NULL)
            cade6[strlen(cade6)-1] = '\0';
        p = cade6; p2 = cadentit;
        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ' ';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        fprintf(fp, "clasi in ('%s')", cadentit);
        flg_and = 1;
        break;

    case 6:
        if((flg_and == 1) fprintf(fp, "\n and ");
        if(strchr(cade7, ',') != NULL)
            cade7[strlen(cade7)-1] = '\0';
        p = cade7; p2 = cadentit;
        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ' ';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        if (cadentit[0] == "")
            fprintf(fp, "id in (select id from gesturno where upper(instruc) not in ('LV'))");
        else
            fprintf(fp, "id in (select id from gesturno where upper(instruc) in ('%s'))", cadentit);
        flg_and = 1;
        break;
    }
}

fprintf(fp, "\n order by numdoc ");
fclose(fp);
return(OK);
}

/*----- decret -----*/
**decret -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.*/
int decret()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
    char fld_buf_gral[500]; /* Valor del campo del Buffer */
    VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;

    EXEC SQL INCLUDE SQLCA;

```

```

/** El argumento es la cadena con los parametros */
memset(fid_buf_gral,0,sizeof(fid_buf_gral));
if((fp = fopen("\\GESTION\\TEMP\\temdec2","w"))==NULL)
    return( FAIL );
strcpy(fid_gral.arr,"ofgest.comando");
fid_gral.len = strlen("ofgest.comando");
EXEC TOOLS GET :fid_gral INTO :fid_buf_gral;
i=499;
while(fid_buf_gral[i]!='\0') i--;
fid_buf_gral[i+1] = '\0';
fprintf(fp,"%s\n",fid_buf_gral);
fclose(fp);
if (dec2sql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
return ( OK );
}

```

```

.....
.....
***** NAME *****
***** decreto.pc - User Exit GENERacion de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL (temdec1), para despues generar un archivo *****
***** SQL (decreto sql), generando asi un SQL externo, el cual se ejecutara *****
***** por medio del reporte de decreto *****
.....
.....

```

```

#ifdef RCSID
static char *RCSid =
"$Header: /forms41/prd/RCS/decreto.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $ ";
#endif /* RCSID */

```

```

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

```

```

/** Some useful Pro*C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */
EXEC SQL INCLUDE sqlca.h;
extern int decreto(); /* User exit */
FILE *fp;

```

```

/** dec1sql - Funcion que genera un archivo query externo */
int dec1sql(void){
int mapa[7],fig_and;
char aster[6] = " ";
char cadenot[600],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80],cade7[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\temdec1","r"))==NULL)
    return( FAIL );
fgets(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 7; i++)
    mapa[i] = 0;

```

```

p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 7){
    p2 = cadenit;
    while('p != ' && 'p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1, cadenit); break;
        case 1: strcpy(cade2, cadenit); break;
        case 2: strcpy(cade3, cadenit); break;
        case 3: strcpy(cade4, cadenit); break;
        case 4: strcpy(cade5, cadenit); break;
        case 5: strcpy(cade6, cadenit); break;
    case 6: strcpy(cade7, cadenit); break;
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\decreto.sql", "w")) == NULL)
    return(FAIL);
fprintf(fp, "select id, tipdoc, numdoc, fecdoc, fecrec, dependencia, \n");
fprintf(fp, "asunto, fecdeap, fecdeac, ref, ant, sub \n");
fprintf(fp, "from gesgral where tipdoc='D' \n");
fig_and=1;
for(i = 0; i < 7; i++){
    if(mapa[i] == 1){
        switch(i){
            case 0:
                if(fig_and == 1) fprintf(fp, "\n and ");
                fprintf(fp, "numdoc between %s and %s ", cade1, cade2);
                fig_and = 1;
            case 1: break;
            case 2:
                if(fig_and == 1) fprintf(fp, "\n and ");
                fprintf(fp, "fecrec >= to_date('%s', '\dd-mm-yy') and fecrec <= to_date('%s', '\dd-mm-yy')
", cade3, cade4);
                fig_and = 1;
            case 3: break;
            case 4:
                if(fig_and == 1) fprintf(fp, "\n and ");
                if( strchr(cade5, ',') != NULL)
                    cade5[strlen(cade5)-1] = '\0';
                p = cade5; p2 = cadenit;
                while('p){
                    if('p == ',){
                        *p2++ = '\n'; *p2++ = ',';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp, "id in (select id from gesturno where upper(turno) in ('%s'))", cadenit);
                fig_and = 1;
                break;
            case 5:
                if(fig_and == 1) fprintf(fp, "\n and ");
                if( strchr(cade6, ',') != NULL)
                    cade6[strlen(cade6)-1] = '\0';
                p = cade6; p2 = cadenit;
                while('p){
                    if('p == ',){
                        *p2++ = '\n'; *p2++ = ',';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp, "classif in ('%s')", cadenit);
                fig_and = 1;

```

```

        break;
    case 6:
        if((fig_and == 1) fprintf(fp,"n and ");
        if(strchr(cade7, ',') != NULL)
            cade7[strlen(cade7)-1] = '\0';
        p = cade7; p2 = cadentit;
        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ',';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        if (cadentit[0] != "")
            fprintf(fp,"id in (select id from gesturno where upper(instruc) not in ('L'))");
        else
            fprintf(fp,"id in (select id from gesturno where upper(instruc) in ('%s'))", cadentit);
        fig_and = 1;
        break;
    }
}

fprintf(fp,"n order by numdoc ");
fclose(fp);
return(OK);
}

/*----- decret -----
**decreto - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/

int decreto()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
    char fld_buf_gral[500]; /* Valor del campo del Buffer */
    VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /* El argumento es la cadena con los parametros */
    memset(fld_buf_gral, 0, sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\temdect1", "w")) == NULL)
        return(FAIL);
    strcpy(fld_gral.arr, "ofgest.comando");
    fld_gral.len = strlen("ofgest.comando");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i] != '\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp, "%s\n", fld_buf_gral);
    fclose(fp);
    if (dec1sql() != 0) return(FAIL); /* Genera el Query externo para el reporte */
    return(OK);
}

```

```

.....
.....
***** NAME .....
***** docfir.pc - User Exit GENERACION de archivo TEMORAL .....
***** DESCRIPCION .....
***** Genera un archivo TEMPORAL (temdoc), para despues generar un archivo .....
***** SQL (docfir.sql), generando asi un SQL externo, el cual se ejecutara .....
***** por medio del reporte de documentos para firma .....
.....
.....
#ifdef RCSID
    static char *RCSid =
        "$Header: /forms41/prd/RCS/docfir.pc.v 1.3 1995/05/03 18:09:01 MLLC Exp $ ";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro*C defs **/
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int docfir(); /* User exit*/
FILE *fp;
/** docsq1 - Funcion que genera un archivo query externo **/
int docsq(void)
{
    int mapa[6], fig_and;
    char aster[6] = " ";
    char cadenot[600], cadenit[80];
    char cade1[80], cade2[80], cade3[80], cade4[80], cade5[80], cade6[80], cade7[80];
    int i;
    char *p,*p2;

    /*Abre el archivo temporal*/
    if(((fp = fopen("\\GESTION\\TEMP\\temdoc","r"))==NULL)
        return( FAIL );
    (gets(cadenot,sizeof(cadenot),fp);
    cadenot[strlen(cadenot)-1] = '\0';
    fclose(fp);
    /* Limpia la variable mapa */
    for(i = 0; i < 6; i++)
        mapa[i] = 0;
    p = cadenot;
    i = 0;
    /* Separa las variables e identifica cual se valida */
    while(i < 7){
        p2 = cadenit;
        while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
        if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
        switch(i){
            case 0: strcpy(cade1, cadenit); break;
            case 1: strcpy(cade2, cadenit); break;
            case 2: strcpy(cade3, cadenit); break;
            case 3: strcpy(cade4, cadenit); break;
            case 4: strcpy(cade5, cadenit); break;
            case 5: strcpy(cade6, cadenit); break;
            case 6: strcpy(cade7, cadenit); break;
        }
        i++;
    }
}

```

## A.5 Listado de 16 programas user exit

```

}
if((fp = fopen("\\GESTION\\SQL\\docfir.sql","w"))==NULL)
    return(FAIL);
fprintf(fp,"select id,numdoc,feccdoc,feccrec,dependencia, \n");
fprintf(fp,"asunto,feccesp,ref.ant,sub \n");
fprintf(fp,"from gesgral where tipdoc='F' \n");
fig_and=1;
for(i = 0; i < 7; i++){
    if(mape[i]==1){
        switch(i){
            case 0:
                if(fig_and == 1) fprintf(fp,"n and ");
                fprintf(fp,"numdoc between %s and %s ",cade1,cade2);
                fig_and = 1;
            case 1: break;
            case 2:
                if(fig_and == 1) fprintf(fp,"n and ");
                fprintf(fp,"feccrec >= to_date('%s',\dd-mm-yy') and feccrec <= to_date('%s',\dd-mm-yy')
",cade3,cade4);
                fig_and = 1;
            case 3: break;
            case 4:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade5,',')!=NULL)
                    cade5[strlen(cade5)-1] = '\0';
                p = cade5; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"id in (select id from gesturno where upper(turno) in ('%s'))",cadenit);
                fig_and = 1;
                break;
            case 5:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade6,',')!=NULL)
                    cade6[strlen(cade6)-1] = '\0';
                p = cade6; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"classif in ('%s')",cadenit);
                fig_and = 1;
                break;
            case 6:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade7,',')!=NULL)
                    cade7[strlen(cade7)-1] = '\0';
                p = cade7; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                if (cadenit[0]!='')
                    fprintf(fp,"id in (select id from gesturno where upper(instruc) not in ('%s'))");
                else

```

```

        fprintf(fp,"id in (select id from gesturno where upper(instruc) in ('%s'))",cadenit);
        flq_and = 1;
        break;
    }
}
fprintf(fp,"n order by numdoc ");
fclose(fp);
return(OK);
}
/*----- viaje -----*/
**docfir - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int docfir()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\verdoc","w"))==NULL)
        return(FAIL);
    strcpy(fld_gral,arr,"ofgest.comando");
    fld_gral.len = strlen("ofgest.comando");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"""%s""n",fld_buf_gral);
    fclose(fp);
    if (docsql()==0) return(FAIL); /* Genera el Query externo para el reporte */
    return (OK);
}

.....
***** NAME *****
***** DocTur.pc - User Exit de Impresion de Documentos Turnados *****
***** DESCRIPCION *****
***** Genera un archivo DOCTUR, para despues generar un archivo *****
***** DOCTUR.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Impresion de Documentos Turnados. *****
.....
#ifndef RCSID
static char *RCSid =
"$Header: /forms41/prd/RCS/DocTur.pc,v 1.3 1995/05/23 18:20:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro*C defs **/
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#endif _WINDLL

```



```

#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int doctur();
FILE *fp;
/*** doctsql - Funcion que genera un archivo query externo ***/
int doctsql(void)
int mapa[27],fig_and;
char cadenot[790],cadenit[790];
char cade1[790];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\DocTur.Txt","r"))==NULL)
    return( FAIL );
fgets(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 2; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 1){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit,"-1")==0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1,cadenit); break; /* Numdoc. */
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\DocTur.SQL","w"))==NULL)
    return(FAIL);
fprintf(fp,"select id,numdoc,feodoc,dependencia,ref,asunto\n");
fprintf(fp,"from gesgral\n");
fprintf(fp,"where tipdoc='G' and ");
fig_and=1;
for(i = 0; i < 1; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                if(strchr(cade1,',')!=NULL)
                    cade1[strlen(cade1)-1] = '\0';
                p = cade1; p2 = cadenit;
                while(*p){
                    if(*p == '|'){
                        *p2++ = '|'; *p2++ = '\0';
                        *p2++ = '\0'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp," numdoc in ('%s')",cade1);
                fig_and = 1;
                break;
        }
    }
}
fprintf(fp,"\n order by numdoc");
fclose(fp);
return(OK);
}
/*----- DOCTUR -----
** DOCTUR - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,

```

```

** FAILURE otherwise. **/
int doctur( )
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fid_buf_gral[800]; /* Valor del campo del Buffer */
        VARCHAR fid_gral[800]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memset(fid_buf_gral,0,sizeof(fid_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\DocTur.Txt","w"))==NULL)
        return( FAIL );
    strcpy(fid_gral.arr,"OFGEST.CADENA1");
    fid_gral.len = strlen("OFGEST.CADENA1");
    EXEC TOOLS GET :fid_gral INTO :fid_buf_gral;
    i=799;
    while(fid_buf_gral[i]!='\0') i--;
    fid_buf_gral[i+1] = '\0';
    fprintf(fp,"%s\n",fid_buf_gral);
    fclose(fp);
    if (doctsql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

/.....
.....
***** NAME .....
***** estadis.pc - User Exit Generacion de archivo TEMoral .....
***** DESCRIPCION .....
***** Genera un archivo TEMPORAL(tamest), para despues generar archivos .....
***** SQL(estadis1.sql,estadie2.sql,estadie3.sql), generando asi un SQL externo, el cual se .....
***** ejecutado por medio del reporte de viajes. ....
.....
/

```

```

#ifdef RCSID
    static char *RCSid =
        "$Header: /forms41/prd/RCS/estadis.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1

/** Some useful Pro*C defs **/
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

```

```

EXEC SQL INCLUDE sqlca.h;
extern int estadis(); /* User exit */
FILE *fp,*fp2,*fp3;
/** estsql - Funcion que genera un archivo query externo **/
int estsql(void){
    int mape[4],fig_end;
    char aster[6] = " ";
    char cadenot[600],cadenit[80];
    char cade1[80],cade2[80],cade3[80],cade4[80];
    int i;

```

## A.5 Listado de 16 programas user exit

```
char *p,*p2;
/* Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\temest", "r"))==NULL)
    return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 4; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 4){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit, "-1")!=0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1, cadenit); break;
        case 1: strcpy(cade2, cadenit); break;
        case 2: strcpy(cade3, cadenit); break;
        case 3: strcpy(cade4, cadenit); break;
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\estad1.sql", "w"))==NULL)
    return(FAIL);
if((fp2 = fopen("\\GESTION\\SQL\\estad2.sql", "w"))==NULL)
    return(FAIL);
if((fp3 = fopen("\\GESTION\\SQL\\estad3.sql", "w"))==NULL)
    return(FAIL);
fprintf(fp, "select area,turmo,count(*) from turmo,gesturmo \n");
fprintf(fp, "where turmo.cve_tur=gesturmo.turmo and Instruc <> 'L' \n");
fprintf(fp2, "select area,turmo,count(*) from turmo,gesturmo \n");
fprintf(fp2, "where turmo.cve_tur=gesturmo.turmo \n");
fprintf(fp3, "select area,turmo,count(*) from turmo,gesturmo \n");
fprintf(fp3, "where turmo.cve_tur=gesturmo.turmo and Instruc = 'L' \n");
fig_and=0;
if(i > 0)
    fprintf(fp, "and \n");
    fprintf(fp2, "and \n");
    fprintf(fp3, "and \n");
for(i = 0; i < 4; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                fprintf(fp, "id between %s and %s ", cade1, cade2);
                fprintf(fp2, "id between %s and %s ", cade1, cade2);
                fprintf(fp3, "id between %s and %s ", cade1, cade2);
                fig_and = 1;
            case 1: break;
            case 2:
                if(fig_and == 1)
                {
                    fprintf(fp, "\n and ");
                    fprintf(fp2, "\n and ");
                    fprintf(fp3, "\n and ");
                }
                if(strchr(cade3, ',')==NULL)
                    cade3[strlen(cade3)-1] = '\0';
                p = cade3; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
            }
        }
    }
}
```

```

        fprintf(fp,"id in (select id from gesgral where tipdoc in ('%s'))",cadenit);
        fprintf(fp2,"id in (select id from gesgral where tipdoc in ('%s'))",cadenit);
        fprintf(fp3,"id in (select id from gesgral where tipdoc in ('%s'))",cadenit);
        flg_and = 1;
        break;

    case 3:
        if(flg_and == 1)
        {
            fprintf(fp,"n and ");
            fprintf(fp2,"n and ");
            fprintf(fp3,"n and ");
        }
        if(strchr(cade4,'.')!=NULL)
            cade4[strlen(cade4)-1] = '\0';
        p = cade4; p2 = cadenit;
        while(*p){
            if(*p == '.'){
                *p2++ = '\n'; *p2++ = '\n';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        fprintf(fp,"id in (select id from gesgral where clasif in ('%s'))",cadenit);
        fprintf(fp2,"id in (select id from gesgral where clasif in ('%s'))",cadenit);
        fprintf(fp3,"id in (select id from gesgral where clasif in ('%s'))",cadenit);
        flg_and = 1;
        break;
    }
}

fprintf(fp,"n group by area,turno ");
fprintf(fp2,"n group by area,turno ");
fprintf(fp3,"n group by area,turno ");
fclose(fp);
fclose(fp2);
fclose(fp3);
return(OK);
}

/*----- estadis -----*/
** estadis -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int estadis()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\temest","w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,err,"ofgest.COMANDO");
    fld_gral.len = strlen("ofgest.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\n') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"%s\n",fld_buf_gral);
    fclose(fp);
    if(estaql!=0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

/.....
.....
***** NAME *****
***** gentem.pc - User Exit GENERacion de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL, para despues generar un archivo *****
***** GESGRAL. SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio del reporte de Consultas Generales. *****
/...../

#ifdef RCSID
static char *RCSid =
"$Header: //forms41/prd/RCS/gentem.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/** Some useful Pro*C defs*/
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int gentem();
FILE *fp;
/** GENSQL - Funcion que genera un archivo query externo ***/
int gensql(void)
int mapa[27], flg_and;
char aster[6] = " ";
char cadenot[600], cadenit[80];
char cade1[80], cade2[80], cade3[80], cade4[80], cade5[80], cade6[80], cade7[80],
cade8[80], cade9[80], cade10[80], cade11[80], cade12[80], cade13[80], cade14[80],
cade15[80], cade16[80], cade17[80], cade18[80], cade19[80], cade20[80], cade21[80],
cade22[80], cade23[80], cade24[80], cade25[80], cade26[80], cade27[80];
int i;
char *p,*p2;
/* Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\temporal", "r"))==NULL)
return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i=0; i < 27; i++)
mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 27){
p2 = cadenit;
while(*p != '\0' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit, "-1")!=0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1, cadenit); break;
case 1: strcpy(cade2, cadenit); break;
case 2: strcpy(cade3, cadenit); break;
case 3: strcpy(cade4, cadenit); break;
case 4: strcpy(cade5, cadenit); break;
case 5: strcpy(cade6, cadenit); break;
case 6: strcpy(cade7, cadenit); break;
}
}

```

```

case 7: strcpy(cade8,cadenit); break;
case 8: strcpy(cade9,cadenit); break;
case 9: strcpy(cade10,cadenit); break;
case 10: strcpy(cade11,cadenit); break;
case 11: strcpy(cade12,cadenit); break;
case 12: strcpy(cade13,cadenit); break;
case 13: strcpy(cade14,cadenit); break;
case 14: strcpy(cade15,cadenit); break;
case 15: strcpy(cade16,cadenit); break;
case 16: strcpy(cade17,cadenit); break;
case 17: strcpy(cade18,cadenit); break;
case 18: strcpy(cade19,cadenit); break;
case 19: strcpy(cade20,cadenit); break;
case 20: strcpy(cade21,cadenit); break;
case 21: strcpy(cade22,cadenit); break;
case 22: strcpy(cade23,cadenit); break;
case 23: strcpy(cade24,cadenit); break;
case 24: strcpy(cade25,cadenit); break;
case 25: strcpy(cade26,cadenit); break;
case 26: strcpy(cade27,cadenit);
}
i++;
}
if(((fp = fopen("\\GESTION\\SQL\\gesgral.sql","w"))==NULL)
return(FAIL);
fprintf(fp,"select id,tipdoc,numdoc,dependencia,clasif,ref,feccod,\n");
fprintf(fp,"fecrec,feccdep,feccdesc,feceven,fecarch,horeven,asunto,\n");
fprintf(fp,"ant,sub,digitalizado from gesgral \n");
fig_and=0;
if(i > 0)
fprintf(fp,"where \n");
for(i = 0; i < 27; i++){
if(mapa[i]==1){
switch(i){
case 0:
fprintf(fp,"id between %s and %s ",cade1,cade2);
fig_and = 1;
case 1: break;
case 2:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"tipdoc = '%s'\ ",cade3);
fig_and = 1;
break;
case 3:
if((fig_and == 1) fprintf(fp,"n and ");
if(strlen(cade4) != NULL)
cade4[strlen(cade4)-1] = '\0';
p = cade4; p2 = cadenit;
while(*p){
if(*p == ','){
*p2++ = '\n'; *p2++ = '!';
*p2++ = '\n'; p++;
}
else
*p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp,"classif in ('%s') ",cadenit);
fig_and = 1;
break;
case 4:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"numdoc between %s and %s ",cade5,cade6);
fig_and = 1;
case 5: break;
case 6:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"upper(ref) like upper('%s') ",cade7);
fig_and = 1;
break;

```

## A.5 Listado de 16 programas user exit

---

```
case 7:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"fecdoc >= to_date('%s',\dd-mon-yy') and fecdoc <= to_date('%s',\dd-mon-yy')
",cade8,cade9);
    flg_and = 1;
case 8: break;
case 9:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"fecrec >= to_date('%s',\dd-mon-yy') and fecrec <= to_date('%s',\dd-mon-yy')
",cade10,cade11);
    flg_and = 1;
case 10: break;
case 11:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"fecdesp >= to_date('%s',\dd-mon-yy') and fecdesp <= to_date('%s',\dd-mon-
yy') ",cade12,cade13);
    flg_and = 1;
case 12: break;
case 13:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"fecdesc >= to_date('%s',\dd-mon-yy') and fecdesc <= to_date('%s',\dd-mon-yy')
",cade14,cade15);
    flg_and = 1;
case 14: break;
case 15:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"feceven >= to_date('%s',\dd-mon-yy') and feceven <= to_date('%s',\dd-mon-yy')
",cade16,cade17);
    flg_and = 1;
case 16: break;
case 17:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"fecarch >= to_date('%s',\dd-mon-yy') and fecarch <= to_date('%s',\dd-mon-yy')
",cade18,cade19);
    flg_and = 1;
case 18: break;
case 19:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"horeven = '%s' ",cade20);
    flg_and = 1;
    break;
case 20:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"upper(dependencia) like upper('%s') ",cade21);
    flg_and = 1;
    break;
case 21:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"id in (select id from gesrem where upper(remite) like upper('%s') ) ",cade22);
    flg_and = 1;
    break;
case 22:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"id in (select id from gesrefal where upper(refal) like upper('%s') ) ",cade23);
    flg_and = 1;
    break;
case 23:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"id in (select id from gestem where upper(bema) like upper('%s') ) ",cade24);
    flg_and = 1;
    break;
case 24:
    if((flg_and == 1) fprintf(fp,"n and ");
    fprintf(fp,"id in (select id from gesexp where upper(expediente) like upper('%s') ) ",cade25);
    flg_and = 1;
    break;
case 25:
    if((flg_and == 1) fprintf(fp,"n and ");
    if(strchr(cade26,',')!=NULL)
```

```

        cade26[strlen(cade26)-1] = '\0';
        if(strchr(cade26, '*') != NULL){
            fprintf(fp, "id in (select id from gesturno) ");
        }else{
            p = cade26; p2 = cadenit;
            while(*p){
                if(*p == ','){
                    *p2++ = '\n'; *p2++ = ',';
                    *p2++ = '\n'; p++;
                }else
                    *p2++ = *p++;
            }
            *p2++ = '\0';
            fprintf(fp, "id in (select id from gesturno where upper(turno) in ('%s\') ", cadenit);
        }
        if((mapa[26] == 1){
            fprintf(fp, "\n and ");
            strcpy(aster, cade27);
            if(aster[0] == "")
                fprintf(fp, "upper(instruc) not in ('L') ");
            else
                fprintf(fp, "upper(instruc) like upper('%s\') ", cade27);
        }else{
            fprintf(fp, " ");
        }
        fig_and = 1;
        break;
    case 26:
        if((mapa[25] == 1) break;
        if((fig_and == 1) fprintf(fp, "\n and ");
        strcpy(aster, cade27);
        if(aster[0] == "")
            fprintf(fp, "id in (select id from gesturno where upper(instruc) not in ('L') ");
        else
            fprintf(fp, "id in (select id from gesturno where upper(instruc) like upper('%s\') ", cade27);
        fig_and = 1;
        break;
    }
}
fprintf(fp, "\n order by id ");
fclose(fp);
return(OK);
}

```

```

/*----- GENTEM -----
** GENTEM -- User exit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.*/
int gentem()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral, 0, sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\temporal", "w")) == NULL)
        return( FAIL );
    strcpy(fld_gral, "BLOQUE1.COMANDO");
    fld_gral.len = strlen("BLOQUE1.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=400;
    while(fld_buf_gral[i] == ' ') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp, "%s\n", fld_buf_gral);
}

```



```

fclose(fp);
if (gensql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
return ( OK );
}

```

```

.....
.....
***** NAME .....
***** invita.pc - User Exit GENERACION de archivo TEMPoral .....
***** DESCRIPCION .....
***** Genera un archivo TEMPORAL (teminv), para despues generar un archivo .....
***** SQL (invita.sql), generando asi un SQL externo, el cual se ejecutara .....
***** por medio del reporte de invita. ....
.....
.....

```

```

#ifdef RCSID
static char *RCSid =
"$Header: /forms41/prd/RCS/invita.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/* Some useful Pro'C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

```

```

EXEC SQL INCLUDE sqlca.h;
extern int invita(); /* User exit */
FILE *fp;
/* Invsql - Funcion que genera un archivo query externo */
int invsql(void){
int mape[11],fg_and;
char aster[6] = " ";
char cadenot[800],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80],cade7[80],
cade8[80],cade9[80],cade10[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\teminv.txt","r"))==NULL)
return( FAIL );
fgets(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mape */
for(i = 0; i < 10; i++)
mape[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 10){
p2 = cadenit;
while('p != ' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit,"-1")!=0) mape[i] = 1;
switch(i){
case 0: strcpy(cade1,cadenit); break;
case 1: strcpy(cade2,cadenit); break;
case 2: strcpy(cade3,cadenit); break;

```

```

case 3: strcpy(cade4,cadenit); break;
case 4: strcpy(cade5,cadenit); break;
case 5: strcpy(cade6,cadenit); break;
case 6: strcpy(cade7,cadenit); break;
case 7: strcpy(cade8,cadenit); break;
case 8: strcpy(cade9,cadenit); break;
case 9: strcpy(cade10,cadenit); break;
}
i++;
}
if((fp = fopen("\\GESTION\\SQL\\invita.sql","w"))==NULL)
return(FAIL);
fprintf(fp,"select id,feceven,dependencia,asunto,\n");
fprintf(fp,"ant,sub,horeven,feccrec from gesgral where \n");
fprintf(fp,"tipdoc=' '");
fig_and=1;
for(i = 0; i < 10; i++){
if(mapa[i]==1){
switch(i){
case 0:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"id between %s and %s ",cade1,cade2);
fig_and = 1;
case 1: break;
case 2:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"feccrec >= to_date('%s','\dd-mm-yy') and feccrec <= to_date('%s','\dd-mm-yy')
",cade3,cade4);
fig_and = 1;
case 3: break;
case 4:
if((fig_and == 1) fprintf(fp,"n and ");
fprintf(fp,"feceven >= to_date('%s','\dd-mm-yy') and feceven <= to_date('%s','\dd-mm-yy')
",cade5,cade6);
fig_and = 1;
case 5: break;
case 6:
if((fig_and == 1) fprintf(fp,"n and ");
if(strchr(cade7,',')==NULL)
cade7[strlen(cade7)-1] = '\0';
p = cade7; p2 = cadenit;
while(*p){
if(*p == ','){
*p2++ = '\t'; *p2++ = ',';
*p2++ = '\t'; p++;
}
else
*p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp,"tipdoc in ('%s')",cadenit);
fig_and = 1;
break;
case 7:
if((fig_and == 1) fprintf(fp,"n and ");
if(strchr(cade8,',')==NULL)
cade8[strlen(cade8)-1] = '\0';
p = cade8; p2 = cadenit;
while(*p){
if(*p == ','){
*p2++ = '\t'; *p2++ = ',';
*p2++ = '\t'; p++;
}
else
*p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp,"id in (select id from gesturno where upper(turno) in ('%s'))",cadenit);
fig_and = 1;
break;
case 8:

```

## A.5 Listado de 16 programas user exit

```

        if(flag_and == 1) fprintf(fp,"n and ");
        if(strchr(cade9,'')!=NULL)
            cade9[strlen(cade9)-1] = '\0';
        p = cade9; p2 = cadenit;
        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ',';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        fprintf(fp,"clasif in ('%s\') ",cadenit);
        flag_and = 1;
        break;
    case 9:
        if(flag_and == 1) fprintf(fp,"n and ");
        if(strchr(cade10,'')!=NULL)
            cade10[strlen(cade10)-1] = '\0';
        p = cade10; p2 = cadenit;
        while(*p){
            if(*p == ','){
                *p2++ = '\n'; *p2++ = ',';
                *p2++ = '\n'; p++;
            }else
                *p2++ = *p++;
        }
        *p2++ = '\0';
        if (cadenit[0]== "")
            fprintf(fp,"id in (select id from gesturno where upper(instruc) not in ('L'))");
        else
            fprintf(fp,"id in (select id from gesturno where upper(instruc) in ('%s\') ",cadenit);
        flag_and = 1;
        break;
    }
}
fprintf(fp,"n order by feceven ");
fclose(fp);
return(OK);
}
/*----- invita -----*/
** invita -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise. ***/
int invita()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;
    /* El argumento es la cadena con los parametros */
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\teminv.txt","w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,ar,"ofgest.COMANDO");
    fld_gral.len = strlen("ofgest.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"""%s""\n",fld_buf_gral);
    fclose(fp);
    if (invacq()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

.....
.....
***** NAME *****
***** InvTur.pc - User Exit de Impresion de Documentos Turnados (Invitaciones) *****
***** DESCRIPCION *****
***** Genera un archivo INVTUR, para despues generar un archivo *****
***** INVTUR.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Impresion de Documentos Turnados. (Invitaciones) *****
.....
.....

#ifndef RCSID
static char *RCSid =
"$Header: /forms41/prd/RCS/InvTur.pc,v 1.3 1995/05/23 18:20:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/** Some useful Pro*C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int InvTur();
FILE *fp;
/** sqlinvtu -- Funcion que genera un archivo query externo */
int sqlinvtu(void)
int mapa[27],fig_and;
char cadenot[790],cadenit[790];
char cade1[790];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if(((fp = fopen("\GESTION\TEMP\InvTur.Txt","r"))==NULL)
return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 27; i++)
mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 1){
p2 = cadenit;
while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1, cadenit); break; /* Numdoc. */
}
i++;
}
if(((fp = fopen("\GESTION\SQL\InvTur.SQL","w"))==NULL)
return(FAIL);
fprintf(fp, "select id,numdoc,feccdoc,dependencia,ref,asunto\n");
fprintf(fp, "from gesgral\n");
fprintf(fp, "where tipdoc='\G\' and clasif=\\1+\' and ");

```

## A.5 Listado de 16 programas user exit

```

fig_and=1;
for(i = 0; i < 1; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                if(strchr(cade1, ',') != NULL)
                    cade1[strlen(cade1)-1] = '\0';
                p = cade1; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = '\n';
                        *p2++ = '\n'; p++;
                    }else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp, " numdoc in (%s)", cade1);
                fig_and = 1;
                break;
            }
        }
    }
    fprintf(fp, "\n order by numdoc");
    fclose(fp);
    return(OK);
}

/*----- InvTUR -----*/
** INVTUR -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/

int invtur( )
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral(800); /* Valor del campo del Buffer */
        VARCHAR fld_gral(800); /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros ***/
    memset(fld_buf_gral, 0, sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\InvTur.Txt", "w")) == NULL)
        return( FAIL );
    strcpy(fld_gral.arr, "OFGEST.CADENA1");
    fld_gral.len = strlen("OFGEST.CADENA1");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=799;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp, "%s", fld_buf_gral);
    fclose(fp);
    if (sqlinvtu()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return( OK );
}

```

```

.....
.....
***** NAME *****
***** ObsTur.pc - User Exit de Impresion de Documentos Turnados (Obsequios) *****
***** DESCRIPCION *****
***** Genera un archivo OBSTUR, para despues generar un archivo *****
***** OBSTUR.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Impresion de Documentos Turnados.(Obsequios) *****
.....
.....
#ifdef RCSID
static char 'RCSid =
"$Header: /forms41/prd/RCS/ObsTur.pc,v 1.3 1995/07/14 12:20:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/* Some usefui Pro'C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int obstur();
FILE *fp;
/*** sqlobstu - Funcion que genera un archivo query externo***/
int sqlobstu(void){
int mapa[27],fig_and;
char cadenot[790],cadenit[790];
char cade1[790];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\ObsTur.Txt","r"))==NULL)
return( FAIL );
fgets(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 2; i++){
mapa[i] = 0;
}
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 1){
p2 = cadenit;
while('p != '\0' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit,"-1")!=0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1,cadenit); break; /* Numdoc. */
}
i++;
}
if((fp = fopen("\\GESTION\\SQL\\ObsTur.SQL","w"))==NULL)
return(FAIL);
fprintf(fp,"select id,numdoc,feccdoc,dependencia,ref,asunto\n");
fprintf(fp,"from gesgral\n");
fprintf(fp,"where tipdoc='\R\' and ");
fig_and=1;
for(i = 0; i < 1; i++){

```

```

        if(mapa[i]==1){
            switch(i){
                case 0:
                    if(strchr(cade1,',')!=NULL)
                        cade1[strlen(cade1)-1] = '\0';
                    p = cade1; p2 = cadenit;
                    while(*p){
                        if(*p == ','){
                            *p2++ = '\n'; *p2++ = *p++;
                            *p2++ = '\n'; p++;
                        }else
                            *p2++ = *p++;
                    }
                    *p2++ = '\0';
                    fprintf(fp," numdoc in (%s)\n",cade1);
                    fig_and = 1;
                    break;
            }
        }
    }
    fprintf(fp,"\n order by numdoc");
    fclose(fp);
    return(OK);
}

/*----- OBSTUR -----
** OBSTUR -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/

int obstur()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[800]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[800]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros **/
    memcpy(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\ObsTur.Txt","w"))==NULL)
        return( FAIL );
    strcpy(fld_gral.arr,"OFGEST.CADENA1");
    fld_gral.len = strlen("OFGEST.CADENA1");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=799;
    while(fld_buf_gral[i]!='\n') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"%s\n",fld_buf_gral);
    fclose(fp);
    if (sqlobstu()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

/.....
.....
***** NAME *****
***** oficio.pc - User Exit GEneracion de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL (temofi), para despues generar un archivo *****
***** SQL (oficio.sql), generando asi un SQL externo, el cual se ejecutara *****
***** por medio del reporte de oficio.pc *****
/.....

#ifndef RCSID
static char *RCSid =
"$Header: //forms41/prd/RCS/oficio.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/** Some useful Pro*C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int oficio(); /* User exit */
FILE *fp;
/** ofisql - Funcion que genera un archivo query externo */
int ofisql(void)
int mapa[5],fg_and;
char aster[6] = " ";
char cadenot[600],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\temofi", "r"))==NULL)
return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 5; i++)
mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 5){
p2 = cadenit;
while('p' != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1, cadenit); break;
case 1: strcpy(cade2, cadenit); break;
case 2: strcpy(cade3, cadenit); break;
case 3: strcpy(cade4, cadenit); break;
case 4: strcpy(cade5, cadenit); break;
case 5: strcpy(cade6, cadenit); break;
}
i++;
}
if((fp = fopen("\\GESTION\\SQL\\oficio.sql", "w"))==NULL)

```



## A.5 Listado de 16 programas user exit

```

return(FAIL);
fprintf(fp,"select id,numdoc,feccdoc,asunto \n");
fprintf(fp,"from gesgral \n");
fprintf(fp,"where tipdoc='O' \n");
fig_and=0;
if(i > 0)
    fprintf(fp,"and \n");
for(i = 0; i < 8; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                fprintf(fp,"numdoc between %s and %s ",cade1,cade2);
                fig_and = 1;
            case 1: break;
            case 2:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade3,',')!=NULL)
                    cade3[strlen(cade3)-1] = '\0';
                p = cade3; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"id in (select id from gesturno where upper(turno) in ('%s'))",cadenit);
                fig_and = 1;
                break;
            case 3:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade4,',')!=NULL)
                    cade4[strlen(cade4)-1] = '\0';
                p = cade4; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                fprintf(fp,"clasi in ('%s')",cadenit);
                fig_and = 1;
                break;
            case 4:
                if(fig_and == 1) fprintf(fp,"n and ");
                if(strchr(cade5,',')!=NULL)
                    cade5[strlen(cade5)-1] = '\0';
                p = cade5; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\n'; *p2++ = ' ';
                        *p2++ = '\n'; p++;
                    }
                    else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                if (cadenit[0]=="")
                    fprintf(fp,"id in (select id from gesturno where upper(instruc) not in ('L'))");
                else
                    fprintf(fp,"id in (select id from gesturno where upper(instruc) in ('%s'))",cadenit);
                fig_and = 1;
                break;
        }
    }
}
}

```

```

    fprintf(fp,"ln order by numdoc ");
    fclose(fp);
    return(OK);
}
/*----- viaje -----*/
**oficio -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int oficio()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;
    /** El argumento es la cadena con los parametros */
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\temofit","w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,arr,"ofgest.comando");
    fld_gral.len = strlen("ofgest.comando");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    l=499;
    while(fld_buf_gral[l]!=' ') l--;
    fld_buf_gral[l+1] = '\0';
    fprintf(fp,""%"s""\n",fld_buf_gral);
    fclose(fp);
    if (ofisql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

/*****
*****
***** NAME *****
***** reldis.pc - User Exit de CONSULTA ESPECIAL archivo TEMPORAL *****
***** DESCRIPCION *****
***** Genera un archivo TEMPORAL (temdis), para despues generar un archivo *****
***** RELDIS.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Consultas Especiales. *****
*****
*****/
#define RCSID
static char *RCSid =
"$Header: /forms41/prd/RCS/reldis.pc,v 1.3 1995/05/18 20:30:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifndef UE
#include "ue.h"
#endif
#define externdef globalef
#define OK 0
#define FAIL 1
/** Some useful Pro"C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifndef WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int reldis();
FILE *fp;
/** GENSQLUNO -- Funcion que genera un archivo query externo */
int relsq(void){

```

## A.5 Listado de 16 programas user exit

```

int mapa[9],fg_and;
char cadenot[600],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80];
char cade7[80],cade8[80],cade9[80],cade10[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\verdis","r"))==NULL)
    return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 9; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 10){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit, "-1") != 0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1, cadenit); break;
        case 1: strcpy(cade2, cadenit); break;
        case 2: strcpy(cade3, cadenit); break;
        case 3: strcpy(cade4, cadenit); break;
        case 4: strcpy(cade5, cadenit); break;
        case 5: strcpy(cade6, cadenit); break;
        case 6: strcpy(cade7, cadenit); break;
        case 7: strcpy(cade8, cadenit); break;
        case 8: strcpy(cade9, cadenit); break;
        case 9: strcpy(cade10, cadenit);
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\verdis.sql","w"))==NULL)
    return(FAIL);
fprintf(fp, "select id,tipdoc,numdoc,feceven,dependencia,\n");
fprintf(fp, "asunto,ant,sub from gesagral where id in (select id \n");
fprintf(fp, "from gesexp where substr(gesexp.expediente,1,7)='001.004' \n");
fg_and=0;
if(i > 0)
    fprintf(fp, "and \n");
for(i = 0; i < 10; i++){
    if(mapa[i]==1){
        switch(i){
            case 0:
                if(fg_and == 1) fprintf(fp, "\n and ");
                fprintf(fp, "id between %s and %s ", cade1, cade2);
                fg_and = 1;
                break;
            case 1: break;
            case 2:
                if(fg_and == 1) fprintf(fp, "\n and ");
                fprintf(fp, "fecrec >= to_date('%s', '\dd-mm-yy') and fecrec <= to_date('%s', '\dd-mm-yy')
", cade3, cade4);
                fg_and = 1;
                break;
            case 3: break;
            case 4:
                if(fg_and == 1) fprintf(fp, "\n and ");
                fprintf(fp, "feceven >= to_date('%s', '\dd-mm-yy') and feceven <= to_date('%s', '\dd-mm-yy')
", cade5, cade6);
                fg_and = 1;
                break;
            case 5: break;
            case 6:
                if(fg_and == 1) fprintf(fp, "\n and ");
                if(strchr(cade7, '.') != NULL)
                    cade7[strlen(cade7)-1] = '\0';

```

```

p = cade7; p2 = cadentit;
while(*p){
    if(*p == ','){
        *p2++ = '\n'; *p2++ = ' ';
        *p2++ = '\n'; p++;
    }
    *p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp, "tipdoc in (%s)\n", cadentit);
fig_and = 1;
break;
case 7:
if((fig_and == 1) fprintf(fp, "\n and ");
if(strchr(cade8, ',') != NULL)
    cade8[strlen(cade8)-1] = '\0';
p = cade8; p2 = cadentit;
while(*p){
    if(*p == ','){
        *p2++ = '\n'; *p2++ = ' ';
        *p2++ = '\n'; p++;
    }
    *p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp, "id in (select id from gesturno where upper(turno) in (%s))", cadentit);
fig_and = 1;
break;
case 8:
if((fig_and == 1) fprintf(fp, "\n and ");
if(strchr(cade9, ',') != NULL)
    cade9[strlen(cade9)-1] = '\0';
p = cade9; p2 = cadentit;
while(*p){
    if(*p == ','){
        *p2++ = '\n'; *p2++ = ' ';
        *p2++ = '\n'; p++;
    }
    *p2++ = *p++;
}
*p2++ = '\0';
fprintf(fp, "clasif in (%s)\n", cadentit);
fig_and = 1;
break;
case 8:
if((fig_and == 1) fprintf(fp, "\n and ");
if(strchr(cade10, ',') != NULL)
    cade10[strlen(cade10)-1] = '\0';
p = cade10; p2 = cadentit;
while(*p){
    if(*p == ','){
        *p2++ = '\n'; *p2++ = ' ';
        *p2++ = '\n'; p++;
    }
    *p2++ = *p++;
}
*p2++ = '\0';
if (cadentit[0] == '*')
    fprintf(fp, "id in (select id from gesturno where upper(instruc) not in (%s))", cadentit);
else
    fprintf(fp, "id in (select id from gesturno where upper(instruc) in (%s))", cadentit);
fig_and = 1;
break;
}
}
fprintf(fp, "\n and feceven is not null order by feceven \n");
fclose(fp);

```

```

        return(OK);
    }

/*----- reldis -----*/
** reldis - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int reldis( )
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros***/
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\reldis", "w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,"OFGEST.COMANDO");
    fld_gral.len = strlen("OFGEST.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp, "%s\n", fld_buf_gral);
    fclose(fp);
    if (rcsql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```

```

/*****
*****
***** NAME *****
***** Reporte.pc - User Exit de Reporte *****
***** DESCRIPCION *****
***** Genera un archivo GenRep.bat, para despues ejecutarlo desde la forma *****
*****
*****/
#ifdef RCSID
    static char *RCSId =
        "$Header: /forms41/prd/RCS/reporte.pc,v 1.3 1995/05/22 20:30:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/** Some useful Pro*C defs **/
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int reporte();
FILE *fp;
int sqlcur(void);
int maps[27];
char cadenot[600], cadent[80];

```

```

char cade1[80],cade2[80],cade3[80],cade4[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\GenRep", "r"))==NULL)
    return( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 4; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 4){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit, "-1")=0) mapa[i] = 1;
    switch(i){
        case 0: strcpy(cade1,cadenit); break;
        case 1: strcpy(cade2,cadenit); break;
        case 2: strcpy(cade3,cadenit); break;
        case 3: strcpy(cade4,cadenit); break;
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\gestur.sql", "w"))==NULL)
    return(FAIL);
printf(fp, "select cve_tur,nomtur,cargo,area,esposa,calleof, \n");
printf(fp, "colof, cpor, ciudof, telof, faxof, calledom, coldom, \n");
printf(fp, "cpdom, ciudadom, telodom, faxdom from turno \n");
printf(fp, "where upper(cve_tur) like upper('%s\') and \n", cade1);
printf(fp, "upper(nomtur) like upper('%s\') and \n", cade2);
printf(fp, "upper(cargo) like upper('%s\') and \n", cade3);
printf(fp, "upper(area) like upper('%s\') \n", cade4);
printf(fp, "order by cve_tur");
fclose(fp);
return(OK);
}
/*----- REPORTE -----*/
** REPORTE - User eXit Genera un archivo temporal
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise. **/
int reporte( )
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
    char fld_buf_gral[200]; /* Valor del campo del Buffer */
    VARCHAR fld_gral[200]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /* ** El argumento es la cadena con los parametros **/
    memset(fld_buf_gral, 0, sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\GenRep", "w"))==NULL)
        return( FAIL );
    strcpy(fld_gral.arr, "DATOS.COMANDO");
    fld_gral.len = strlen("DATOS.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=199;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    printf(fp, "%s\n", fld_buf_gral);
    fclose(fp);
    if (sqltur()==0) return( FAIL ); /* Genera el query Externo*/
    return(OK);
}

```

## A.5 Listado de 16 programas user exit

```

.....
.....
***** NAME *****
***** Turno.pc - User Exit de Impresion de Turnos *****
***** DESCRIPCION *****
***** Genera un archivo TURNO, para despues generar un archivo *****
***** TURNO.SQL, generando asi un SQL externo, el cual se ejecutara *****
***** por medio de la forma de Impresion de Turnos. *****
...../
#ifdef RCSID
    static char *RCSid =
        "$Header: /forms41/prd/RCS/Turno.pc,v 1.3 1995/05/23 10:30:01 MLLC Exp $ ";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/** Some useful Pro*C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int turno();
FILE *fp;
/** TURSQR - Funcion que genera un archivo query externo */

int tursqr(void){
int mapa[27],fig_and;
char cadenot[80],cadenit[80];
char cade1[80],cade2[80],cade3[200];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if((fp = fopen("\\GESTION\\TEMP\\Turno","r"))==NULL)
    return( FAIL );
fgetc(cadenot,sizeof(cadenot),fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 7; i++)
    mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 3){
    p2 = cadenit;
    while(*p != '|' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
    if(strcmp(cadenit,"-1")!=0) mapa[i] = 1;
    switch(i){
    case 0: strcpy(cade1,cadenit); break; /* Numdoc. Ini */
    case 1: strcpy(cade2,cadenit); break; /* Numdoc. Fin */
    case 2: strcpy(cade3,cadenit); break; /* Turno */
    }
    i++;
}
if((fp = fopen("\\GESTION\\SQL\\Turno.sql","w"))==NULL)
    return(FAIL);
fprintf(fp,"select gesgral.id,gesgral.numdoc,gesgral.ref,gesgral.fecdoc,v");

```

```

fprintf(fp,"turno.nomtur,turno.cargo,gesturno.turno,gesgral.asunto \n");
fprintf(fp,"from gesgral,gesturno,turno where gesgral.id=gesturno.id and \n");
fprintf(fp,"gesturno.turno=turno.cve_tur and gesgral.tipdoc='G' \n ");
    flg_and=1;
    for(i = 0; i < 6; i++){
        if(mapa[i]==1){
            switch(i){
            case 0:
                if(flg_and == 1) fprintf(fp," and ");
                fprintf(fp,"numdoc between %s and %s ",cade1,cade2);
                flg_and = 1;
                break;
            case 1: break;
            case 2:
                if(flg_and == 1) fprintf(fp,"\n and ");
                if(strchr(cade3,',')==NULL)
                    cade3[strlen(cade3)-1] = '\0';
                p = cade3; p2 = cadenit;
                while(*p){
                    if(*p == ','){
                        *p2++ = '\1'; *p2++ = '\1';
                        *p2++ = '\1'; *p2++ = '\1';
                    }else
                        *p2++ = *p++;
                }
                *p2++ = '\0';
                /*cadeint*/
                fprintf(fp,"gesturno.turno in (%s) ",cade3);
                flg_and = 1;
                break;
            }
        }
    }
    fprintf(fp,"\n order by numdoc");
    fclose(fp);
    return(OK);
}
/*----- TRUNO -----*/
** TURNO - User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.***/
int turno()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral[500]; /* Valor del campo del Buffer */
        VARCHAR fld_gral[500]; /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /*** El argumento es la cadena con los parametros***/
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\Turno","w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,ar,"OFGEST.COMANDO");
    fld_gral.len = strlen("OFGEST.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\n') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"%s\n",fld_buf_gral);
    fclose(fp);
    if (turno()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}

```



## A.5 Listado de 16 programas user exit

```
.....
.....
***** NAME *****
***** viaje.pc - User Exit GENERACION de archivo TEMoral *****
***** DESCRIPCION *****
***** Genera un archivo TEMVIAJE, para despues generar un archivo *****
***** VIAJE.SQL, generando así un SQL externo, el cual se ejecutara *****
***** por medio del reporte de viajes. *****
.....
...../

#ifdef RCSID
static char 'RCSid =
"$Header: /forms41/prd/RCS/viaje.pc,v 1.3 1995/05/03 18:09:01 MLLC Exp $";
#endif /* RCSID */
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef UE
#include "ue.h"
#endif
#define externdef globaldef
#define OK 0
#define FAIL 1
/* Some useful Pro'C defs */
#define SQLCA_FAILURE (sqlca.sqlcode != 0)
#define VCHNULL(v) v.arr[v.len] = '\0'
#ifdef _WINDLL
#define SQLCA_STORAGE_ extern
#endif /* _WINDLL */

EXEC SQL INCLUDE sqlca.h;
extern int viaje(); /* User exit */
FILE *fp;
/* viajesql - Funcion que genera un archivo query externo */
int viajesql(void)
int mapa[8],fig_and;
char aster[8] = " ";
char cadenot[80],cadenit[80];
char cade1[80],cade2[80],cade3[80],cade4[80],cade5[80],cade6[80],cade7[80],
cade8[80],cade9[80];
int i;
char *p,*p2;
/*Abre el archivo temporal*/
if(((fp = fopen("\\GESTION\\TEMP\\werviaje", "r"))==NULL)
return ( FAIL );
fgets(cadenot, sizeof(cadenot), fp);
cadenot[strlen(cadenot)-1] = '\0';
fclose(fp);
/* Limpia la variable mapa */
for(i = 0; i < 8; i++)
mapa[i] = 0;
p = cadenot;
i = 0;
/* Separa las variables e identifica cual es valida */
while(i < 8){
p2 = cadenit;
while("p != ' ' && *p) *p2++ = *p++; *p2++ = '\0'; p++;
if(strcmp(cadenit, "-1")=0) mapa[i] = 1;
switch(i){
case 0: strcpy(cade1,cadenit); break;
case 1: strcpy(cade2,cadenit); break;
case 2: strcpy(cade3,cadenit); break;
case 3: strcpy(cade4,cadenit); break;
case 4: strcpy(cade5,cadenit); break;
case 5: strcpy(cade6,cadenit); break;
case 6: strcpy(cade7,cadenit); break;
case 7: strcpy(cade8,cadenit); break;
case 8: strcpy(cade9,cadenit); break;
}
}
```

```

        i++;
    }
    if((fp = fopen("\\GESTION\\SQL\\viaje.sql","w"))==NULL)
        return(FAIL);
    fprintf(fp,"select id,tipdoc,numdoc,feceven,horeven,\n");
    fprintf(fp,"asunto from gesgral where \n");
    fprintf(fp,"gesgral.id in (select id from gesexp where \n");
    fprintf(fp,"substr(gesexp.expediente,1,7)='001.010' \n");
    fig_and=0;
    if(i > 0)
        fprintf(fp,"and \n");
    for(i = 0; i < 8; i++){
        if(mapa[i]==1){
            switch(i){
                case 0:
                    fprintf(fp,"id between %s and %s ",cade1,cade2);
                    fig_and = 1;
                case 1: break;
                case 2:
                    if(fig_and == 1) fprintf(fp," \n and ");
                    fprintf(fp,"fecrec >= to_date('%s','\dd-mm-yy') and fecrec <= to_date('%s','\dd-mm-yy')
",cade3,cade4);
                    fig_and = 1;
                case 3: break;
                case 4:
                    if(fig_and == 1) fprintf(fp," \n and ");
                    fprintf(fp,"feceven >= to_date('%s','\dd-mm-yy') and feceven <= to_date('%s','\dd-mm-yy')
",cade5,cade6);
                    fig_and = 1;
                case 5: break;
                case 6:
                    if(fig_and == 1) fprintf(fp," \n and ");
                    if(strchr(cade7,',')!=NULL)
                        cade7[strlen(cade7)-1] = '\0';
                    p = cade7; p2 = cadenit;
                    while(*p){
                        if(*p == ','){
                            *p2++ = '\n'; *p2++ = ',';
                            *p2++ = '\n'; p++;
                        }
                        else
                            *p2++ = *p++;
                    }
                    *p2++ = '\0';
                    fprintf(fp,"lipdoc in (%s)\n",cadenit);
                    fig_and = 1;
                    break;
                case 7:
                    if(fig_and == 1) fprintf(fp," \n and ");
                    if(strchr(cade8,',')!=NULL)
                        cade8[strlen(cade8)-1] = '\0';
                    p = cade8; p2 = cadenit;
                    while(*p){
                        if(*p == ','){
                            *p2++ = '\n'; *p2++ = ',';
                            *p2++ = '\n'; p++;
                        }
                        else
                            *p2++ = *p++;
                    }
                    *p2++ = '\0';
                    fprintf(fp,"clasef in (%s)\n",cadenit);
                    fig_and = 1;
                    break;
            }
        }
    }
    fprintf(fp," \n and feceven is not null order by feceven ");
    fclose(fp);
    return(OK);
}

```

```
/*----- viaje -----*/
** viaje -- User eXit Genera un archivo temporal y general un Query Externo
** Argumento 1 - Son todas las variables concatenadas
** This function returns SUCCESS if there are no errors,
** FAILURE otherwise.*/
int viaje()
{
    int i;
    EXEC SQL BEGIN DECLARE SECTION;
        char fld_buf_gral(500); /* Valor del campo del Buffer */
        VARCHAR fld_gral(500); /* El campo de la forma */
    EXEC SQL END DECLARE SECTION;
    EXEC SQL INCLUDE SQLCA;

    /** El argumento es la cadena con los parametros */
    memset(fld_buf_gral,0,sizeof(fld_buf_gral));
    if((fp = fopen("\\GESTION\\TEMP\\viviage", "w"))==NULL)
        return( FAIL );
    strcpy(fld_gral,err,"ofgest.COMANDO");
    fld_gral.len = strlen("ofgest.COMANDO");
    EXEC TOOLS GET :fld_gral INTO :fld_buf_gral;
    i=499;
    while(fld_buf_gral[i]!='\0') i--;
    fld_buf_gral[i+1] = '\0';
    fprintf(fp,"%s\n",fld_buf_gral);
    fclose(fp);
    if (viajesql()==0) return( FAIL ); /* Genera el Query externo para el reporte */
    return ( OK );
}
```

# APÉNDICE B

## LISTADO DE FIGURAS

---

<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
1.1	Organización General de Cualquier Empresa	2
2.1	Componentes Generales de la Implementación	16
2.2	Configuración Monousuario	17
2.3	Configuración Red	18
3.1	Representación de un Proceso	23
3.2	Flujo de Entrada y Flujo de Salida en un proceso	24
3.3	Representación gráfica del Archivo	25
3.4	Representación gráfica de la entidad externa	27
3.5	Ejemplo de como dibujar una entidad	33
3.6	Formato para escribir la relación entre dos entidades	34
3.7	Opcionalidad	34
3.8	El grado representado por una "pata de gallo" y una línea simple	34
3.9	Representación gráfica de una relación entre entidades	35
3.10	Matriz de Relaciones	36
3.11	La barra ID indica que la relación con el Banco es parte del ID de CUENTA	38
3.12 a.	LA TAREA ASIGNADA no tiene ID	38
3.12 b.	Una TAREA ASIGNADA es identificada de manera única por el EMPLEADO a quien se le dio la TAREA ASIGNADA, por el PROYECTO que contiene la TAREA ASIGNADA y por la fecha asignada	39
3.13	Candidatos a ID	39
3.14	Cada ID se marca con (#) o se deja sin ninguna marca	40
3.15	En la entidad CLIENTE	40
3.16	Ejemplo de una relación M: M	41
3.17	Ejemplo de una Entidad Intersección	42
3.18	Diagrama Típico de una Transición de Estados	48
3.19	Cambios de Estado	49
3.20	Estados inicial y final	50

---

<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
3.21	Estados finales múltiples de un sistema	51
3.22	Muestra de condiciones y acciones	52
3.23	Dos niveles de un DTE	53
3.24	DTE incompleto	54
3.26	Mapas de Instancias	61
4.1	Diagrama de Contexto SICADE	68
4.2	Diagrama de Flujo de Datos Nivel 0. SICADE	69
4.3	DFD-1 Recibir los Documentos	70
4.4	DFD-2 Controlar (Digitalizar) los Documentos	71
4.5	DFD-3 Generar los reportes (1/2)	74
4.6	DFD-4 Generar los Reportes (2/2)	74
4.7	DFD-4 Mantener los Catálogos	75
4.8	DFD-5 Mantener el Sistema	76
4.9	Modelo de Entidad Relación	79
4.10	Cuadrantes de la Matriz de Relación	80
5.1	Menú General	92
5.2	Menú de Catálogos	92
5.3	Menú de A,B y C de Catálogos o Consultas a Catálogos	93
5.4	Menú de Documentos	94
5.5	Menú de Reportes	95
5.6	Menú de Utilerias	96
6.1	Grupo e Icono Principal del Sistema	106
6.2	Ventana de Seguridad del Sistema	107
6.3	Menú Principal del Sistema de Control y Automatización de Documentos	107
6.4	Estructura Principal del Sistema	108
6.5	Submenú del menú de Catálogos	109
6.6	Opciones del submenú de altas, bajas y cambios de catálogos	109
6.7	Pantalla de Catalogo de Clasificaciones	110

---

<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
6.8	Funciones de los iconos del catálogo de Clasificaciones	110
6.9	Submenú de Acción	111
6.10	Ayuda de Teclas oprimiendo Alt + T	111
6.11	Ventanas de Confirmación para borrar un registro	112
6.12	Lista de Clasificaciones	112
6.13	Pantalla del catalogo de documentos	113
6.14	Pantalla del Catalogo de Instrucciones	113
6.15	Pantalla del Catalogo de Temas	114
6.16	Pantalla del Catalogo de Expedientes	114
6.17	Pantalla del Catalogo de Dependencias	115
6.18	Pantalla del Catalogo de Turnados	116
6.19	Iconos del Catalogo de Turnados	116
6.20	Menú de Edición	116
6.21	Menú de Acción	117
6.22	Pantalla de Consultas de Turnados	118
6.23	Menú de Consultas	118
6.24	Opciones para las Consultas de Catálogos	119
6.25	Catálogos de Clasificaciones	120
6.26	Iconos de Pantalla de Consultas (Clasificaciones)	120
6.27	Reporte de Clasificaciones	121
6.28	Funciones básicas de los iconos de reportes	121
6.29	Pantalla de Configuración de Impresión	122
6.30	Catalogo de Documentos	122
6.31	Catalogo de Dependencias	123
6.32	Catalogo de Temas	123
6.33	Catalogo de Expedientes	124
6.34	Catálogos de Instrucciones	124
6.35	Catálogos de Turnados	125
6.36	Menú de Reportes de Turnados	125

<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
6.37	Reporte General de Turnados	126
6.38	Reporte de Catalogo	126
6.39	Menú de Etiquetas	127
6.40	Pantalla de Cuatro Etiquetas	127
6.41	Menú de Documentos	128
6.42	Pantalla Completa de la pantalla de Altas y Cambios de Documentos	128
6.43	Catálogos de Documentos	129
6.44	Menú de Acción	129
6.45	Grupo de Iconos de la Pantalla de Altas y Cambio de Gestión	129
6.46	Menú de Clasificaciones	130
6.47	Menú de Dependencias	130
6.48	Menú de Temas	131
6.49	Menú de Expedientes	131
6.50	Menú de Turnados	132
6.51	Menú de Instrucciones	132
6.52	Menú de Editar	133
6.53	Editor de Textos presentado en los campos de Asunto y Resolución	133
6.54	Pantalla de un registro dado de alta	134
6.55	Menú de Teclas	135
6.56	Pantalla de Consultas Generales	135
6.57	Confirmación de la Búsqueda	136
6.58	Generación de reporte por pantalla, presentación de distintos iconos	136
6.59	Presentación del manejo de impresión	137
6.60	Consulta Alfabética	138
6.61	Pantalla de Consultas Alfabéticas	138
6.62	Consultas de Árbol	139
6.63	Registro de Transformación	140



<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
6.64	Pantalla de Bajas de Gestiones	140
6.65	Pantalla de Consultas Especiales	141
6.66	Reporte por pantalla de Consultas Especiales	141
6.67	Menú de Reportes	142
6.68	Navegación entre los Campos	143
6.69	Funciones de los iconos	143
6.70	Pantalla de la impresión de Oficios Turnados	144
6.71	Documentos Turnados	144
6.72	Relación de Discursos	145
6.73	Programa de Viajes	145
6.74	Relación de Oficios Despachados	146
6.75	Documentos para Firma	146
6.76	Relación de Proyectos de Decreto	147
6.77	Catalogo Alfabético	147
6.78	Relación de Invitaciones	148
6.79	Estadística de Documentos	148
6.80	Menú de Utilerías	149
6.81	Pantalla de Confirmación de la clave del administrador	149
6.82	(Pantalla principal de Respaldo de Datos	149
6.83	Pantalla de presentación de tablas o usuarios	150
6.84	Pantalla de opciones avanzadas	151
6.85	Pantalla principal de Restauración de Datos	152
6.86	Pantalla de opciones avanzadas	153
6.87	Salida del sistema	154
7.1	Ambiente de Procesamiento basado en un Host	159
7.2	Ambiente de procesamiento Maestro/Esclavo	160
7.3	Ambiente de Procesamiento de dispositivos compartidos	160
7.4	Ambiente de Procesamiento Cliente/Servidor	161
7.5	Ambiente Operativo Cliente/Servidor de SICADE	163

<b>Figura</b>	<b>Nombre de la Figura</b>	<b>Página</b>
7.6	Estructuras de Directorios de la Base de Datos	164
7.7	Estructura de Directorios para SICADE	164
7.8	Diagrama de Entidad Relación	170
7.9	Distribución de menús de SICADE	171

# APÉNDICE C

## LISTADO DE TABLAS

---

<b>Tabla</b>	<b>Tema</b>	<b>Página</b>
Tabla 3.1	Reglas de las Formas Normales	45
Tabla 3.2	Notación del Diccionario de Datos	56
Tabla 4.1	Cuadrante I de la Matriz de Relación	80
Tabla 4.2	Cuadrante II de la Matriz de Relación	81
Tabla 4.3	Cuadrante III de la Matriz de Relación	81
Tabla 4.4	Cuadrante IV de la Matriz de Relación	82
Tabla 5.1	Menú Principal	97
Tabla 5.2	Catálogos	97
Tabla 5.3	Altas, Bajas y Cambios	97
Tabla 5.4	Consultas	99
Tabla 5.5	Documentos	101
Tabla 5.6	Reporte	103
Tabla 5.7	Utilerias	104
Tabla 7.1	Organización de los Archivos de la Base de Datos	165
Tabla 7.2	Organización de los Archivos de SICADE	165
Tabla 7.3	GesExp(id)	165
Tabla 7.4	GesGral(id)	166
Tabla 7.5	GesRefAlf	166
Tabla 7.6	GesRem	166
Tabla 7.7	GesRes	167
Tabla 7.8	GesTem	167
Tabla 7.9	GesTurno	167
Tabla 7.10	Clasif	167
Tabla 7.11	Tema	168
Tabla 7.12	Dependencia	168
Tabla 7.13	Expediente	168
Tabla 7.14	Documento	168
Tabla 7.15	Turno	169
Tabla 7.16	Instruc	169

<b>Tabla</b>	<b>Tema</b>	<b>Página</b>
Tabla 7.17	Sicade (Menú Principal)	171
Tabla 7.18	Catálogos	172
Tabla 7.19	Altas, Bajas y Cambios	172
Tabla 7.20	Consultas	172
Tabla 7.21	Documentos	172
Tabla 7.22	Reporte	173
Tabla 7.23	Utilertías	173
Tabla 7.24	Formas	174
Tabla 7.25	Reportes	175
Tabla 7.26	User Exit	179
Tabla 7.27	Valores para respaldo de información	183
Tabla 7.28	Términos del Respaldo de Información	184
Tabla 7.29	Valores para la restauración de la Información	186
Tabla 7.30	Términos para la restauración de información	187
Tabla 7.31	Calendario de Mantenimiento	193
Tabla 7.32	Descripción de eventos para calendario de mantenimiento	194
Tabla 8.1	Menús Principales	202
Tabla 8.2	Opciones de Submenús	203
Tabla 8.3	Subopción Altas, Bajas y Cambios de Opción Catálogos	204
Tabla 8.4	Subopción Consultas a Catálogos de Opción Catálogos	204
Tabla 8.5	Opción de Menú Documentos	204
Tabla 8.6	Opción del Submenú de Reportes	205
Tabla 8.7	Opción del Submenú de Utilerias	205
Tabla 8.8	Pantalla Catálogo de Clasificación	206
Tabla 8.9	Pantalla Catálogo de Documentos	206
Tabla 8.10	Pantalla Catálogo de Instrucciones	206
Tabla 8.11	Pantalla Catálogo de Temas	207
Tabla 8.12	Pantalla Catálogo de Turnados	207
Tabla 8.13	Pantalla Catálogo de Expedientes	208

---

<b>Tabla</b>	<b>Tema</b>	<b>Página</b>
Tabla 8.14	Pantalla Catálogo de Dependencias	208
Tabla 8.15	Pantalla Catálogo de Clasificaciones	208
Tabla 8.16	Pantalla Catálogo de Documentos	209
Tabla 8.17	Pantalla Catálogo de Instrucciones	209
Tabla 8.18	Pantalla Catálogo de Temas de subopción ABC de Catálogos	209
Tabla 8.19	Pantalla Catálogo de Turnados de subopción ABC de Catálogos	209
Tabla 8.20	Pantalla Catálogo de Expedientes de subopción ABC de Catálogos	210
Tabla 8.21	Pantalla Catálogo de Dependencias de subopción ABC de Catálogos	210
Tabla 8.22	Pantalla Control de Gestiones de Opción Documentos	210
Tabla 8.23	Pantalla de Consultas Generales de Opción Documentos	211
Tabla 8.24	Pantalla Consultas por Arbol de Opción Documentos	211
Tabla 8.25	Pantalla Consultas Alfabéticas de Opción Documentos	211
Tabla 8.26	Pantalla Registro de Transformación de Opción Documentos	212
Tabla 8.27	Pantalla Borrar Gestiones de Opción Documentos	212
Tabla 8.28	Pantalla Consultas Especiales de Opción Documentos	212
Tabla 8.29	Pantalla Impresión de Oficios Turnados de Opción Documentos	213
Tabla 8.30	Pantalla Documentos Turnados de Opción Reportes	213
Tabla 8.31	Pantalla Relación de Discursos Turnados de Opción Reportes	213
Tabla 8.32	Pantalla Programa de Viajes Turnados de Opción Reportes	214
Tabla 8.33	Pantalla Relación de Oficios Turnados de Opción Reportes	214
Tabla 8.34	Pantalla Documentos para Firma de Opción Reportes	214
Tabla 8.35	Pantalla Relación de Proyectos de Opción Reportes	215

<b>Tabla</b>	<b>Tema</b>	<b>Página</b>
Tabla 8.36	Pantalla Catálogo Alfabético de Opción Reportes	215
Tabla 8.37	Pantalla Relación de Invitaciones de Opción Reportes	215
Tabla 8.38	Pantalla Estadísticas de Documentos de Opción Reportes	216
Tabla 8.39	Pantalla BD Exporter de Opción Utilerias	216
Tabla 8.40	Pantalla BD Importer de Opción Utilerias	217

# **GLOSARIO**





Las ciencias de la computación se caracterizan por incluir en su "vocabulario" una serie de frases o términos que para muchas personas resultan ser totalmente desconocidas. Sin embargo, cuando se puede disponer de herramientas que faciliten la comprensión y el trabajo, su utilizan sin dudar. Pues bien, el objetivo de este glosario es apoyar al usuario en la consulta de algún término desconocido.

<i>Users to Import</i>	Si se ha seleccionado la opción del modo de "Users", con esta opción podrá seleccionar las tablas que corresponden al usuario que desea restaurar.
<i>Users</i>	Especifica al proceso el modo de restaurar las entidades o tablas de la Base de Datos, pertenecientes a los usuarios
<i>User_data</i>	Cuando la base de datos es inicialmente instalada, este tablespace esta vacío. Por lo que hay que crear tablas e insertar datos en ellas, para generar la ocupación del espacio de este. Cuando se agregan usuarios en el sistema usando las herramientas del User Manager, utilizara automáticamente este tablespace
<i>Turnado</i>	Es la persona capaz de dar solución al oficio recibido.
<i>Tipo de documento</i>	Es la clasificación que recibe un documento o una gestión, por ejemplo: oficio, gestión, invitación, regalo, etc. Numero de documento. Al primer documento recibido el primer día laborable del año se le asigna el numero uno y así consecutivamente hasta el ultimo día laborable del año.
<i>Temporary_data</i>	La base de datos usa este tablespace para scratchpad, es decir datos de scratch creados por el Sistema de Mantenimiento de la Base de Datos Relacional ( RDBMS), mientras se procesan ciertas opciones o características de los comandos SQL. Si se requiere incrementar el tamaño de este espacio, también crecerá el tablespace User_data en tamaño.
<i>Tema</i>	Se asigna un titulo que engloba de manera general el asunto

---

<i>Tables to Import</i>	Si se ha seleccionado la opción del modo de "Tables", con esta opción podrá seleccionar las tablas que desea restaurar.
<i>Tables</i>	(1) Selección que generara el respaldo de Tablas o Entidades que se requieren. (2) Especifica al proceso el modo de restaurar algunas entidades o tablas
<i>System</i>	almacena el contenido del diccionario de datos, incluyendo definiciones de tablas, vistas y procedimientos de almacenamiento necesarios para el Sistema de Mantenimiento de la Base de Datos Relacional ( RDBMS ) . La información en esta área tiene mantenimiento automático.
<i>Statistics Type</i>	Permite intercambiar el tipo de las estadísticas de optimización de la base de datos para generarse cuando los datos respaldados son restaurados dentro de la base de datos.
<i>Specify</i>	(1) Si se ha seleccionado el modo de "Tables" o "Users", con esta opción se podrá seleccionar una lista de entidades o tablas o una lista de usuarios para restaurar. (2) Si se a seleccionado Tables o Owners, al seleccionar Specify, se podrá especificar las tablas o entidades de un usuario o los usuarios a respaldar
<i>Rows</i>	(1) Esta opción permitirá respaldar los renglones de las tablas de datos, ya que si no se selecciona únicamente se respaldara las definiciones de las base de datos (2) Especifica al proceso que los renglones serán restaurados
<i>Rollback_data</i>	La base de datos usará este tablespace para respaldar las transacciones que fallan por completo normalmente. Se hará lo necesario para expandir este tablespace si se tiene una gran ejecución o un volumen de datos transaccionados
<i>Resolución</i>	Es la respuesta que da el turno
<i>Referencia del documento</i>	La primera instancia o medio por el cual se recibe, si es una copia, fax, vía telefónica, etc

---

<i>Referencia alfabética</i>	Es la palabra o palabras que involucran al documento, por lo general se toma el nombre de la institución solicitante o el remitente
<i>Record Length</i>	Permite intercambiar la longitud del registro de intercambio para el respaldo
<i>Record Export in System Data</i>	Permite respaldar los procesos almacenados en la base de datos
<i>Owners</i>	Selección que generara el respaldo de los objetos que se requieren pertenecientes a un usuario en particular
<i>Log File</i>	Nombre del archivo de log o salida de resultados de los procesos de respaldo
<i>Keep Components Consistent</i>	Permite exportar la consistencia de lectura desde una vista de la base de datos
<i>Instrucción</i>	Es tipo de orden que recibe el turnado para dar seguimiento a la documentación, es decir, si la gestión es "para su conocimiento", "para conceder audiencia", etc
<i>Indexes</i>	(1) Esta opción permitirá respaldar los índices de la base de datos. (2) Especificara al proceso que los índices serán restaurados
<i>Increment Type</i>	Permite seleccionar el mode de respaldo incremental, con las siguientes opciones: NONE, INCREMENTAL, COMPLETE, y CUMULATIVE.
<i>Import File</i>	Archivo del Sistema Operativo específicamente para restaurar la información hacia la base de datos. (Especifica el requerido del parámetro File)
<i>Ignore Creation Errors</i>	Si se selecciona esta opción, el proceso de restauración no reportara el mensaje de "Object already exists" además de los errores cuando se creen los objetos en la base de datos.
<i>Hora del evento</i>	En caso de ser invitación o audiencia se especifican horas y minutos.
<i>Grants</i>	(1) Seleccionara esta opción si usted requiere los permisos de los objetos a respaldar. (2) Seleccionara esta opción si usted requiere que los permisos de los objetos sean restaurados

<i>Generate Log File</i>	Permite intercambiar la generación del archivo de log o salida de resultados en el respaldo.
<i>Full Database</i>	(1) Selección que genera el respaldo total de la Base de Datos (2) Especifica al proceso del modo total de la Base de Datos
<i>Fecha del evento</i>	Fecha del la invitación o audiencia
<i>Fecha del documento</i>	Fecha cuando se elaboro el documento
<i>Fecha de recepción</i>	Fecha cuando se recibió el oficio
<i>Fecha de despacho</i>	Fecha cuando se turna la gestión
<i>Fecha de descargo</i>	Fecha cuando el turnado da una respuesta
<i>Fecha de archivo</i>	Fecha de culminación de una gestión
<i>Export File</i>	Ingresar en este elemento el nombre de uso en el sistema operativo el cual contendrá los datos que desea exportar. ( Se requiere un valor específico y correcto del nombre de un archivo, con respecto al sistema operativo).
<i>Expediente</i>	Es la clave que se le asigna a cada institución de acuerdo a su organización jerárquica. Por ejemplo, a la Secretaría de Educación Pública se le asigna el numero 1, a la Dirección General de Educación Básica (D.G.E.B.) el 1.1, a la Dirección General de Higiene Escolar (D.G.H.E.) el 1.2, a la Dirección General No.1 que depende de la D.G.E.B. el 1.1.1, al Centro de Orientación Sicipedagógica que depende de la D.G.H.E. el 1.2.1, etc.
<i>Dependencia</i>	Nombre de la institución solicitante
<i>Defaults</i>	(1) Reincorpora las opciones por omisión definidas por el fabricante (2) Cuando es accionado pone todos los parámetros en su modo o estado de omisión.
<i>Database Management System (DBMS)</i>	Es un sistema de software que controla y manipula los datos, para eliminar redundancia y asegurarse de la integridad, consistencia y disponibilidad de los datos, entre otras características.

<b>Database</b>	(1) Permite seleccionar la base de datos con la cual se requiere trabajar para realizar el respaldo (2) Permite intercambiar la base de datos con la cual se esta trabajando actualmente.
<b>Data Manipulation Language (DML)</b>	Es una parte del Lenguaje de Consultas Estructuradas (SQL), el cual es responsable de realizar las operaciones de la manipulación de datos, tales como SELECT, DELETE, UPDATE e INSERT
<b>Data Definition Language (DDL)</b>	Es una parte del Lenguaje de Consultas Estructuradas (SQL), el cual consiste en llevar la responsabilidad de los comandos de creación/eliminación de objetos de la base de datos
<b>Constrains</b>	Esta opción permitirá respaldar las restricciones de las base de datos
<b>Compress when imported</b>	Permite realizar la compresión de datos mientras se realiza el respaldo
<b>Clasificación</b>	La clave que recibe un documento de acuerdo a su importancia se le llama clasificación, es decir, el oficio es mas o menos importante o normal o relevante, etc
<b>Buffer Sise</b>	Permite intercambiar el tamaño del Buffer para realizar los procesos de Fetch asignados al respaldo
<b>Browse</b>	(1) Este elemento permite seleccionar un nombre de archivo proveído por el sistema operativo para que almacene los datos que se requieran respaldar. (2) Se usa esta opción para poder seleccionar del sistema operativo el archivo que se va a restaurar en la Base de Datos
<b>Asunto</b>	Es un resumen del oficio.
<b>Aplication Program</b>	(1) Es un programa realizado por un usuario, que establece un mejor rendimiento para el trabajo de los usuarios. (2) Es un programa usado para conectar y comunicar una red con sus estaciones de trabajo.
<b>API</b>	Application Programing Interface

---

<i>American National Standards Institute (ANSI)</i>	Es una organización regida por las asociaciones de Manufactura de Equipo de Negocios y Computadoras para establecer estándares voluntarios dentro de la industria
<i>American National Standard Code For Information Interchange (ASCII)</i>	Códigos, desarrollados por ANSI para el intercambio de información entre sistemas de procesamiento de datos, sistemas de comunicación de datos y dispositivos o periféricos asociados. Los caracteres ASCII consisten en un conjunto de 7 bits por caracter, mas un bit para verificar la paridad.
<i>Advanced</i>	(1) Habilitara opciones avanzadas del respaldo de la base de datos (Habilitará el dialogo de Advanced Export Options) (2) Permite habilitar la edición de parámetros avanzados del proceso de restaurar
<i>Especificación del proceso</i>	Se refiere al detalle de lo que sucede dentro de un proceso, es decir, de cómo transforma los datos de entrada para producir los de salida.
<i>Paquete de datos</i>	Conjunto de datos que describen las características de un objeto o persona.
<i>Entidad externa o terminador</i>	Es la persona, grupo de personas, departamento, empresa, institución, otro sistema, etc, que está involucrada de alguna forma con el sistema.
<i>Diagrama de contexto</i>	Es el diagrama principal de un diagrama de flujo de datos, en él se muestran las entidades y flujos principales que entran y salen del sistema.
<i>Archivo de sólo lectura</i>	Quiere decir que este archivo jamás será actualizado, es decir, únicamente se extraerán datos de él.
<i>Archivo de sólo escritura</i>	Significa que únicamente se podrán guardar datos en él pero nunca extraerlos.
<i>Archivo externo</i>	Es un archivo que sirve de interfaz o conexión entre dos sistemas o subsistemas diferentes.
<i>DFD cero o de primer nivel</i>	En este DFD se muestran aquellos módulos o subsistemas principales, así como los flujos más relevantes o un conjunto de ellos generalizados en uno solo.
<i>Partición</i>	División de módulos de un DFD de acuerdo a lo especializado de la función que realiza.

<i>Top-Down</i>	De arriba a abajo.
<i>Modelo Entidad-Relación</i>	Es una herramienta gráfica que permite representar las entidades y sus atributos que participan en el procesamiento de la información del sistema.
<i>Atributos</i>	Características que definen o describen a una entidad.
<i>Ocurrencias, instancias o registros</i>	Conjunto de datos que definen todas las características de un elemento de una entidad.
<i>Identificador único</i>	Atributo o conjunto de atributos que identifican de manera única a una instancia de una entidad.
<i>Relación</i>	Evento bidireccional que representa la asociación entre dos entidades, o entre una entidad consigo misma.
<i>Grado o cardinalidad</i>	Se refiere a la cantidad de elementos que puede contener una relación en el sentido de destino.
<i>Matriz de relaciones</i>	Matriz en la cual se establecen las relaciones detectadas entre las entidades o archivos del sistema.
<i>Relación recursiva</i>	Es aquella relación que establece una entidad consigo misma.
<i>Normalizar</i>	Conformar las entidades de tal forma que desaparezca la redundancia de información.
<i>JOIN</i>	Intersección de consultas.
<i>CONSTRAINTS</i>	Restricciones aplicadas a la estructura de una entidad o tabla.

## BIBLIOGRAFÍA

---



Anderson, George, Berson, Alex. **SYBASE AND CLIENT/SERVER COMPUTING**. Edit. McGraw-Hill. United States of America, 1995.

Koch George. **MANUAL DE REFERENCIA ORACLE**. Edit. McGraw-Hill. México, 1994.

Yourdon, Edward. **ANÁLISIS ESTRUCTURADO MODERNO**. Edit. Prentice-Hall. México, 1993.

**ORACLE 7 SERVER ADMINISTRATOR'S GUIDE**. Versión 7.0. Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER APPLICATION DEVELOPER'S MANUAL**. Versión 7.0 Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER MESSAGES AND CODE MANUAL**. Versión 7.1. Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER SQL LANGUAGE REFERENCE MANUAL**. Versión 7.0 Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER SQL LANGUAGE QUICK REFERENCE**. Versión 7.0 Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER SQL LANGUAGE REFERENCE MANUAL**. Versión 7.0 Oracle Corporation. United States of America, 1992.

**ORACLE 7 SERVER UTILITIES USER'S GUIDE**. Versión 7.1 Oracle Corporation. United States of America, 1992.

**PRO\*C**. Versión 7.0 Oracle Corporation. United States of America, 1992.

**PL/SQL USER'S GUIDE AND REFERENCE**. Versión 2.0 Oracle Corporation. United States of America, 1995.

**PERSONAL ORACLE 7 INSTALLATION AND USER'S GUIDE**. Versión 7.1 for Windows. Oracle Corporation. United States of America, 1995.

**INTRODUCCIÓN A ORACLE. DISEÑO DE BASE DE DATOS RELACIONALES**. Oracle de México. México, 1995.

**GUÍA DE PROCEDIMIENTOS**. Manual de la Coordinación Administrativa. Secretaría de Hacienda y Crédito Público. México, 1993.