

32  
Lej



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

---

---

## FACULTAD DE INGENIERIA

Sistema de Información de la  
Administración Universitaria

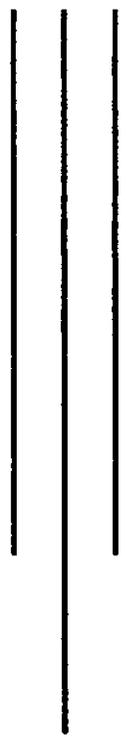
### T E S I S

Que para obtener el título de:

**INGENIERO EN COMPUTACION**

Presenta:

**Arturo Jallath Coria**



Director: Ing. Sebastián Poblano Ordoñez

México, D.F.

1999

TESIS CON  
FALLA DE ORIGEN

271996



Universidad Nacional  
Autónoma de México



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

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

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

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



A Rosa, mi querida esposa, por haber aceptado el reto de compartir su vida siempre a mi lado.

A la memoria de mi Padre, que donde quiera que te encuentres, estoy seguro estarás muy orgulloso de saber terminado este proyecto.

A mi Madre, por haberme forjado el carácter y alentarme siempre a seguir adelante

A mis hermanos, Salomón, Ma. Luisa y Ricardo, por su incondicional apoyo durante todas las etapas de mi vida.

A mis amigos, Javier, Julio, Isidro y Gerardo por los grandes momentos que juntos hemos compartido.

**A la Universidad Nacional Autónoma de México, a la Facultad de Ingeniería y al Patronato Universitario, por haberme dado los medios para enfrentar, desarrollar y proyectar la vida profesionalmente.**

**Al Ing. Sebastián Poblano, por su gran compromiso con la Universidad y entusiasta colaboración para la realización de este trabajo.**

# INDICE GENERAL

INTRODUCCION.....	1
<b>CAPITULO I ANTECEDENTES .....</b>	<b>5</b>
<b>1.1 PATRONATO UNIVERSITARIO DE LA UNAM .....</b>	<b>5</b>
1.1.1 <i>Tesorería</i> .....	8
1.1.2 <i>Dirección General de Control e Informática</i> .....	8
1.1.3 <i>Dirección General de Finanzas</i> .....	8
1.1.4 <i>Dirección General del Patrimonio</i> .....	9
1.1.5 <i>Contraloría Interna</i> .....	9
1.1.6 <i>Auditoría Interna</i> .....	9
1.2 <b>PROBLEMÁTICA CON LAS DEPENDENCIAS .....</b>	<b>11</b>
1.3 <b>OBJETIVO .....</b>	<b>14</b>
1.4 <b>ALCANCE .....</b>	<b>15</b>
1.5 <b>SOLUCIÓN.....</b>	<b>19</b>
<b>CAPITULO 2 INFRAESTRUCTURA TECNOLÓGICA.....</b>	<b>21</b>
<b>2.1 RED-UNAM.....</b>	<b>21</b>
2.1.1 <i>Historia</i> .....	22
2.1.2 <i>Red de Telecomunicaciones</i> .....	24
2.1.3 <i>Principales Características</i> .....	25
2.1.4 <i>Servicios</i> .....	27
<b>2.2 PROTOCOLOS DE COMUNICACIÓN .....</b>	<b>32</b>
2.2.1 <i>Protocolo TCP/IP</i> .....	32
2.2.2 <i>¿En qué consiste TCP/IP?</i> .....	33
2.2.3 <i>Direcciones IP</i> .....	34
2.2.4 <i>Principales Características</i> .....	37
<b>2.3 PLATAFORMAS DE HARDWARE.....</b>	<b>38</b>
2.3.1 <i>Procesadores CISC</i> .....	38
2.3.2 <i>Procesadores RISC</i> .....	39
2.3.3 <i>Arquitectura Multiprocesador</i> .....	40
2.3.4 <i>Multiprocesamiento Simétrico vs. Asimétrico</i> .....	41
2.3.5 <i>Memoria, almacenamiento y velocidad</i> .....	42
2.3.6 <i>Escalabilidad</i> .....	42
<b>2.4 SISTEMAS OPERATIVOS .....</b>	<b>45</b>
2.4.1 <i>Solaris</i> .....	45
2.4.2 <i>MS Windows NT Server</i> .....	48

2.5 MANEJADORES DE BASES DE DATOS (DBMS).....	51
2.5.1 SYBASE.....	51
2.5.2 MS SQL SERVER.....	52
2.6 SELECCIÓN DE PLATAFORMA DE DESARROLLO.....	56
<b>CAPITULO 3 ARQUITECTURA DE SOLUCION .....</b>	<b>60</b>
3.1 ARQUITECTURA CLIENTE/SERVIDOR.....	60
3.1.1 <i>Ventajas de la Arquitectura Cliente/Servidor</i> .....	61
3.1.2 <i>Desventajas de la Arquitectura Cliente/Servidor</i> .....	65
3.2 ALMACENES DE DATOS (DATA WAREHOUSING).....	67
3.2.1 <i>Manejo de Datos</i> .....	68
3.2.2 <i>Procesos en Data Warehousing</i> .....	70
3.2.3 <i>Consideraciones</i> .....	71
3.3 LENGUAJES DE PROGRAMACIÓN.....	77
3.3.1 <i>MS Visual Basic</i> .....	77
3.4 CONECTIVIDAD ABIERTA PARA BASES DE DATOS (ODBC).....	81
3.4.1 <i>Arquitectura ODBC</i> .....	83
3.4.2 <i>Visual Basic y ODBC</i> .....	84
3.5 HERRAMIENTAS ADICIONALES.....	86
3.5.1 <i>QELINK.VBX</i> .....	87
3.5.2 <i>VBALINK.DLL</i> .....	88
3.5.3 <i>FSSOCKET.VBX</i> .....	89
3.5.4 <i>PPRTR.DLL</i> .....	89
3.5.5 <i>GVBOX.VBX</i> .....	89
3.5.6 <i>VBSQL.VBX</i> .....	90
<b>CAPITULO 4 DESARROLLO DEL SISTEMA .....</b>	<b>91</b>
4.1 ANÁLISIS.....	91
4.1.1 <i>Planteamiento del Problema</i> .....	92
4.1.2 <i>Propuesta de Solución</i> .....	98
4.1.3 <i>Metodología</i> .....	106
4.1.3.1 <i>Modelo en Espiral</i> .....	107
4.1.4 <i>Flujo de Datos</i> .....	109
4.1.5 <i>Diagrama Entidad-Relación</i> .....	124
4.1.6 <i>Diccionario de Datos</i> .....	128
4.2 DISEÑO Y DESARROLLO.....	151
4.2.1 <i>Diseño Descendente</i> .....	152
4.2.2 <i>Desarrollo modular</i> .....	155

4.2.3 Diseño de la Base de Datos .....	156
4.2.3.1 Organización de la Base de Datos.....	159
4.2.3.2 Bases de Datos Relacionales.....	160
4.2.3.3 Normalización.....	161
4.2.4 Construcción de la Base de Datos .....	164
4.2.5 Diseño y Construcción del Esquema de Seguridad.....	186
4.2.6 Diseño y Construcción de la Interfaz del Usuario .....	192
4.2.6.1 Interfaz de Menús.....	193
4.2.7 Programación y Desarrollo.....	197
4.2.7.1 Diagrama estructural.....	197
4.3 PRUEBAS DEL SISTEMA .....	207
4.4 IMPLANTACIÓN.....	212
4.4.1 Proceso de Instalación.....	213
4.4.2 Tipos de Ayuda.....	216
4.4.3 Monitoreo del Sistema.....	219
4.5 MANTENIMIENTO .....	223
<b>CONCLUSIONES.....</b>	<b>227</b>
<b>BIBLIOGRAFÍA.....</b>	<b>231</b>
<b>ANEXO A: INDICE DE ILUSTRACIONES .....</b>	<b>233</b>

## INTRODUCCION

El crecimiento acelerado que han presentado en los últimos años las redes de área local (LAN -Local Area Network) y las redes de área amplia (WAN -Wide Area Network), prácticamente ha hecho desaparecer el esquema de concentrar todas las operaciones de cómputo en un equipo central, optándose por operar las aplicaciones críticas en forma distribuida utilizando diferentes servidores de la red. Asimismo, con el paso del tiempo y gracias a los sistemas de cómputo y las nuevas tecnologías, las empresas cuentan actualmente con una gran cantidad de datos históricos sobre las operaciones diarias de su negocio. Por ejemplo, se tienen almacenados datos sobre las transacciones comerciales, cuentas por cobrar, inventarios, cheques, facturas y demás operaciones realizadas durante años.

En las grandes corporaciones estos archivos históricos pueden ser de grandes dimensiones, sin embargo, dado que no existe una organización y clasificación de la información, este conjunto de datos suele tener un uso netamente histórico, a pesar de que, como es obvio, podría extraerse una considerable cantidad de indicadores de gran utilidad para el grupo de directores o ejecutivos encargados de tomar las decisiones del rumbo de una empresa.

Un ejemplo de las grandes corporaciones que existen en México, aunque de naturaleza no lucrativa, lo constituye la Universidad Nacional Autónoma de México (UNAM), institución educativa, que tiene como objetivos sustantivos: impartir educación superior para formar profesionistas, investigadores, profesores universitarios y técnicos útiles

a la sociedad; organizar y realizar investigaciones principalmente acerca de las condiciones y problemas nacionales y extender con la mayor amplitud posible, los beneficios de la cultura.

En apoyo a estos objetivos, la UNAM cuenta con una instancia de carácter administrativo denominada Patronato Universitario, cuya principal labor consiste en brindar servicios a las autoridades y dependencias para su adecuada operación administrativa, financiera y contable, abarcando todas las facultades, escuelas, colegios, centros, institutos, direcciones generales y unidades multidisciplinarias que conforman la máxima casa de estudios.

Dicha instancia es la que concentra y almacena, entre otros rubros, datos y registros presupuestales, contables, financieros y de inventario, y a su vez brinda a todas y cada una de las dependencias y subdependencias universitarias información procesada, emitiendo para ello grandes volúmenes de documentación impresa, que es distribuida mensualmente.

Conociendo el papel preponderante que para la Institución tienen las funciones de Patronato y a fin de aprovechar el cúmulo de información que dicha dependencia genera, surge la necesidad de crear un sistema de manejo de datos por computadora, con la intención de brindar un medio de información ágil y oportuno que facilite la comunicación entre el Patronato y las diferentes dependencias universitarias.

Dicho sistema, objetivo principal de este trabajo, deberá ofrecer a todos los usuarios la consulta, en pantalla, de información actualizada en los rubros financieros, presupuestales y de gestión administrativa, brindando

además la posibilidad de imprimir en forma total o parcial los datos consultados, así como su exportación a medios magnéticos.

Por otra parte, haciendo uso de la red de cómputo y comunicaciones de la UNAM (Red UNAM), el alcance del sistema deberá abarcar no sólo las áreas que se encuentran dentro de Ciudad Universitaria, Distrito Federal y Zona Metropolitana, sino que también incluir el servicio hacia aquellas dependencias que se ubican en el interior de la República, e incluso incorporar al ámbito internacional, pudiendo ser utilizado por aquellas dependencias universitarias que se encuentran más allá de las fronteras del país. Todo lo anterior con el fin de brindar a la comunidad universitaria una herramienta que permita disponer de información veraz y oportuna para la toma de decisiones.

Para fines de explicación, el presente trabajo se dividió en cuatro capítulos, cada uno de los cuales está dedicado a explicar a detalle los elementos utilizados para el desarrollo del sistema.

De esta manera, en el primer capítulo, se habla de los antecedentes del sistema; del papel que juega el Patronato Universitario dentro del quehacer de la UNAM; de la manera en que se ha venido operando en materia de la distribución de información a las dependencias; de los problemas detectados, y, finalmente, se expone el procedimiento para enfrentar la problemática observada mediante una propuesta de solución.

En el segundo Capítulo, se desarrollan los temas relacionados con la infraestructura tecnológica sobre la cual se pretende poner en operación el sistema; se muestra un panorama general de lo que es la red de

comunicaciones de la UNAM; se habla de protocolos de comunicación, sistemas operativos de red y manejadores de bases de datos relacionales, así como también de la plataforma de *hardware* (equipo de cómputo) que mejor se adapta al proyecto.

En el tercer capítulo, se define la arquitectura de *software* (programas de cómputo) a utilizar para el desarrollo del sistema, analizando las ventajas y desventajas que presenta; además, se habla del acceso remoto a bases de datos y de las herramientas de programación necesarias para la aplicación.

El cuarto capítulo, describe la metodología utilizada durante el desarrollo, especificando a detalle cada una de las etapas que conforman la construcción del sistema (análisis, diseño, construcción, pruebas, implantación y mantenimiento), tanto a nivel cliente como a nivel servidor.

Finalmente, en el último apartado, se exponen las conclusiones obtenidas a través del desarrollo de la aplicación; de los alcances y ventajas del sistema, así como de las posibles adecuaciones que se sugieren con base en la observación de su desempeño.

# CAPITULO 1 ANTECEDENTES

## 1.1 Patronato Universitario de la UNAM

Con el fin de comprender los orígenes del sistema, resulta conveniente ubicar en primer lugar la instancia responsable de la integración y distribución de la información: el Patronato Universitario, área que al enfrentarse cotidianamente a los problemas, reflejó la necesidad de cambio y se convirtió en el factor que detonó la propuesta para una nueva forma de trabajo.

Para entender claramente qué es y cuáles son las actividades que lleva a cabo el Patronato Universitario, se comenzará por explicar el lugar que ocupa dentro de la estructura orgánica de la UNAM. Esquemáticamente, la Universidad Nacional Autónoma de México está organizada como se muestra a continuación:

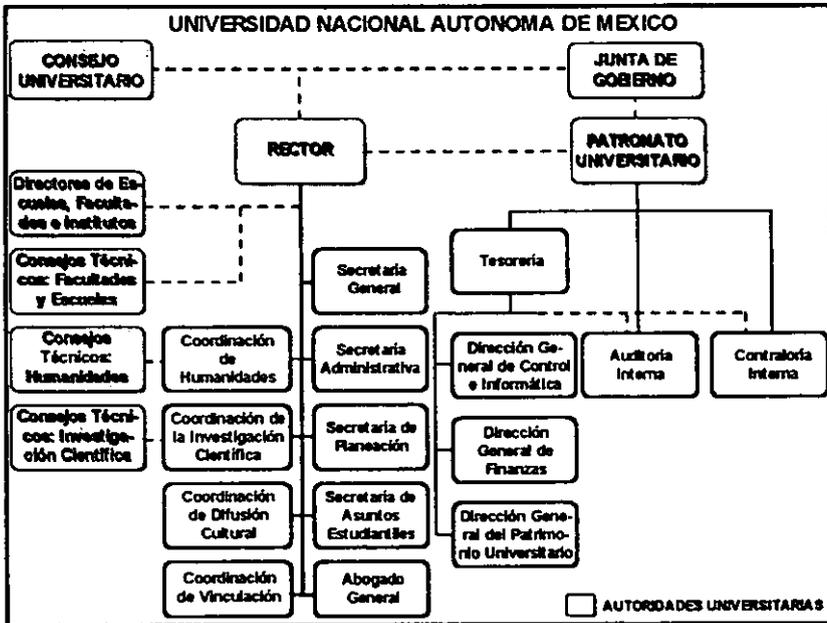


Figura 1.1 Organigrama General de la Universidad Nacional Autónoma de México.

Como se observa en la figura anterior, el Patronato Universitario forma parte de grupo de autoridades de la Institución y tiene como principales objetivos:

- Contribuir a la realización de los fines de la Universidad brindando apoyo y servicios a las autoridades y dependencias universitarias para su adecuada operación administrativa, financiera y contable; para el buen manejo de sus recursos humanos y materiales, así como para el uso adecuado de los bienes muebles e inmuebles.
- Administrar eficientemente el patrimonio de la Universidad, constituido por los bienes muebles e inmuebles, así como por los recursos financieros de la inversión y operación; ejercer adecuadamente el presupuesto de ingresos y egresos, y cuidar el adecuado registro, control e información sobre el patrimonio y las operaciones económicas que realice la UNAM.

Dentro de sus principales funciones se encuentran:

- Administrar el patrimonio universitario y sus recursos ordinarios, así como los extraordinarios que por cualquier concepto pudieran allegarse.
- Formular el presupuesto general anual de ingresos y egresos, así como las modificaciones que haya que introducir durante cada ejercicio, oyendo para ello a la Comisión de Presupuestos del Consejo Universitario y al Rector.
- Presentar al Consejo Universitario la cuenta respectiva, dentro de los tres primeros meses a la fecha en que concluya un ejercicio; previa revisión de la misma que practique un contador público independiente, designado con antelación por el propio Consejo Universitario.
- Gestionar el mayor incremento del patrimonio universitario, así como el

aumento de los ingresos de la Institución.

La estructura interna del Patronato Universitario está conformada por la Tesorería, las Direcciones Generales de Finanzas, Patrimonio y Control e Informática, así como por la Contraloría y Auditoría Internas, auxiliadas todas ellas por la Secretaría Administrativa del Patronato Universitario. Las subdependencias que conforman el Patronato Universitario son las siguientes:

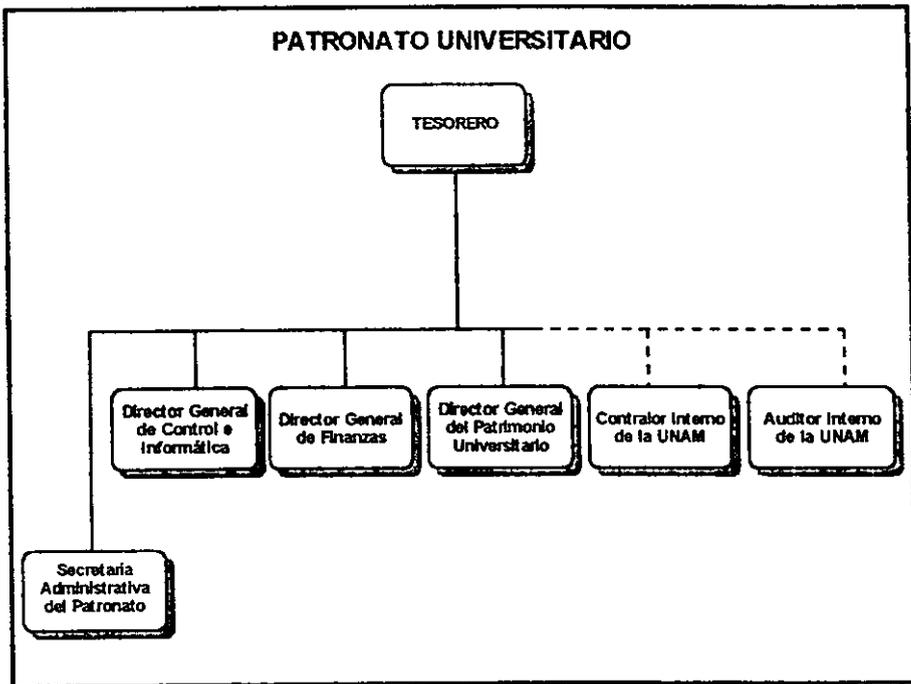


Figura 1.2 Estructura interna del Patronato Universitario.

En los párrafos que se presentan a continuación se describe brevemente la personalidad de cada una de las subdependencias del Patronato Universitario, así como sus principales funciones.

### **1.1.1 Tesorería**

Es el órgano responsable de la administración tanto del ejercicio y control financiero, como del patrimonio universitario.

Su objetivo es coordinar y ejecutar la administración financiera y patrimonial de la UNAM, dentro de un marco de eficiente control contable, presupuestario y patrimonial. Sus funciones son planear, estructurar y establecer los programas financieros y administrativos necesarios para el logro de los objetivos del Patronato y de la Institución.

### **1.1.2 Dirección General de Control e Informática**

Se encarga de coordinar y supervisar el registro, aplicación e información de las operaciones financieras y presupuestales de la UNAM; la formulación oportuna y correcta de la Cuenta Anual, los Estados Financieros mensuales y la información financiera complementaria; el mejoramiento de las estructuras y sistemas de operación, así como el suministro y control de los servicios de apoyo administrativo del Patronato.

### **1.1.3 Dirección General de Finanzas**

Tiene la responsabilidad de administrar y controlar los fondos de la Institución, provenientes del subsidio y de los ingresos propios ordinarios y extraordinarios, así como de promover la mayor utilización y aprovechamiento de los mismos. Dirige y controla el plan financiero anual; prepara y vigila el plan de arbitrios, los programas de flujo de fondo y verifica que los compromisos financieros de la Institución sean oportunamente liquidados.

#### **1.1.4 Dirección General del Patrimonio**

Su función es administrar el patrimonio de la UNAM, constituido por sus bienes muebles e inmuebles tangibles e intangibles; promueve el incremento patrimonial, la eficiencia en su utilización y en el servicio a la comunidad universitaria.

#### **1.1.5 Contraloría Interna**

Tiene como objetivo salvaguardar el patrimonio de la UNAM a través del fortalecimiento del marco de control interno y legislación institucional, así como del desarrollo y/o adopción de sistemas y procedimientos de registro e información que promuevan la eficiencia de operación.

#### **1.1.6 Auditoría Interna**

Es el órgano de vigilancia de la UNAM con amplias facultades para realizar las auditorías informáticas, financieras y operacionales en todas las dependencias de la Institución. Su objetivo es vigilar el uso adecuado del patrimonio de la UNAM, de acuerdo con las políticas, disposiciones y legislación universitaria.

Como se puede apreciar, la subdependencia del Patronato Universitario encargada de brindar información al resto de las dependencias universitarias, es la Dirección General de Control e Informática, por lo que a partir de este momento la explicación se centrará en sus funciones.

La Dirección General de Control e Informática, a través de las Unidades de Proceso Administrativo (UPA's), es la encargada de canalizar la información, registro y control de los recursos financieros para cubrir los trámites y gestiones inherentes a las operaciones que se generan dentro de las diferentes dependencias universitarias.

En apoyo a esta labor, las Unidades de Proceso Administrativo hacen uso del "Sistema Contable-Presupuestal", herramienta computarizada que permite llevar el registro contable de todas las operaciones financieras que día a día realizan las dependencias universitarias sobre su correspondiente presupuesto anual. La información que arroja diariamente dicho sistema, se une con aquella que generan las dependencias centralizadoras y, una vez integrada, se procesa y se almacena en medios magnéticos, para que, a su vez, se generen los Estados Financieros de toda la Institución.

## 1.2 Problemática con las Dependencias

Durante los últimos años, el Patronato Universitario era el encargado de consolidar la información generada en las diferentes dependencias y, tras de su proceso, era el responsable de emitir mensualmente la impresión de reportes de Estados Financieros de la Institución y los correspondientes a los Compromisos Previos y Gastos a Reserva de Comprobar; así como los Estados de Cuenta Presupuestales, de Ingresos Extraordinarios y de Inversión por Ingresos Extraordinarios.

El proceso para la generación de dichos reportes, se llevaba a cabo a fin de mes en un equipo Unisys A9, que con el paso del tiempo se actualizó a un Unisys A7. No obstante este avance, derivado del tipo de tecnología utilizada, el proceso de afectación, consolidación y preparación de información tardaba más de 8 horas.

Una vez que la información estaba lista, se iniciaba la tarea de impresión misma que se prolongaba por más de una jornada de trabajo. Posteriormente, ya generados los reportes, los documentos tenían que ser cortados y separados por bloques de mil hojas para ser entregados a la Contaduría General, donde finalmente se agrupaban por subdependencia, hasta completar los correspondientes a cada una de las 253 con las que cuenta la UNAM.

En suma, la totalidad del proceso mencionado se llevaba a cabo en un lapso de 4 a 5 días hábiles, lo que provocaba que las dependencias, en el mejor de los casos, recibieran su información al cabo de 35 a 45 días de haber realizado sus movimientos en las Unidades de Proceso del Patronato. Aunado a esto habría que sumar el tiempo en que el responsable administrativo de cada dependencia tardaba en desplazarse a Ciudad Universitaria para poder realizar la conciliación de sus saldos.



central que afectaban el presupuesto local, situación que era desconocida por la dependencia hasta que recibía el reporte impreso de Patronato con el desfase en tiempo detallado anteriormente; lo cual ocasionaba graves problemas de control para consolidar saldos.

Ante esto, la Dirección General de Control e Informática, a través del área de Contaduría General, se vio en la necesidad de establecer, días y horarios específicos, para realizar la conciliación de saldos con cada una de las 253 subdependencias, con el fin de darles atención oportuna y resolver todas sus dudas.

Aún más, considerando que la UNAM tiene dependencias distribuidas a todo lo largo y ancho no sólo de la Ciudad de México, sino también en el interior del país y dos en el extranjero, el ciclo de entrega de reportes y conciliación de saldos para las dependencias foráneas se llevaba a cabo en un lapso significativamente mayor que aquel realizado para las dependencias cercanas.

Esta forma de operación impuesta desde hace 20 años, derivó en procedimientos obsoletos que, lejos de favorecer el desarrollo institucional, se convirtieron en un severo obstáculo para el flujo administrativo. Dicha situación aunada al impulso generado por la integración a la UNAM de nueva tecnología de cómputo y comunicaciones, hicieron evidente la necesidad de plantear una nueva forma de realizar el trabajo más rápida y eficiente, a fin de satisfacer los requerimientos planteados por la comunidad universitaria.

Por otro lado, también se observó que el Patronato y las otras autoridades universitarias se enfrentaban a la imposibilidad de enviar, a través de la vía electrónica, oficios y circulares de importancia al resto de las dependencias, lo que representaba atrasos en la difusión de nuevas políticas y/o estrategias

institucionales. No obstante, que el uso del correo electrónico había auxiliado en esta tarea, subsistía el problema de que la gran mayoría de los Secretarios Administrativos no poseían clave para este servicio y, aún más, algunos de ellos carecían de la experiencia para el manejo de archivos de imágenes.

Por último, se detectó la imposibilidad de distribuir a todas las dependencias universitarias, de manera expedita, manuales de procedimientos, fascículos informativos y aplicaciones de uso general, lo que ocasionaba la necesidad de imprimir y distribuir grandes volúmenes de información y/o la grabación de cientos de disquetes, con la inversión de tiempo que esto representaba.

### **1.3 Objetivo**

Proporcionar a las 143 dependencias pertenecientes a la Universidad Nacional Autónoma de México, un sistema que permita difundir información en forma oportuna y eficiente en los rubros financieros, presupuestales y de gestión, además de aspectos relativos a organización, procedimientos y comunicados con carácter oficial, así como la transferencia de aplicaciones vía electrónica; accedando para ello, las bases de datos de Patronato desde una computadora personal (a través de Red-UNAM), con la finalidad de brindar a la comunidad universitaria, una herramienta que permita disponer de información oportuna para la toma de decisiones.

## 1.4 Alcance

Inicialmente el sistema conocido bajo el nombre de Sistema Integral de Consulta (SIC), fue ideado para dar solución a los requerimientos de Patronato, sin embargo, después de sus primeros diez meses de uso en la Universidad, en septiembre de 1997 se llevó a cabo una reunión de trabajo entre la Secretaría Administrativa de la UNAM y el Patronato Universitario, representada cada una por su titular y funcionarios de primer orden.

De conformidad a la agenda establecida, se discutieron y analizaron opciones, y se tomaron decisiones y acuerdos sobre diversos puntos con carácter administrativo, tendientes a una mejor administración que apoyara, de manera eficaz y oportuna, al propósito académico de esta Institución.

Entre las decisiones tomadas en materia de sistemas informáticos, ambas autoridades acordaron: *"Utilizar como eje y base del conjunto de sistemas y/o aplicaciones el Sistema Integral de Consulta, que pasará a tener un alcance institucional y no sólo del Patronato"*.

Derivado de lo anterior, el Sistema Integral de Consulta se reestructuró, dando lugar a un nuevo desarrollo denominado: *Sistema de Información de la Administración Universitaria (SIAU)*, a través del cual se pretendió constituir un medio de divulgación hacia las dependencias de información con carácter oficial, proveniente, en una primera etapa, de la Secretaría Administrativa y del Patronato, y al corto plazo, de otras instancias involucradas en la administración universitaria, constituyéndose como el único medio oficial de información y gestión administrativa para las dependencias de la UNAM.

Por tal motivo el sistema evolucionó de su idea original, para abarcar más

servicios, de tal manera que fue necesario dividirlo en dos apartados: el correspondiente a Patronato Universitario y el de la Secretaría Administrativa; los módulos propuestos son los siguientes:

◆ **Secretaría Administrativa**

- *Normatividad para la Emisión de Oficios*
- *Funciones y Estructura*
- *Directorio Telefónico*

◆ **Patronato Universitario**

- *Información Presupuestal y de Gestión*
- *Funciones y estructura*
- *Directorio Telefónico*

◆ **Circulares y Oficios**

◆ **Transferencia de Aplicaciones**

◆ **Mensajes del Sistema**

Sin lugar a dudas, el módulo más completo y que dio origen al sistema, es el referente a *Información Presupuestal y de Gestión*, mismo que se integra por los siguientes submódulos:

◆ **Información Presupuestal y de Gestión**

- *Compromisos Previos*

- *Gastos a Reserva de Comprobar*
- *Consulta Presupuestal*
- *Ejercido Presupuestal*
- *Apoyos*
- *Transferencias Presupuestales*
- *Estado de Cuenta de Ingresos Extraordinarios*
- *Inversión por Ingresos Extraordinarios*
- *Folios UPA's*
- *Pólizas*
- *Boletos de Avión*
- *Proyectos PAPIIT*
- *Depósitos CONACYT*

Finalmente, el módulo de Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (P.A.P.I.I.T.), tendiente a proveer de información a investigadores, también se descompone en nueve submódulos, los cuales se enuncian a continuación:

◆ **Proyectos PAPIIT**

- *Resumen de Proyectos*

- *Presupuesto*
- *Super Cray (Ministraciones Entregadas)*
- *Movimientos Presupuestales*
- *Honorarios Proyectos*
- *Becarios Proyectos*
- *Boletos de Avión*
- *Otros Servicios*
- *Compromisos Previos*

## 1.5 Solución

Para dar solución a la problemática observada, se planteó la posibilidad de desarrollar un sistema bajo la tecnología Cliente/Servidor, que contara con un servidor de bases de datos sobre un equipo basado en la tecnología RISC, utilizando a Windows NT Server como sistema operativo de red, SQL Server como manejador de base de datos relacional y TCP/IP como protocolo de comunicación. Por lo que se refiere al cliente, se utilizaría la *interface* gráfica Windows en cualquiera de sus versiones (3.1, 3.11, 95 o 98) con protocolo de comunicación TCP/IP y Visual Basic como lenguaje de programación.

La programación del sistema, dentro del servidor se realizó con base en procedimientos almacenados (*Stored Procedures*), mismos que serían ejecutados cuando así lo requiriera el usuario a través de una petición del cliente; por lo que la programación Visual Basic desarrollada en el cliente consistiría únicamente en la presentación y modelado de resultados hacia el usuario final.

Mediante el uso de este sistema la Universidad estaría en posibilidad de ofrecer a sus dependencias y subdependencias un servicio de información que les permitiría realizar consultas, impresión y exportación de información financiera y de gestión administrativa, accedendo directamente desde su computadora personal a las bases de datos de Patronato Universitario.

Los beneficios que un sistema de esta naturaleza generaría de manera inmediata serían, entre otros:

- Poner fin a la impresión de más de 20,000 hojas de reportes mensuales, que se realizó durante los últimos 20 años.

- Entregar oportunamente información actualizada al cierre de operaciones del día anterior.
- Permitir la consulta del estado de trámites gestionados en las Unidades de Proceso Administrativo.
- Disminuir considerablemente las conciliaciones con dependencias por parte de la Contaduría General.
- Proporcionar información histórica de los trámites realizados durante el año en curso y años anteriores.
- Difundir oportunamente oficios y circulares emitidos por el Patronato Universitario y la Secretaría Administrativa dirigidos a las dependencias
- Brindar información de alcance institucional a las dependencias centralizadoras y autoridades universitarias.
- Distribuir ágilmente aplicaciones y grandes documentos a la comunidad universitaria.

Todo lo anteriormente descrito, permitirá a las dependencias contar con información actual y oportuna para el mejor aprovechamiento de los recursos disponibles y la toma de decisiones que favorezcan al desarrollo institucional.

## CAPITULO 2 INFRAESTRUCTURA TECNOLOGICA

Como se mencionó anteriormente, en el presente capítulo se abordan temas relacionados con la infraestructura tecnológica sobre la cual se puso en operación el sistema; se muestra un panorama general de lo que es la red de comunicaciones de la UNAM; se habla de protocolos de comunicación, plataformas de *hardware*, sistemas operativos de red y manejadores de bases de datos relacionales, así como de la plataforma elegida para el desarrollo del sistema.

### 2.1 Red-UNAM

Para las universidades mexicanas Internet representa una poderosa herramienta que pone a disposición de los estudiantes un cúmulo de información reciente, estimulando su interés hacia la investigación y la comunicación con jóvenes universitarios e instituciones de alto nivel, buscando concretar una formación educativa activa e integral.

Consciente de la tendencia mundial hacia el desarrollo de nuevas tecnologías, la máxima casa de estudios ha creado uno de los proyectos más ambiciosos de toda Latinoamérica: la Red Integral de Telecomunicaciones de la Universidad Nacional Autónoma de México, conocida como REDUNAM.

REDUNAM es el proyecto desarrollado para la transmisión de información (datos, video y voz) entre las facultades, institutos, centros de difusión, coordinaciones y demás dependencias que conforman a la UNAM.

### 2.1.1 Historia

El final de los años 60's y el principio de la década de los 70's marcaron para la Universidad Nacional Autónoma de México, la etapa de inicio de las comunicaciones telefónicas y de datos. Es en ese período cuando se realizan las primeras conexiones de teletipos hacia una computadora central, utilizando líneas telefónicas de cobre de la recién instalada red telefónica dentro de la Institución. Rápidamente esta tecnología es usada al interior de la UNAM y difundida al exterior, con lo cual se efectúan una gran cantidad y diversidad de conexiones; de terminales de caracteres, graficación e impresión, incluyendo la interconexión de estaciones de trabajo remotas, todas ellas a través de líneas telefónicas.

A partir de la segunda mitad de la década de los 80's, surge en la UNAM el interés por la evolución en las comunicaciones. Así, en 1987, la UNAM establece la primera conexión, mediante enlaces telefónicos, a la Red Académica BITNET desde la Ciudad Universitaria hasta el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM) y de ahí hasta San Antonio, Texas en los EUA.

Posteriormente, la UNAM intentó consolidar su enlace a esa red internacional mediante la computadora IBM 4381, la cual sirvió como residencia del correo electrónico y otros servicios de BITNET; dentro de ese proceso se inició la conexión de terminales IBM con emulación 3270, estableciéndose además un enlace con la Red TELEPAC de la Secretaría de Comunicaciones y Transportes (SCT), bajo la finalidad, nunca lograda, de brindar este servicio en el ámbito nacional.

No fue sino hasta 1989, cuando la UNAM, a través del Instituto de Astronomía, establece un convenio de enlace a la red de la NSF (*National Science Foundation*)

en EUA, el cual se realizó utilizando el satélite mexicano Morelos II entre el Instituto de Astronomía en la UNAM y el UCAR (*University Corporation for Atmospheric Research*)-NCAR (*National Center for Atmospheric Research*) con residencia en Boulder, Colorado. Asimismo, se llevó a cabo el primer enlace para conectar las redes de área local, entre el Instituto de Astronomía y la Dirección General de Servicios de Cómputo Académico (DGSCA), utilizando enlaces de fibra óptica. A partir de ese momento se inició dentro de la UNAM una revolución en las comunicaciones.

Este proceso, aunado a la adquisición masiva de computadoras personales y su interconexión e intercomunicación en redes de área local, principalmente en las dependencias del subsistema de la investigación científica, permitió desarrollar la infraestructura de comunicaciones con fibra óptica, y establecer más enlaces satelitales hacia Cuernavaca, Morelos, y San Pedro Mártir, en Ensenada, Baja California Norte, a la par del primer enlace de microondas de alta velocidad entre la Torre II de Humanidades y la Dirección General de Servicios de Cómputo Académico, en la Ciudad de México.

Con esto, se estableció definitivamente el final de la era del teleproceso en UNAM, para dar paso a las redes de computadoras y sus enlaces a través de fibra óptica.

En 1990, la UNAM fue la primera Institución en Latinoamérica que se incorporó a la red mundial Internet, la cual enlaza a millones de máquinas y decenas de millones de usuarios en todo el mundo. Dicha red es producto de un proyecto del Gobierno de los Estados Unidos que data de 1970. En sus primeras etapas (como parte de un programa de investigación militar de ARPA (*Defense Advanced Research Projects Agency*)) se logra demostrar la viabilidad de las comunicaciones entre computadoras, por medio de la conmutación de paquetes; lo cual creó la red

ARPANET, que enlazó varias decenas de sitios en una red nacional dedicada a la comunidad de investigación en computación. El concepto de conmutación de paquetes se extendió en muy pocos años para incluir redes satelitales y redes basadas en radio.

El ininterrumpido desarrollo de Internet, que no tiene límite a la fecha, contempla como elemento fundamental el diseño de una arquitectura para comunicar redes, tal que permite la coexistencia de paquetes de comunicación de diferentes tipos bajo el protocolo TCP/IP; el cual, dada su funcionalidad y posibilidad de adaptación a los requerimientos que se van presentando, se mantiene como estándar en la actualidad.

### **2.1.2 Red de Telecomunicaciones**

A finales de 1989, la UNAM estableció un ambicioso proyecto que debía sustituir sus antiguos conmutadores para renovar totalmente el sistema telefónico interno, con base en los estándares más modernos y con capacidad de crecer conforme a las necesidades de la Institución.

Para este proyecto, que constituye la parte fundamental del Programa Institucional de Informática en la Dirección General de Servicios de Cómputo Académico, se creó la Dirección de Telecomunicaciones Digitales cuyo objetivo sería la creación de la Red Integral de Telecomunicaciones de la UNAM, la cual debería ser capaz de transmitir indistintamente datos e imágenes entre las dependencias universitarias independientemente de su ubicación geográfica.

Ante la necesidad de integrar los diferentes servicios y recursos de cómputo como soporte de desarrollo eficiente para la investigación y la docencia, en 1990 surge el Laboratorio de REDUNAM, como un espacio para el estudio, análisis de

comunicación, topologías de redes, protocolos y servicios, entre otras cuestiones.

### 2.1.3 Principales Características

La Red Integral de Telecomunicaciones de la Universidad Nacional Autónoma de México se inaugura oficialmente en 1992. La figura 2.1 muestra de manera general cómo se encuentra constituida dicha red.

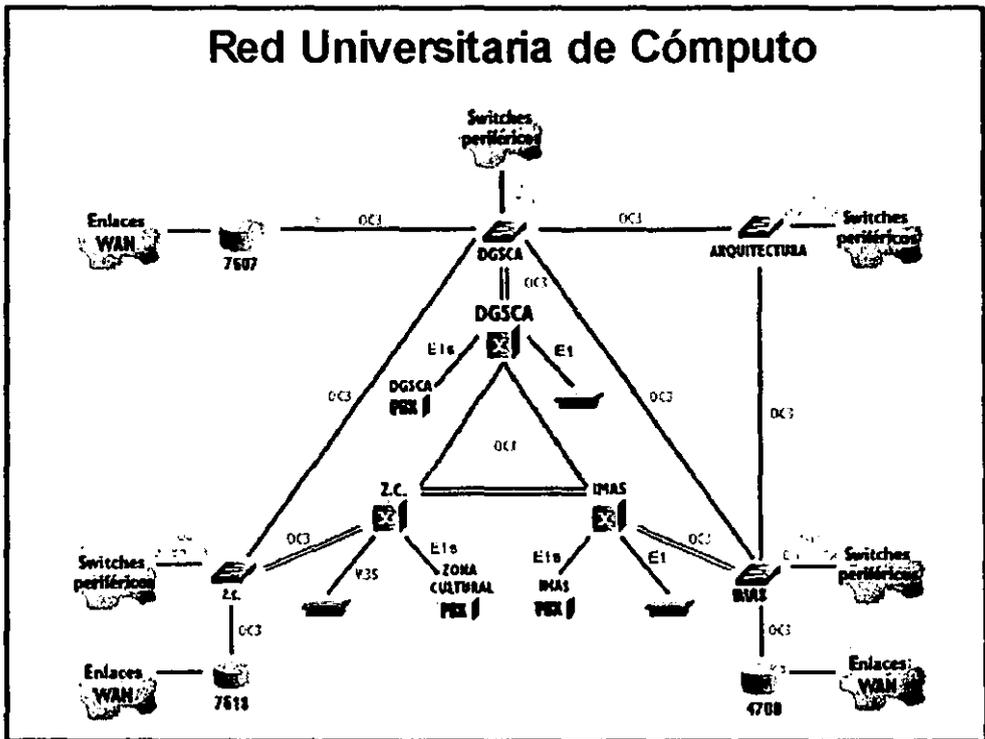


Figura 2.1 Red Integral de Telecomunicaciones de la Universidad Nacional Autónoma de México.

Entre las principales características que destacan hoy en día en la Red Integral de Telecomunicaciones, se pueden mencionar las siguientes:

- Permite la transmisión indistinta de datos y video, mediante sistemas digitales basados en las más modernas normas internacionales.
- Integra a la red las principales instalaciones de la Universidad. Esto significa, que considerando los niveles de bachillerato, licenciatura, posgrado e investigación, alrededor del 95% de sus miembros se encuentran operando en instalaciones cubiertas por la red; abarcando varias regiones del país desde Ensenada, Baja California; hasta Puerto Morelos, en Quintana Roo.
- La red está conformado por 32 nodos operacionales de telefonía enlazados entre sí mediante fibra óptica, enlaces satelitales y de microondas.
- Disfruta de una infraestructura instalada para 13,000 servicios telefónicos alimentados por 2,400 troncales digitales conectadas vía fibra óptica con las centrales telefónicas públicas.
- Posee una red complementaria de respaldo de más de 1000 servicios, basada en telefonía celular y 17 líneas telefónicas directas.
- Cuenta con más de 600 redes locales en ocho regiones del país. La red enlaza cerca de 10,000 computadoras de la UNAM entre sí y alrededor de 15 millones de computadoras en el resto del mundo.

Actualmente, se lleva a cabo la instalación de 3,500 nuevos servicios que se sumarán a la infraestructura existente, y la modernización de los equipos de datos. Asimismo, se realiza una ardua labor para integrar al servicio a las principales instalaciones de la UNAM en el ámbito metropolitano y nacional; a la par de atender los campus de Hermosillo, Ensenada, Martínez de la Torre, Cuernavaca, Juriquilla y Morelia.

También en este rubro, como resultado de una labor ininterrumpida, existen sistemas de tarificación y renovación de los servicios telefónicos de la UNAM con las tecnologías más modernas y eficientes, a fin de brindar a la Institución el soporte necesario para el mejoramiento de sus actividades sustantivas.

Por lo que respecta a la infraestructura tecnológica, la UNAM cuenta con sistemas Unisys, Hewlett Packard y Digital para sus procesos administrativos y de atención a usuarios; varios sistemas UNIX para los cientos de proyectos de investigación y recopilación de información que se desarrollan en los distintos institutos y centros de difusión, y diversos equipos Mainframes para el desarrollo de la cultura informática entre la población estudiantil.

La máquina más conocida de la UNAM es la supercomputadora CRAY Y-MP4/464 en la que se desarrollan interesantes proyectos de Astronomía, Química, Física, investigación en materiales y mecánica de fluidos, entre otros. Esta es la única computadora de su tipo en toda América Latina y entre sus usuarios se distinguen investigadores de varias instituciones nacionales y extranjeras.

Además, en la Coordinación de Servicios de Cómputo se buscan constantemente nuevos servicios que ofrecer, siempre con la finalidad de brindar al usuario una mejor opción para la adquisición de información nacional e internacional.

#### **2.1.4 Servicios**

Los servicios que ofrece REDUNAM son, en primer lugar, aquellos que permiten la comunicación entre los diversos *hosts* (entendiéndose por éstos, las máquinas que fungen como servidores y como clientes).

Entre los servicios brindados por REDUNAM destacan los que se muestran en el siguiente cuadro:

Servicio	Descripción
Ruteo	Comunicación entre Computadoras
DNS	Conversión entre direcciones numéricas y nombres lógicos
NOC	Monitoreo de la Red
NIC	Distribución de Información y atención a usuarios

- **Ruteo**

El ruteo se encarga de que los archivos no se extravíen en alguna parte del mundo o lleguen a la persona equivocada. Sin este servicio no sería posible la comunicación con los *hosts* de otras redes. También es posible limitar el acceso a usuarios indeseables.

- ***DNS (Domain Name Service)***

Se encarga de resolver la conversión entre las direcciones numéricas y los nombres lógicos de las máquinas, de forma tal que el usuario pueda utilizar el nombre: servidor.unam.mx para comunicarse con su servidor de correo, en lugar de la dirección: 132.248.10.1

- ***NOC (Network Operation Center)***

Se encargan de monitorear el comportamiento de la red, conservar su óptima

operación, así como darle el mantenimiento necesario.

- **NIC (Network Information Center)**

Se encarga de distribuir la información de los servicios de red, soportarlos dentro de la REDUNAM y dar la capacitación necesaria a los usuarios sobre sus beneficios.

Con base en los servicios mencionados anteriormente, el usuario puede hacer uso de otros, tales como los que se enlistan en el siguiente cuadro:

<b>Servicio</b>	<b>Descripción</b>
Correo Electrónico	Comunicación entre usuarios a través de mensajes.
<i>Gopher</i>	Menús jerárquicos para búsqueda de información.
<i>Archie</i>	Búsqueda de archivos.
Telnet	Sesiones remotas a grandes computadores.
<i>World Wide Web</i>	Tecnología multimedia para presentar información.
Almacenamiento Masivo	Almacenamiento de grandes volúmenes de información.
Supercomputadora CRAY	Investigación científica y tecnológica

- **Correo electrónico**

Es uno de los servicios de mayor demanda, el cual permite la comunicación entre usuarios de cualquier parte del mundo a través de la colocación de mensajes de texto en el buzón electrónico del destinatario.

- ***Gopher***

Se trata de menús jerárquicos que permiten buscar información en REDUNAM e Internet a través de conexiones transparentes al usuario.

- ***Archie***

Búsqueda electrónica de archivos. Basta con definir el tema a buscar y en pocos minutos se obtendrá una lista que contiene los nombres y direcciones de las máquinas que contienen archivos referentes al tema y que son de dominio público. Dichos archivos pueden ser copiados por el usuario vía FTP Anónimo.

- ***Telnet***

Sesiones remotas a grandes computadoras para aprovechar sus altas capacidades de cálculo y otros recursos que difícilmente se encuentran en máquinas menores.

- ***World Wide Web***

Sin duda uno de los servicios más interesantes y completos que se pueden encontrar en Internet. Este servicio aprovecha la tecnología de multimedia para ofrecer una presentación de la información mucho más interesante mediante el uso de imágenes, texto y sonido desde cualquier punto de Internet a la

computadora del usuario.

- **Almacenamiento Masivo de Información**

Si no se cuenta con dispositivos suficientes para conservar grandes cantidades de información, se puede utilizar el servicio de almacenamiento masivo de información.

- **Supercomputadora CRAY**

Para llevar a cabo la investigación científica y el desarrollo tecnológico de una manera competitiva se requieren servicios de cómputo modernos. Con el propósito de proporcionar este tipo de servicio a la comunidad universitaria y productiva del país, la Universidad Nacional Autónoma de México puso en operación, en 1991, una Supercomputadora CRAY Y-MP/43 2.

## **2.2 Protocolos de Comunicación**

Dentro del mercado nacional e internacional existen diferentes tipos de protocolos, entre los cuales los de mayor uso en empresas pequeñas y medianas, así como en los grandes corporativos, son los siguientes:

- TCP/IP
- IPX/SPX
- NetBeui
- DLC

Sin lugar a dudas, debido al gran auge y crecimiento que ha tenido Internet en los últimos años, el protocolo que se perfila como estándar mundial es el TCP/IP, constituyéndose también como estándar dentro de Red UNAM. Por tal motivo la aplicación que se desarrolló en el presente trabajo, tomó como base el uso de este protocolo.

### **2.2.1 Protocolo TCP/IP**

TCP/IP (*Transmission Control Protocol/Internet Protocol*) es un protocolo diseñado con la finalidad de establecer conexiones entre diferentes redes.

A comienzos de los años ochenta comenzó a ponerse de manifiesto la necesidad de establecer conexiones entre computadoras de naturaleza diferente, considerando una escala global. TCP/IP se desarrolló fundamentalmente, gracias a la investigación realizada en la DARPA, dando lugar a un concepto de conexión inter-red denominado ARPANET, el cual evolucionó posteriormente hasta

desembocar en la actual INTERNET-RED (Internet).

El propósito de TCP/IP es proporcionar los protocolos necesarios de transmisión de información independientemente de la red o equipo utilizado. Dada la flexibilidad que caracteriza un protocolo de esa naturaleza, TCP/IP ha sido elevado casi de forma inmediata al dominio de la Internet, aplicándose en organizaciones que precisaban establecer conexiones entre distintos tipos de LAN.

### **2.2.2 ¿En qué consiste TCP/IP?**

El protocolo TCP/IP define una red de intercambio por paquetes, lo que significa que la información se transmite entre las computadoras en partes: una parte está conformada por los paquetes que contienen una “carga” de datos y, la otra, por la información de cabecera que identifica el envío, constituida por la dirección de destino y los códigos de corrección de errores. Si bien, algunas órdenes requieren tan sólo un único paquete, las transmisiones de archivos de gran extensión se dividen en múltiples paquetes, debiendo incluir una secuencia de números indicativos del orden según el cual los paquetes se deben re-ensamblar en el sistema de destino.

Una de las características más importantes del protocolo TCP/IP es su esquema de direccionamiento de red, el cual proporciona una forma de direccionar diferentes redes, así como los nodos de las mismas. Este esquema es extensivo, dando cabida a millones de posibles direcciones en una escala global. De hecho, las direcciones Internet son registradas y asignadas por el NIC (*Network Information Center*), a fin de evitar conflictos aunque no es esencial registrar las direcciones, si tan sólo se trata de establecer una conexión entre redes en un ámbito reducido.

En realidad, TCP/IP está compuesto por un conjunto de diferentes protocolos. Cada

uno de ellos cae dentro de un esquema de estratos que se compara a menudo con el modelo *Open System Interconnection* (OSI). A continuación se describe cada uno de estos protocolos.

**TCP (*Transmission Control Protocol*).** El protocolo TCP corresponde al estrato de sesión y transporte del modelo OSI. TCP recibe información de las aplicaciones que operan en niveles superiores en la pila de protocolos, y es responsable del empaquetado y la transmisión a través de la red. TCP pasa a IP los paquetes que crea.

**IP (*Internet Protocol*).** El protocolo IP corresponde al estrato Red en el modelo OSI. Este se encarga de crear los paquetes de información, y es el responsable de añadir las direcciones IP de origen y destino.

Algo que IP no puede proporcionar es la garantía de que los paquetes llegarán a su destino, y en el orden adecuado. Por tal motivo, TCP añade información a cada uno de los paquetes que este módulo crea, en la que incluye datos de identificación y ubicación. Si IP pierde un paquete, TCP es el responsable de determinar cuál es el paquete perdido, y de volver a enviarlo.

### **2.2.3 Direcciones IP**

La dirección *Internet Protocol* (IP) de un nodo es una dirección lógica, que es independiente de la dirección física asignada a la tarjeta de red por el fabricante de la misma. La dirección IP es también independiente de la configuración de la red. La dirección IP tiene la misma forma, no importando el tipo de red que se usa. Este formato único es un valor numérico de 4 bytes (32bits), que sirve para identificar, tanto a la red, como al nodo de la misma. Cada dirección IP debe ser única y

constar de cuatro números decimales, separados por puntos.

Como se mencionó anteriormente, las direcciones IP no dependen de ningún tipo de red en concreto. Ello significa que los paquetes pueden atravesar diferentes tipos de redes. En cada uno de los tipos de red, el protocolo TCP/IP se encarga de asignar la dirección IP a un nodo físico, para poder realizar el tránsito. Los paquetes contienen las direcciones IP del remitente, lo cual permite que el destinatario pueda reclamar los datos, en caso de ser necesario. La dirección IP permite identificar tanto la red como el nodo que corresponde al remitente.

Una forma de definir direcciones IP consiste en unirse a la comunidad DARPA Internet, contactando con el NIC (*Network Information Center*) mencionado en párrafos anteriores. Si no se está interesado en registrar oficialmente una red, se pueden elegir números arbitrarios que se rijan por el esquema de direccionamiento IP (cuatro números decimales, separados por puntos). Sin embargo, es aconsejable que se utilice el esquema DARPA Internet, por si en el futuro se precisase entrar en contacto con algún punto externo.

La dirección IP de 4 bytes está dividida en dos partes: una de ellas, sirve para identificar a la red, mientras que la otra, identifica a la computadora (nodo). De esta manera, la parte que corresponde a la red, debe ser siempre la misma para todos los nodos que la conforman, en tanto que la parte que identifica al nodo, ha de ser única y diferente para cada equipo conectado a dicha red.

Existen diversos esquemas para la asignación de estos números:

### **1. Esquema de direccionamiento Clase A.**

El primer byte es la dirección de la red, y los tres últimos la del nodo. El rango para el primer byte es de 1 a 126, lo cual permite un total de 126 redes diferentes, con 16 millones de nodos cada una.

### **2. Esquema de direccionamiento Clase B.**

Es el esquema utilizado dentro de Red UNAM, donde los dos primeros bytes identifican la red, y los dos últimos, el nodo. El rango para el primer byte es de 128 a 191, quedando el segundo para identificaciones más finas. Esto permite un total de 16,000 redes con 65,000 nodos.

### **3. Esquema de direccionamiento Clase C.**

Los tres primeros bytes identifican a la red, y el último es la dirección del nodo, lo cual permite un total de 2 millones de redes diferentes, con 254 nodos cada una.

Algunas direcciones están reservadas, y el bit de orden más elevado en cada dirección requiere un número identificador especial.

Es posible dividir una red en varias subredes, para así poder hacer uso de múltiples tipos de medios, o bien para reducir la congestión, disminuyendo el número de puestos de trabajo de la red. Cuando existen subredes, las direcciones IP constan de una dirección de red, una dirección de subred y una dirección de nodo. La parte de dirección que corresponde al nodo en la dirección IP, se divide para incluir las direcciones de la subred y del nodo. Para las redes externas, la red total seguirá apareciendo como un único conjunto, con una única dirección de red.

## 2.2.4 Principales Características

Entre las principales características que posee el protocolo TCP/IP, se pueden mencionar las siguientes:

- **Independencia de la tecnología de redes.**

Ya que no está basado sobre el *hardware* de ningún vendedor en especial. Los protocolos de TCP/IP definen la unidad de transmisión como Datagrama y especifican cómo transmitir datagramas en una red.

- **Interconexión universal.**

Una red de TCP/IP permite la comunicación de cualquier pareja de computadoras que forman parte de la red. A cada computadora se le asigna una dirección, que es reconocida por todos los equipos integrantes de la red. Cada datagrama lleva la dirección de origen y destino. Las computadoras intermedias usan la dirección de destino para tomar decisiones de ruteo.

- **Acuse de recibo entre las computadoras que dialogan.**

Los protocolos de TCP/IP proporcionan un acuse de recibo sólo entre las dos computadoras que dialogan y no entre máquinas sucesivas que forman parte de la trayectoria.

## 2.3 Plataformas de Hardware

Antes de abordar el tema referente a los equipos (*hardware*), es necesario exponer un conjunto de conceptos que precisan las características que a este respecto se deben cumplir. Además, dada la dimensión del sistema a generar, la plataforma a utilizar requiere de ser robusta, confiable, eficiente, escalable y con un alto rendimiento, de tal manera que pueda soportar en un inicio entre 400 y 600 usuarios simultáneamente.

Así pues, derivado de una investigación comparativa entre la tecnología CISC y la tecnología RISC, se obtuvieron los siguientes resultados:

### 2.3.1 Procesadores CISC

Hasta principios de los 80's, se creía que la mejor forma de crear un procesador era incrementando su complejidad, como parte de un esfuerzo por emular fielmente las instrucciones proporcionadas por un lenguaje de alto nivel. A este tipo de procesadores se les conoce como procesadores CISC (*Complex Instruction Set Computer*). Un procesador CISC se caracteriza por tener un conjunto de instrucciones amplio, una gran variedad de modos de direccionamiento y el uso extensivo de microcódigo.

En teoría, un procesador CISC parece una buena opción ya que ejecuta directamente en *hardware* las instrucciones de *software*. Sin embargo, el análisis de cientos de programas en código objeto reveló los siguientes hechos:

- Los compiladores muy frecuentemente no hacen uso completo de las instrucciones proporcionadas por el procesador.

- Un diseño complejo lleva a implantaciones ineficientes de las instrucciones.
- El uso de las instrucciones simples como *LOAD*, *STORE*, *ADD*, *SUB* y *BRANCH* son mucho más frecuentes que el de las instrucciones más complejas.

### 2.3.2 Procesadores RISC.

Con estos antecedentes surge la filosofía RISC (*Reduced Instruction Set Computer*). Los principales puntos de diseño de un procesador RISC son:

- **Conjunto reducido de instrucciones.** Se emplean únicamente las instrucciones que estadísticamente son más frecuentes. Estas instrucciones se implantan directamente en el *hardware*, eliminando el microcódigo que les correspondería. Idealmente, se busca que se ejecute una o más instrucciones por cada ciclo de reloj.
- **No existen modos de direccionamiento.** Sólo existen instrucciones para cargar registros o escribir el contenido de un registro hacia la memoria (*LOAD/STORE*). Con esto se simplifica el diseño del procesador.
- **Tienen un número grande de registros de CPU.**
- Como resultado de los puntos anteriores, **la lógica de control del CPU se reduce de un 50% del área del chip a un 6%**, lo que a su vez permite integrar más funciones en el chip (un caché, por ejemplo).

Ahora bien, un punto interesante es el hecho de que un procesador RISC necesariamente requiere un compilador inteligente. Otro punto que destaca es que las instrucciones sencillas de un lenguaje de alto nivel son traducidas directamente a instrucciones en ensamblador, en tanto que construcciones más complejas se operan a través del uso de subrutinas. Esto implica que un procesador RISC debe estar extremadamente bien diseñado en su manejo de salto a subrutinas.

Siguiendo la inercia del mercado (que de ninguna manera se puede subestimar) y el poderío económico/tecnológico de Intel, la balanza podría inclinarse en favor de los procesadores CISC. Sin embargo, desde el punto de vista puramente técnico RISC es la mejor opción, fuera del mundo de las computadoras personales.

### **2.3.3 Arquitectura Multiprocesador**

Las aplicaciones de Base de Datos son típicamente intensas en manejo de entrada/salida, por tal motivo aquellas máquinas que soportan mejor este comportamiento son las que obtienen resultados superiores. Las arquitecturas de *hardware* que permiten la utilización de múltiples procesadores comúnmente otorgan un rendimiento superior a aquellas que no lo poseen.

Los reportes de *benchmark TPC-A y TPC-B* generados por el *Transaction Processing Performance Council*, confirman este comportamiento de manera muy consistente, ya que casi siempre los mejores rendimientos corresponden a máquinas que soportan más de un procesador. Adicionalmente, las plataformas multiproceso son fácilmente escalables. Esto se considera una ventaja, sobre todo, si se desconoce la velocidad con la que van a crecer las necesidades de los sistemas de información que corren en estos equipos.

### **2.3.4 Multiprocesamiento Simétrico vs. Asimétrico**

Las arquitecturas multiprocesador difieren entre sí. Una de las más importantes consideraciones de rendimiento es aquella que se refiere a la asimetría o simetría. Una implantación simétrica es aquella en que ninguno de los procesadores es “maestro” sobre los otros procesadores del sistema. Cada procesador, para realizar su propio trabajo, funciona independiente sin necesidad de los servicios de otro diferente.

En la alternativa asimétrica, un procesador llamado “maestro” trabaja como una especie de filtro para controlar algunos aspectos del proceso (por ejemplo: manejo de entrada/salida). En el peor de los casos, donde el sistema está orientado a entrada/salida, el sistema como un todo, no puede funcionar más rápido que aquel procesador que está haciendo el filtrado; en este tipo de esquemas, agregar más procesadores no mejorará significativamente el rendimiento del sistema.

En la alternativa simétrica, cualquier procesador puede ejecutar, independientemente de los otros, todas las funciones que requiere.

Ahora bien, si las principales preocupaciones al seleccionar un equipo son el rendimiento y la capacidad de expansión, como en el caso de aplicación motivo del presente trabajo, seleccionar un sistema con arquitectura de multiproceso simétrica resulta la mejor opción.

#### **¿Cuántos Procesadores?**

Como se puede notar, es posible que incrementando el número de procesadores permita mejorar el rendimiento, sin embargo, esta medida depende de la habilidad de los equipos para manejar varios de ellos, ya que al aumentar el número de

procesadores aumenta también la complejidad para que trabajen juntos.

### **2.3.5 Memoria, almacenamiento y velocidad.**

La memoria principal es otro punto importante a considerar, ya que el mantener la mayor parte de la aplicación en memoria aumenta el rendimiento de los sistemas, por lo que, aquellos sistemas que ofrecen más memoria pueden ser más ventajosos.

Memorias rápidas como la *caché*, provocan que los procesadores estén ocupados más tiempo. Por lo que entre mayor es este tipo de memoria mayor es la posibilidad de tener información ahí, permitiendo que el procesador trabaje a velocidades de *caché*, en lugar de funcionar a velocidades de memoria real. El *caché* reduce la necesidad de entrada/salida del sistema, con lo que logra una velocidad 15 a 17 nanosegundos, mientras que las memorias reales oscilan en el rango de los 60 a 80 nanosegundos.

En algunas aplicaciones, la máxima cantidad de espacio en disco puede ser un factor importante, así como también lo es la velocidad del BUS.

### **2.3.6 Escalabilidad**

La escalabilidad se refiere a que el equipo pueda crecer a medida que crecen las necesidades de cómputo. Este crecimiento puede involucrar a la capacidad del disco, la memoria principal o la memoria *caché*, pero además puede consistir en agregar más procesadores a una misma máquina o incluso tener más máquinas interconectadas.

En resumen, las características más deseables en un equipo de cómputo son:

- Que el equipo sea escalable. Tanto en memoria y disco como en procesadores.

- Que acepte multiprocesamiento y que sea simétrico.
- Que optimice especialmente los procesos de Entrada/Salida.

Además, por cuestiones de desempeño, manejabilidad y seguridad, es muy recomendable que los equipos cuenten como mínimo con: 128 MB de memoria, 2 GB en disco, unidad de cinta DAT (*Data Audio Tape*) y unidad de CD-ROM.

Considerando lo anterior, se realizó una investigación para conocer las marcas de los equipos que mejor se adaptaban a las necesidades del proyecto. Los resultados obtenidos, arrojaron las siguientes conclusiones: las estaciones de trabajo *SUN Microsystems* en su modelo *SPARCserverTM 1000E* y los servidores *Digital ALPHA* en su modelo *2100 4/200* son la mejor opción, debido a que cumplen con todos los requerimientos técnicos y expectativas necesarios para el desarrollo del sistema.

En el siguiente cuadro se muestran las características principales y los precios de ambos equipos:

<b>Descripción</b>	<b>Sparc 1000E</b>	<b>Digital 2100</b>
Procesadores	1 a 60 MHz.	1 a 190 MHz.
Memoria RAM	128 MB	192 MB
Disco Duro	2.1 GB	2.1 GB
Unidad de Cinta	DAT 4mm 5 GB	DAT 4mm 8 GB
Unidad de CD	Interna 644 MB	Interna 675 MB
<b>Costo Total (USD):</b>	<b>\$ 49,238.00</b>	<b>\$ 24,418.00</b>

La base instalada de servidores para la academia y la investigación dentro de la UNAM está conformada por equipos *Sun*, sin embargo, se encontraron tres grandes desventajas de este equipo: la velocidad de su procesador trabaja a una tercera parte de la DIGITAL; tiene 64 Mb menos de RAM y, por último, acepta un crecimiento a 2 procesadores como máximo, mientras que su oponente permite crecer a 4, incluso a velocidades de 400 y 600 Mhz.

Otro factor muy importante recayó sobre el costo total del equipo, ya que al ser el modelo 2100 el primero que colocó la empresa Digital dentro de la UNAM, su precio de introducción representó un poco menos de la mitad que el modelo de *Sun*.

Derivado de lo anterior, se llegó a la conclusión de que la mejor opción es la que presenta el equipo Digital, pues ofrece la garantía de no requerir el cambio de equipo durante un período aproximado de 5 a 10 años, ya que bastará con agregar más memoria, discos duros y procesadores sobre la configuración original para mantener el nivel de servicio conforme vaya creciendo el número de usuarios y el volumen de información manejada.

## 2.4 Sistemas Operativos

Dada la magnitud del sistema a desarrollar, se requería de un sistema operativo robusto; capaz de soportar la operación de un gran número de usuarios en forma simultánea; compatible con diferentes tipos de clientes (Windows 3.1, 3.11, 95 y NT); con capacidad de correr en diferentes plataformas; que maneje adecuadamente la memoria y el multiprocesamiento simétrico, y, sobre todo, que se encuentre dentro de los estándares que solicita el Comité Asesor de Cómputo para el desarrollo de sistemas institucionales, tanto académicos como administrativos de la UNAM.

La investigación efectuada reportó que Solaris (una versión Unix de *SUN Microsystems*) y Windows NT de la compañía Microsoft reunían los requisitos y el potencial para el desarrollo requerido, de tal manera, que se procedió a realizar un análisis de ambos, obteniéndose las siguientes conclusiones:

### 2.4.1 Solaris

Por lo que a UNIX se refiere, se identificó como un Sistema Operativo Multiusuario y Multitarea, el cual está conformado por varios módulos o componentes principales, como son: el Núcleo, el *Shell*, el Sistema de Archivos y las Aplicaciones.

Las características más importantes que presenta UNIX, son:

- **Herramientas de *Software***

Habla de la modularidad del sistema, con la filosofía de que los problemas y las aplicaciones pueden ser creadas mediante interconexión de unos cuantos

módulos. En el desarrollo de aplicaciones, se manejan subrutinas empaquetadas o combinadas para formar módulos más satisfactorios, esto es "Reutilización de Software".

- **Portabilidad**

Puede ser reinstalado en prácticamente cualquier tipo de máquina, desde una PC hasta un Mainframe. Generalmente se reconoce que el Sistema UNIX proporciona el entorno adecuado para permitir el fácil traslado de aplicaciones desde Microcomputadoras a Macrocomputadoras, entre arquitecturas diferentes y en especial de los diferentes tipos de versiones de este Sistema.

- **Flexibilidad**

Se refiere a la fácil adaptabilidad a aplicaciones divergentes entre sí.

- **Potencia**

Su potencia se basa en la sintaxis de órdenes claras y concisas, que permiten realizar, a la vez, varias cosas rápidas y sencillas.

- **Multiusuario y Multitarea**

El sistema UNIX es un entorno multitarea de tiempo compartido. Está diseñado para manejar sin esfuerzo las necesidades múltiples y simultáneas de un usuario. También es un entorno multiusuario que soporta las operaciones de más de una persona a la vez.

- **Orientación a Red**

Las versiones modernas del Sistema UNIX están organizadas para un uso de red fácil y funcional. Esto por las herramientas de comunicación internas del sistema, la aceptación de diversos dispositivos de bajo nivel.

Por lo que respecta a deficiencias de este sistema, se pueden enumerar las siguientes:

- Una de las principales comparaciones, y en la que reside una gran desventaja de UNIX sobre MS DOS y MS OS/2, es el tamaño y la complejidad (aunque esto ofrece otras opciones).
- Los comandos UNIX basados en la filosofía "hacer una cosa bien" contienen un gran número de opciones y controles, por lo que se ha perdido el concepto original. La utilización de módulos se ha dejado para acrecentar dichas opciones y este hecho perjudica en gran medida la conceptualización del usuario.
- El UNIX está orientado principalmente a terminales "Basadas en Caracteres", requiriendo un *software* especial para el uso de gráficos. Esto debido a que el UNIX fue creado para línea de comandos y no a ventanas o ambientes gráficos, mostrando un pequeño atraso en este aspecto. Aunque en la actualidad se cuenta con el *Open Desktop UNIX* de Sta. Cruz Operation, el UNIX gráfico (SCO UNIX), su instalación es complementaria, lo que se traduce en costos adicionales para el proyecto.
- Por otra parte, el sistema operativo UNIX va en contra de la tendencia de realizar los sistemas operativos "invisibles" al usuario. La filosofía de UNIX

se basa en la idea de que este Sistema fue creado para expertos en su operación, por lo que el usuario debía conocerlo a fondo tanto en su funcionamiento como en su manejo. Por su parte MS-DOS, OS/2, Macintosh Finder, Windows NT, etc., auxiliados de otras herramientas gráficas se sobreponen para ocultar el sistema operativo a la vista del usuario.

- La inconsistencia y confusión de algunas de sus órdenes es algo que se ha trabajado, sin llegar a resultados satisfactorios.
- El costo de la licencia de server para Solaris 2.4 es de \$1,155.00 dólares.

#### **2.4.2 MS Windows NT Server**

Windows NT es un sistema operativo diseñado para operaciones de 32 bits. Esto le permite sacar ventaja de los microprocesadores más avanzados, como los Intel 80386 y 80486, así como de las computadoras con tecnología RISC.

Los microprocesadores antiguos utilizaban conjuntos de instrucciones de 16 e incluso de 8 bits, ahora los microprocesadores de 32 bits pueden trabajar con cantidades numéricas, direcciones de memoria e instrucciones de mayor tamaño sin necesidad de dividir las operaciones en bloques más pequeños para que puedan tener cabida en el microprocesador. El rendimiento general, es decir, la combinación del rendimiento del microprocesador, transferencia de datos y acceso a memoria es, en definitiva, lo que este sistema operativo ha venido a mejorar sensiblemente.

La multitarea significa que el sistema operativo puede realizar varias tareas al mismo tiempo. En este esquema, la tarea o usuario que se identifique como preferente, puede interrumpir la secuencia determinada si fuese necesario, sin tener

que esperar obligatoriamente a que la fila termine.

Conforme aumenta la velocidad de proceso, las actividades relacionadas con el *hardware*, tales como el acceso al disco, pueden resultar tremendamente lentas. Cuando un sistema sin derecho preferente accede al disco, el microprocesador ha de esperar hasta que tenga lugar efectivamente el acceso mecánico al disco, malgastando en definitiva ciclos de procesamiento. En Windows NT existe la posibilidad de concurrencia de diversas tareas, de manera que si, por ejemplo, una de ellas queda parcialmente parada como consecuencia de un lento acceso al disco, el microprocesador puede concentrar su atención en otras tareas. Básicamente, lo que se consigue es que no se malgasten ciclos de procesamiento. Puede verse cada tarea individual como un corredor de velocidad. Si decae el ritmo de uno de ellos, los otros no frenan su marcha. El beneficio de esto es que los usuarios pueden realizar otras tareas mientras tiene lugar un acceso al disco en segundo plano, o bien, se prepara un trabajo de impresión.

La protección de memoria asegura que los diversos programas van a ser ejecutados cada uno en su parte individual de memoria, no pudiendo contaminar las áreas correspondientes a otras aplicaciones. En el caso de que una determinada aplicación quede bloqueada, las aplicaciones restantes, así como el sistema operativo permanecerán inalterados, dando así la oportunidad al usuario de dar fin a su trabajo en la forma correcta.

El microprocesamiento simétrico es una característica exclusiva de Windows NT, que permite sacar todo el partido de los procesadores múltiples. Si bien es cierto que la vida de los sistemas multiproceso es relativamente corta, NT es el primer sistema operativo que está preparado para realizar un uso eficaz de este tipo de tecnología. Los sistemas operativos anteriores se limitaban a asignar tareas

específicas a cada microprocesador individual, como por ejemplo, para la entrada/salida en una red local.

El **multiprocesamiento asimétrico**, en donde un microprocesador se dedica exclusivamente a una tarea específica, da lugar a que un éste permanezca inactivo en cuanto finaliza su tarea. En el **multiprocesamiento simétrico**, el sistema operativo puede asignar diferentes tareas a un mismo microprocesador; de esta manera, si uno de ellos termina su trabajo antes que otro, el sistema operativo podrá ocuparlo en otra actividad. El multiprocesamiento simétrico es bastante más difícil de instituir, pero ofrece un rendimiento considerablemente superior.

Las características de trabajo en red local con las que cuenta Windows NT permiten compartir archivos de un sistema con cualquier usuario en un entorno de red, así como conectarse a directorios compartidos de otros sistemas. Las computadoras donde se ejecuta Windows para Grupos de Trabajo pueden aprovechar las ventajas del trabajo en red. Además, Windows NT viene equipado con el *software* y los controladores necesarios para establecer las conexiones con otros tipos de sistemas operativos, como es el caso de UNIX o las grandes computadoras IBM.

El producto Windows NT Server es una versión mejorada de Windows NT que cuenta con características sofisticadas de servidor de archivos para entornos de red de grandes dimensiones. Incluye algunas características adicionales relacionadas con la protección de datos, como es el caso de la duplicación automática de datos a discos secundarios.

El costo de la licencia de server para Windows NT Server 4.0 a precio académico es de \$3,865.00 pesos m.n., aproximadamente \$415.00 dólares.

## 2.5 Manejadores de Bases de Datos (DBMS)

Una vez identificados los posibles sistemas operativos a utilizar, se requería también de un manejador de base de datos que se acoplara al sistema operativo a utilizar; que fuera robusto y manejara altos niveles de seguridad, distribución de bases de datos, mecanismos de respaldo y espejeo de información, entre otras características. De tal manera que después de la investigación respectiva, las opciones a elegir basadas en los sistemas operativos antes descritos, se redujeron a dos, que son:

- Sybase
- MS SQL Server

Para este caso, también se procedió a realizar un análisis de ambas tecnologías, así como la realización de algunas pruebas, obteniéndose las siguientes observaciones.

### 2.5.1 SYBASE

Sybase es una base de datos relacional diseñada para aplicaciones críticas y en línea, que requieren un alto rendimiento escalable; el servidor fuerza la integridad, la disponibilidad y las características distribuidas avanzadas. Sybase emplea una arquitectura *multithreaded* (multitareas) en la que la base de datos, más que el sistema operativo, es la que programa y controla las tareas para obtener una máxima eficiencia. Los *stored procedures* (procedimientos almacenados) y *triggers* (procedimientos autoejecutables) convierten a Sybase en un servidor programable. Las rutinas de mantenimiento de la base de datos son realizadas mientras que el sistema está totalmente operando. Sybase también se protege contra fallas del sistema con su propio mecanismo de tolerancia a fallas implantadas en el *software*. La interoperabilidad de Sybase permite a cualquier aplicación cliente trabajar

cooperativamente con cualquier fuente de datos a través de diferentes plataformas de *hardware*, red local y sistema operativo. El API (*Application Program Interface*) del *Open Client* de Sybase es un método bastante simple y flexible para acceder una gran variedad de fuentes de datos. El *Open Server* de Sybase permite a cualquier fuente de datos o servicio de cómputo responder a las peticiones de los clientes y comportarse como si fuera un servidor inteligente. Los procedimientos remotos de Sybase RPC's (*Remote Procedure Calls*) son un mecanismo de mensajes para la comunicación entre cualquier cliente y el servidor. Los precios para Sybase SQL Server en su versión 10.0.2 se pueden ver en la siguiente tabla:

Producto	Versión	Usuarios	Precio (USD)
Sybase SQL Server	10.0.2	1	\$3,750
		2 a 8	\$9,600
		9 a 16	\$16,560
		17 a 32	\$27,760

### 2.5.2 MS SQL SERVER

El Microsoft SQL Server es un Sistema Manejador de Bases de Datos Relacional (*RDBMS – Relational Data Base Management System*) inteligente, basado en la arquitectura cliente/servidor, que soporta aplicaciones de procesamiento de transacciones en línea y brinda apoyo en la toma de decisiones a través de múltiples ambientes de red. Este software combina el poder, la escalabilidad y manejabilidad del sistema operativo Windows NT.

La seguridad y garantía del procesamiento de transacciones en SQL Server lo hacen ideal para las más exigentes aplicaciones de misión crítica. Las capacidades avanzadas de SQL Server tales como: *triggers*, procedimientos almacenados, reglas, *defaults* y su tecnología de llamado a procedimiento remoto (RPC), hacen de SQL Server la herramienta ideal para el desarrollo de aplicaciones dorsales (*back-ends*) en estaciones de trabajo, minicomputadoras y mainframes. Su arquitectura diseñada para el alto rendimiento, multitarea, asegura un rendimiento consistente y estable que se mantiene aún cuando el número de usuarios aumenta. Los puentes SQL Server (*gateways*), también proveen la tecnología clave para la integración de microcomputadoras con minicomputadoras y mainframes de larga escala.

En 1993 se dio a conocer una versión especial de SQL Server expresamente diseñada para el sistema operativo Windows NT de Microsoft. Apoyándose fundamentalmente en las capacidades avanzadas de nueva generación, basadas en la arquitectura cliente/servidor del Windows NT, el SQL Server para Windows NT ofrece un buen rendimiento, seguridad y escalabilidad, características que lo hacen especialmente adecuado para grandes corporaciones.

Básicamente, para trabajar el sistema operativo Windows NT se requiere un mínimo de 32 MB en RAM, pero si se utiliza la replicación de datos y los cursores incluidas en la versión 6.5 de SQL, entonces se recomienda tener 64 MB.

Para obtener un mayor rendimiento es conveniente utilizar SQL Server sobre una máquina con procesador RISC, con un mínimo de 64 MB de memoria, y para aumentar su rendimiento, en el caso de tener bases de datos muy grandes con un número de usuarios bastante elevado, es recomendable tener 128 MB.

Una de las grandes ventajas de SQL Server para Windows NT es que al igual que la mayoría de los sistemas desarrollados por la compañía Microsoft, se ejecutan sobre ambientes gráficos que ayudan tanto al cliente para el desarrollo de aplicaciones, como al administrador para tener mejor controlado el sistema.

### **Características Principales de SQL Server**

- Administración remota de los recursos del sistema.
- Arquitectura de 32 bits.
- Capacidad para generación automática de copias de seguridad.
- Ejecución de procedimientos almacenados en forma remota.
- Un robusto soporte de tolerancia a fallas.
- El reforzamiento centralizado de la integridad de los datos minimiza el mantenimiento, programación y reduce los costos, asegurando que los datos son correctos.
- Interoperabilidad dinámica con la red local.
- Monitoreo gráfico del rendimiento del sistema.
- Interfaces de programación remota como: *DB-Library* y *ODBC*.
- Un complejo y robusto manejo de transacciones con capacidad de *roll-forward* y *roll-back* que permiten la recuperación automática de transacciones al reinicio del sistema.

- Una arquitectura de un sólo proceso pero multitareas, asegura un desempeño del sistema eficiente para un gran número de usuarios con poco consumo de recursos.

Los precios de Microsoft SQL Server se muestran en la siguiente tabla:

Producto	Versión	Usuarios	Precio (USD)
MS SQL Server	6.5	1	\$995
		2 a 10	\$2,995
		11 a 64	\$7,995
		Ilimitados	\$14,995

Ahora bien, el costo de la licencia de server para Windows SQL 6.5 a precio académico es de \$7,043.00 pesos m.n., aproximadamente \$758.00 dólares.

## 2.6 Selección de Plataforma de Desarrollo

Una vez realizada la investigación y pruebas de rendimiento, velocidad, respuesta, seguridad y administración, entre otras, sobre los equipos seleccionados, los sistemas operativos de red y los manejadores de bases de datos, se generó un resumen comparativo en cuestión de costos, el cual se ilustra a continuación:

Producto	Opción No. 1	Costo	Opción No. 2	Costo
Servidor	Sparc 1000E	\$49,238	Alpha 2100	\$24,418
Sistema Operativo	Solaris 2.4	\$1,155	Windows NT 4.0	\$415
DBMS	Sybase 10	\$27,760	SQL Server 6.5	\$758
Costo Total (USD):		\$78,153		\$25,591

Es importante resaltar que los precios tanto de MS Windows NT como de MS SQL Server, son considerablemente bajos con respecto al precio de lista que tienen en el mercado. Esto se debe a que existe un convenio entre Microsoft y la UNAM, donde dicha compañía otorga un considerable descuento (mayor al 50%) en su *software* para las dependencias universitarias, de tal manera que los costos de las versiones de Windows NT y SQL Server son muy atractivos. Dado lo anterior, el costo total para la segunda opción sería de \$25,591.00, obteniendo una diferencia significativa de 52,562 dólares.

En cuestión de sistemas operativos, puede inferirse el problema de operación que

UNIX representa para usuarios no expertos y, en especial, para principiantes. En este sentido hay que considerar que el Patronato no cuenta a la fecha con personal experimentado en este ambiente, y que el capacitarlo le representaría altos costos, tanto económicos como de tiempo. Por otra parte resultaría más complicado acercar y familiarizar al usuario de desarrollo con este sistema que dirigirlos hacia el empleo del ambiente Windows, el cual cuenta con gran aceptación dentro del personal técnico y administrativo del Patronato. De lo anterior, se desprende que la implantación de sistema operativo UNIX en las unidades informáticas requeriría de más tiempo y esfuerzo para su cabal explotación.

Por otro lado, el Patronato fue la primer dependencia universitaria que utilizó Windows NT para el manejo y concentración de información masiva en un ambiente de red. Al inicio se instaló una red de área local (LAN), que con el paso del tiempo y con la infraestructura de RED-UNAM en pocos meses se convirtió en una red de área amplia (WAN). Dentro de Patronato, el ambiente Windows NT ha proporcionado resultados favorables, dado que se ha instalado tanto en equipos Intel, en donde se utilizan aplicaciones que no requieren mucho poder de cómputo (por ejemplo: procesadores de palabras, hojas de cálculo, correo electrónico, etc.), como en equipos con plataforma RISC, para aquellas aplicaciones que manejan grandes volúmenes de información y que son de misión crítica, tales como el manejo de bases de datos institucionales.

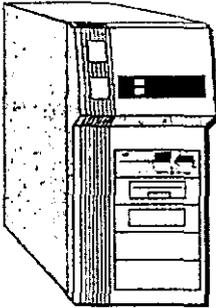
Hoy en día se tienen instaladas redes locales con sistema operativo Windows NT en la torre de Rectoría, en el edificio "B" de Zona Cultural, en las Unidades de Proceso Administrativo Ciencias, Cuernavaca y Juriquilla, en las ENEP's Acatlán y Aragón, y en las FES Iztacala y Cuautitlán; todas ellas interconectadas a través de RED-UNAM. Lo anterior, ha generado que en los últimos 6 años se haya ganado

un alto grado de experiencia en el manejo y administración del sistema operativo por parte del personal técnico y de desarrollo del Patronato.

En materia de manejadores de Bases de Datos, para tener una mejor perspectiva en la selección de la herramienta, se realizaron pruebas donde se colocó la misma base de datos sobre SQL Server con Windows NT y Sybase con UNIX, ambos corriendo sobre plataformas similares, a fin de poder obtener una comparación real entre los dos manejadores. En términos generales, ambos sistemas probaron tener la misma potencialidad y cumplieron con todas las pruebas realizadas, sin embargo, es importante hacer notar que SQL Server presentó la ventaja de operar en ambiente gráfico lo cual permite un mejor manejo, reduce la curva de aprendizaje y facilita por mucho la administración. En contrapartida Sybase está desarrollado para trabajar a nivel de texto, y aunque ya existen algunos productos para ambiente gráfico, éstos aún son muy limitados y no contemplan la gama de opciones que ofrece SQL Server.

En resumen, la plataforma que se decidió adquirir y utilizar para el desarrollo del sistema debido a que se adapta mejor a los requerimientos técnicos y económicos de Patronato, fue la del servidor RISC Digital modelo 2100, como sistema operativo de red, se eligió MS Windows NT Server en su versión 4.0 y como manejador de bases de datos se seleccionó a MS SQL Server versión 6.5. La Figura 2.2 esquematiza el equipo de cómputo a utilizar:

## Servidor de Desarrollo



### **Hardware**

- Digital Modelo 2100
- Procesador a 190 MHz.
- 192 Mb de RAM
- 6 Gb de Disco Duro

### **Software**

- MS Windows NT 4.0
- MS SQL Server 6.5

Figura 2.2 Plataforma de equipo seleccionada para el proyecto SIAU.

El componente de *software* cliente a utilizar, debía ser totalmente compatible con el *software* seleccionado hasta este momento para el servidor, a fin de garantizar la confiabilidad en la operación y manejo del nuevo sistema por parte de los usuarios. Este tema y algunos otros, se abordan a detalle en el siguiente capítulo.

## CAPITULO 3 ARQUITECTURA DE SOLUCION

Una vez seleccionada la infraestructura tecnológica a utilizar para el proyecto, se realizó la investigación correspondiente al *software*, es decir, lo relacionado al lenguaje de programación y las herramientas adicionales que debían utilizarse para el proyecto. En el presente capítulo se habla acerca de la arquitectura de software que se utilizó para el desarrollo del sistema; se toca también el tema de la filosofía Cliente/Servidor y de los almacenes masivos de datos, mejor conocidos como *Data Warehousing*.

### 3.1 Arquitectura Cliente/Servidor

Los sistemas de computación han evolucionado de tal modo que ya proporcionan una variedad de formas para el procesamiento distribuido, el cual además de simplificar la ejecución y el desarrollo de aplicaciones, favorece al máximo el aprovechamiento de los recursos de la red a nivel corporativo.

La arquitectura computacional cliente/servidor (c/s) es la primera solución en tecnología informática que satisface las presiones de costo-rendimiento de una empresa actual de alto nivel. Por esto se observa una clara tendencia de migración hacia este modelo.

Los motivos que llevan a evaluar la posibilidad de implantar una arquitectura cliente/servidor dentro de una empresa pueden ser varios, pero los principales pueden resumirse de la siguiente manera:

- Flexibilidad para satisfacer las necesidades del negocio.
- Reducción de Costos

- Identificación de la solución que se adecue más a las necesidades de la empresa y que crezca a la par de los requerimientos.
- Incremento en la productividad del personal.
- Aprovechamiento del equipo existente que no se utiliza al 100% de su capacidad.
- Consolidación de los procesos más confiables, ya que los recursos de cómputo pueden distribuirse en varios servidores y los clientes pueden configurarse de modo tal que cualquier aplicación o archivo de datos pueda ser accesado desde cualquier parte. Además, el aislamiento a fallas que se pueden incluir en este ambiente reduce la pérdida de tiempo.

Lo anterior, llevó al Patronato Universitario a inclinarse hacia el uso de esta arquitectura, basándose principalmente en servidores con procesadores RISC y algunos más con procesadores Intel; por lo que toca a las estaciones de trabajo (clientes) se adquirieron e instalaron equipos PC's, en su mayoría con procesador Pentium y algunas con 486, tratando de cubrir las necesidades de procesamiento de bajo y alto nivel que requiere el usuario final.

Una ventaja adicional en el caso del proyecto en cuestión, es que se puede obtener un aprovechamiento total de los recursos dado que todos los equipos (servidores y estaciones de trabajo), a pesar de no estar físicamente en el mismo lugar, se encuentran interconectados a través de Red-UNAM, lo cual hace a Patronato ponerse a la vanguardia en el uso y explotación de los canales de comunicación.

### **3.1.1 Ventajas de la Arquitectura Cliente/Servidor**

- Con la arquitectura cliente/servidor se pueden integrar diferentes tipos de

equipos y sistemas operativos en un ambiente único de procesamiento.

- Anteriormente, la capacidad de cómputo de una empresa se medía directamente por la capacidad del Mainframe o de las Minis con las que contaba. En el ambiente cliente/servidor la capacidad está constituida por todos los equipos interconectados, entre los que se incluyen las PC's.
- Otra ventaja de esta arquitectura es la considerable reducción en el tráfico de la red, ya que las operaciones de procesamiento se jerarquizan y la red da prioridad a las necesidades del usuario, transmitiéndole sólo la información realmente útil. Esto constituye un estímulo al uso de sistemas abiertos dado que tanto clientes como servidores corren en diferentes plataformas de *hardware* y *software*, lo cual permite que las empresas compren productos de diferentes proveedores, sin interferir en el desempeño de las demás aplicaciones y equipos instalados en la red.

Una de las grandes ventajas de la arquitectura cliente/servidor es la eliminación de los recursos ociosos de la empresa, porque hace que éstos se utilicen de manera óptima, incrementando la productividad de la red, con lo que la inversión en recursos adicionales se disminuye o se suprime.

El ambiente cliente/servidor permite la estructuración de una estrategia que permite que los recursos de cómputo se utilicen de la mejor manera posible, con lo cual se reducen los costos de manera considerable. Para lograr esto, se deben llevar a cabo ciertas medidas, las cuales se mencionan a continuación:

1. **Separación de Tareas.**- Las funciones de cómputo se pueden sincronizar con los recursos de tal manera que el servidor procese sólo las que son más complejas o que requieren operaciones intensas, liberando al cliente de esa

labor, dándole la oportunidad de ejecutar otras tareas simultáneas. Con esta configuración puede integrarse una estrategia que equilibre la combinación y costo de los dispositivos de cómputo.

2. **Comportamiento de Periféricos.-** La combinación de estos dispositivos, también puede realizarse al hacer que sus características coincidan con las funciones que se desea ejecutar y como la arquitectura c/s es abierta, permite la incorporación de un dispositivo especializado para realizar tareas muy específicas.
3. **Compartimiento de Herramientas.-** Los clientes pueden compartir diversas aplicaciones con la gran ventaja de que éstas pueden ser controladas de manera centralizada por un mismo administrador. Con esta ventaja, se elimina la necesidad de tener que instalar, configurar y dar de alta la aplicación en cada cliente cada vez que surja una nueva versión de *software*.
4. **Acceso a la Información.-** A pesar de que una empresa se encuentre geográficamente separada, la arquitectura c/s permite el acceso a la información a nivel del servidor de manera transparente, siempre y cuando existan estándares comunes.

En el caso del Patronato Universitario, se implantó un estándar para todas los sistemas y aplicaciones que corren tanto en los servidores como en las estaciones de trabajo. Esto es, se eligió como estándar una familia de productos que contara con una interface gráfica, sencilla y amigable para el usuario final, tal como el ambiente Windows.

Por tal motivo, como ya se mencionó en el capítulo anterior, el *software* utilizado en Patronato es de un sólo fabricante, la empresa Microsoft. En las estaciones

cliente se encuentra instalado MS Windows 3.1, MS Windows para Trabajo en Grupo 3.11 ó MS Windows 95/98; mientras que los servidores cuentan con MS Windows NT Server 4.0 y MS SQL Server 6.5 como sistema operativo de red y manejador de bases de datos, respectivamente.

La plataforma de productos mencionada, ha permitido dentro del Patronato, tener una compatibilidad total para el desarrollo de aplicaciones e intercambio de información entre aplicaciones, lo cual ha dado como resultado mejores niveles de productividad.

La esencia de c/s está precisamente en que se aprovecha lo que ya se tiene, con la única variante de que ahora se contará con nuevas funcionalidades. Las aplicaciones siguen siendo las mismas, pero son mucho más poderosas y, como han sido integradas, pueden comunicarse entre sí en un ambiente c/s.

La necesidad de procesar en forma distribuida las aplicaciones de tamaño mediano o grande, aprovechando mejor la base instalada de computadoras personales y redes locales, dio como resultado esta arquitectura, que permite descargar a los servidores de las funciones de procesamiento, eliminando cuellos de botella.

En esta arquitectura, el cliente maneja la interface del usuario, construye los requerimientos del servidor y controla la presentación de resultados, incluso puede contener la lógica de la aplicación. Cuando el servidor recibe una solicitud de servicio, realiza el proceso, regresa el resultado al cliente y provee el servicio de administración de datos. Tanto el cliente como el servidor pueden ejecutar el trabajo, así como la lógica de la aplicación.

De esta manera, las aplicaciones dentro del Patronato Universitario se desarrollan como se describe a continuación: el servidor se encarga de ejecutar todos aquellos

procesos que para el cliente podrían representar mucho tiempo de proceso, de tal forma que el servidor se encarga de realizar el trabajo pesado y el cliente se dedica a presentar los resultados a los usuarios finales.

### **3.1.2 Desventajas de la Arquitectura Cliente/Servidor**

Las cualidades que ofrece esta arquitectura, son interesantes y muy convincentes, pero esto no significa que sea perfecta. Sobre todo cuando lo que se persigue es la integración plena de ambientes heterogéneos.

Al trabajar en un ambiente de esta naturaleza la administración de la red se vuelve más pesada con respecto a los ambientes monolíticos tradicionales. Además, con esta arquitectura, la reproducción de ambientes de operación es un poco más complicada y aumenta el riesgo de que no se satisfagan todos los niveles de servicio esperados.

En un ambiente cliente/servidor, los recursos de cómputo de la red enfrentan cargas de trabajo muy grandes respecto a los sistemas centralizados, por lo que el rendimiento no es el mismo.

Por lo que respecta al caso Patronato, la plataforma cliente/servidor ha arrojado resultados muy favorables, ya que se ha explotado la infraestructura de Red-UNAM, lo que ha permitido aprovechar al máximo cada uno de los recursos con los que cuentan los servidores ubicados en la torre de Rectoría, en el edificio de Zona Cultural, en las Unidades de Proceso Administrativo Ciencias, Cuernavaca y Juriquilla, en las ENEP's Acatlán y Aragón, y en las FES Iztacala y Cuautitlán.

A pesar de que la administración no puede ser totalmente centralizada, gran parte de los procesos delicados de administración se pueden realizar remotamente.

dejando sólo los procesos de menor importancia en manos de administradores con menor experiencia y que no requieren un alto dominio en el manejo de su red local.

### 3.2 Almacenes de datos (DataWarehousing)

El *Data Warehousing* es la creación de una estructura para la toma de decisiones a través de bases de datos que son en realidad bases de metadatos, es decir, que tienen información acerca de los datos. La labor del *Data Warehousing* es resumir, analizar y filtrar los datos operativos para obtener información valiosa que ayude en la toma de decisiones.

Aunque los conceptos para el *Data Warehousing* se han establecido en la última década, es en realidad en los últimos años cuando ha cobrado relevancia debido al surgimiento de los sistemas cliente/servidor y a las nuevas tecnologías en manejadores de bases de datos (*DBMS, Data Base Management Systems* o Sistemas de Administración de Bases de Datos).

El concepto es sencillo. La idea es crear bases de datos y sistemas informáticos que analicen los archivos históricos para obtener información valiosa en la toma de decisiones diaria y a futuro. Su gran aportación consiste en poner en las manos de los administradores la información crítica sobre el comportamiento de su negocio de una manera rápida, sencilla y resumida.

Una de las bases fundamentales de este nuevo tipo de herramientas es el reconocimiento de los dos tipos de información que pueden encontrarse en cualquier organización:

- **Los (sistemas) procesos operativos.** Son aquellos procesos “de todos los días” que mantienen en operación a la empresa (sistemas de contabilidad, inventario, cheques, etc.).
- **Los informacionales.** Son aquellos más bien enfocados a la planeación y toma

de decisiones sobre el rumbo del negocio (análisis financiero, planeación de mercados, etc.).

Gracias a los adelantos en *hardware* y *software* el *Data Warehousing* ha pasado rápidamente de una serie de conceptos técnicos, a productos reales que ya están dando buenos resultados en muchos lugares alrededor del mundo.

### 3.2.1 Manejo de Datos

Aunque el manejo de datos parece un problema sencillo, no lo es así, ya que existen una serie de dificultades a vencer:

- Los datos suelen estar dispersos en una gran cantidad de plataformas, sistemas operativos y programas diferentes. De hecho este es uno de los principales problemas que se pueden tener en la creación de un *Data Warehousing* por su nivel de complejidad y trabajo necesario para solucionarlo.
- Claramente la meta del *Data Warehousing* es liberar la información almacenada y escondida en los datos operacionales y mezclarla con información de otras fuentes, incluso externas, para la toma de decisiones rápidas y acertadas.

El desarrollo de un buen sistema de *Data Warehousing* no difiere en mucho al de los programas tradicionales empleados en cualquier otro desarrollo de sistemas informáticos. La parte más importante radica en el proceso de planeación, a través del cual se establecerá con claridad el tipo de solución a implantar en la empresa.

Existen tres factores de suma importancia que se deben considerar a la hora de establecer un *Data Warehousing*, éstos son:

1. **Alcance del *Data Warehousing*.** Pueden crearse sistemas que analicen los datos a lo largo de períodos muy grandes y desde todas las bases de datos, o bien, puede necesitarse sólo un sistema personal para el gerente de un departamento. En general, conforme mayor sea el alcance de un sistema, mayor será la utilidad que se obtenga del mismo, pero también crecerá su complejidad y los costos de utilización.
2. **Redundancia de los datos.** Pueden establecerse bases de datos virtuales, centralizadas o distribuidas. Aunque en ocasiones se suele asociar esta herramienta con una base central, los otros tipos también pueden ser implantados.
3. **Usuario Final.** Es importante conocer el perfil y características de los usuarios finales. No es lo mismo desarrollar un sistema para personal técnico que uno para administradores, pues los usuarios tienen necesidades e intereses diferentes, mismos que deben ser considerados dentro de la aplicación.

### 3.2.2 Procesos en *Data Warehousing*

Se pueden establecer varios procesos en cualquier sistema de *Data Warehousing*; los más importantes se muestran a continuación:

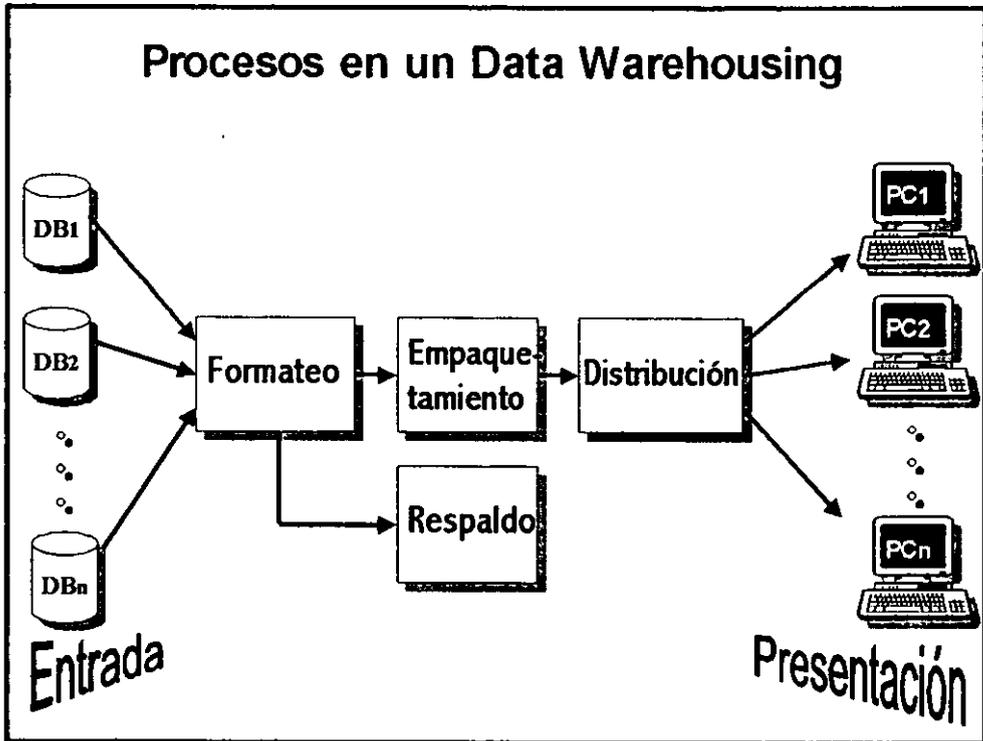


Figura 3.1 Procesos de un sistema *Data Warehousing*.

- **Inflow** (Entrada). Es el flujo de información que va desde las bases de datos operativas hacia el sistema. Este procedimiento puede ser complejo y a gran escala, debido a la gran cantidad de datos existentes y generados día a día a lo largo y ancho de una organización.

- **Upflow** (Formateo). Es el proceso de agregar valor a los datos contenidos en el *warehouse*. Aunque a veces se pueda pensar que basta con un buen *query* (consulta) para obtener información valiosa, la verdad es que antes es necesario realizar una serie de tratamientos para facilitar el proceso, de entre los cuales destacan:

1. **Packing** (Empaquetamiento). Hay que transformar la información resumida en algún formato más adecuado para su lectura (hojas de cálculo, gráficas, diagramas, etc.)
2. **Distributing** (Distribución). Enviar información hasta los usuarios finales para que puedan hacer uso de ella; es aquí donde cobran particular relevancia las arquitecturas cliente/servidor.
3. **Downflow** (Respaldo). Es necesario establecer mecanismos para transferir los datos viejos e innecesarios hacia localidades fuera de línea, tales como unidades de cinta o discos ópticos.
4. **Outflow** (Presentación). Constituye el fin último de todo proceso de *Data Warehousing*, ofrecer información precisa para la toma de decisiones.

### 3.2.3 Consideraciones

*Data Warehousing* no es un fenómeno reciente, sus bases teóricas han sido establecidas en la última década, pero ya existen sistemas en muchas empresas que están ofreciendo sus primeros frutos y demuestran ser de gran utilidad.

Sin embargo, debido a las grandes inversiones de tiempo y dinero que requiere, el establecimiento de *Data Warehousing* no es de ninguna manera barato o rápido.

puede ir desde 6 meses hasta 2 ó 3 años para grandes corporaciones.

Es por ello que se debe tener especial cuidado en las etapas tempranas, pues si el sistema no está correctamente diseñado, podrá ser causa de frustraciones y gastos innecesarios.

Sin lugar a dudas, lo que este trabajo plantea es el desarrollo de un *Data Warehousing* del Patronato Universitario y la Secretaría Administrativa de la Universidad, ya que se cuenta con gran cantidad de información financiera y de gestión en ambas dependencias, la cual se encuentra almacenada en diversos medios y que, desafortunadamente, no está al alcance de las personas encargadas de la toma de decisiones dentro de la Institución, es decir, de los Directores y Secretarios Administrativos de facultades, escuelas, institutos y de dependencias administrativas, principalmente.

El proceso de crear una gran base de datos no es tarea sencilla, ya que se tiene que obtener información de diferentes fuentes; además, en muchos de los casos a esta información se le tiene que “maquillar” para que adquiera el formato requerido, lo que lleva a la construcción de pequeños sistemas de actualización e inserción de información.

Para el caso del sistema en cuestión, antes de alimentar la base de datos central, se corren programas encaminados a modelar la información para que se pueda integrar a las tablas correspondientes dentro de la base de datos. Una vez terminados estos procesos, la información obtenida se carga para que pueda ser consultada.

En la siguiente página se presenta el cuadro que describe de manera general dichos programas:

<b>Nombre del Programa</b>	<b>Descripción</b>	<b>Periodicidad</b>
Pólizas	Lee 2 archivos de datos de un promedio de 20 mil registros, el primero contiene el número de póliza y su cantidad; el segundo, el número de póliza y un comentario. Al finalizar el proceso genera un sólo archivo con los datos de los dos primeros, relacionando el comentario al número de póliza correspondiente.	Diariamente
Actualiza Códigos	Lee 2 archivos de datos de más de 30 mil registros, el primero contiene los códigos presupuestales hasta el día inmediato anterior, y el segundo tiene los mismos códigos, además de aquellos de nueva creación. Al finalizar el proceso, se genera un archivo con los códigos y saldos del día anterior, aunados a los códigos y saldos del día en curso.	Diariamente
Transferencias	Se encarga de subir a la base de datos los 14 archivos generados por el área de Procesamiento, más los 2 generados por los anteriores programas. Subiendo un promedio de 250 mil registros a las diferentes tablas de la base de datos del sistema.	Diariamente

(continuación)

<b>Nombre del Programa</b>	<b>Descripción</b>	<b>Periodicidad</b>
Folios UPA's	Se encarga de leer en cada uno de los 6 servidores de las Unidades de Proceso Administrativo 5 tablas correspondientes al estado que guardan los folios (documentos) procesados. Al final del proceso toda la información procesada es grabada directamente en 3 tablas de la base de datos.	Diariamente
PAPIIT	Se conecta al servidor del área de PAPIIT, lee 10 archivos y los inserta directamente a 10 tablas de la base de datos.	Diariamente
Intereses	Realiza el cálculo de los intereses que por ingresos extraordinarios haya generado cada dependencia.	Mensualmente
Ingresos Extraordinarios	Actualiza con la información generada por el programa anterior las tablas correspondientes a Ingresos Extraordinarios del mes calculado.	Mensualmente
Boletos	Lee información de un archivo enviado por la agencia de viajes, y actualiza la tabla de boletos de avión del mes correspondiente.	Mensualmente

Los primeros cinco programas se corren por las noches llevándose un promedio de 50 minutos, con ello se garantiza que la información esté disponible para los usuarios a primer hora del día siguiente; los últimos 3, se ejecutan una vez al mes también por las noches.

Cabe señalar que antes y después de ejecutar los procesos anteriores, se realiza un respaldo completo de la base de datos, ya que si por alguna razón uno de los programas no terminara satisfactoriamente, se procede a levantar el respaldo del día inmediato anterior. Ahora bien, si el proceso termina con éxito, se lleva a cabo el respaldo de información tanto en disco como en cinta, con el fin de que cuando se presente una falla grave del equipo (ya sea de *hardware* o de *software*), la base de datos pueda ser cargada en otro equipo con características similares e incluso menores al servidor principal para su operación emergente.

En la figura que se muestra en la siguiente página, se esquematiza el proceso de *datawarehouse* que se realiza para el Sistema de Información objeto del presente trabajo.

## Data Warehousing de Patronato

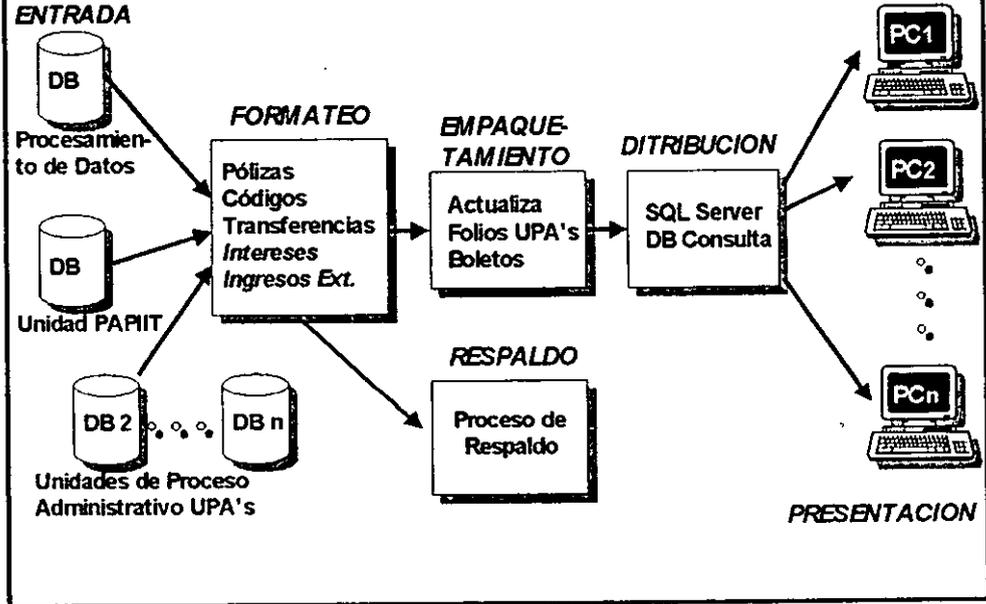


Figura 3.2 Proceso de Datawarehouse para el SIAU.

### **3.3 Lenguajes de Programación**

Contando ya con el sistema operativo y el manejador de bases de datos, el paso siguiente fue seleccionar el lenguaje de programación. Para el desarrollo del proyecto, se seleccionó a MS Visual Basic como la herramienta de desarrollo, ya que por un lado, pertenece a la misma línea (marca) de los productos anteriores. Por otro lado, el uso constante de productos de un mismo fabricante ha permitido, como ya se mencionó, obtener *software* a precios muy competitivos, además de tener respaldo de una empresa reconocida en el país y, principalmente, garantizar el intercambio de información a todos los niveles de aplicación, sin importar la diversidad de equipo con el que se trabaje. A continuación, se citan las características más sobresalientes de esta herramienta de desarrollo.

#### **3.3.1 MS Visual Basic**

Visual Basic es un lenguaje de programación gráfico para el desarrollo de aplicaciones en ambiente Windows. Sus características lo hacen muy diferente a la gran mayoría de los lenguajes de programación tradicionales de alto nivel, ya que resulta de fácil manejo y presenta una forma ingeniosa, innovadora, rápida y sencilla de crear aplicaciones.

La técnica de programación es denominada "Orientada a Eventos", debido a que con base en cada uno de los eventos requeridos, se realizan las funciones de las aplicaciones desarrolladas.

El lenguaje Visual Basic se puede describir como un sistema de desarrollo diseñado especialmente para crear aplicaciones gráficas. Su facilidad radica en que presenta una visión intuitiva de los botones de mando, botones de opciones, cuadros de verificación, cuadros de listas, ventanas, iconos, menús y cuadros de diálogo que

están ahora al alcance de los entornos de programación.

Como su nombre lo indica, en este lenguaje los programas son creados de forma totalmente visual. Con la ayuda del ratón, el programador extrae de un cuadro de utilidades los objetos, tales como botones, barras de desplazamiento, cuadros de entrada de texto y otros controles, ubicándolos directamente en la pantalla.

Después de la preparación de las pantallas o ventanas de un programa, de forma directa e inmediata, sin escribir ni una línea de programa, el programador describe las funciones de cada objeto y su relación con los estímulos externos, como pulsaciones de tecla o en los botones del ratón.

Finalmente, se escribe el código fuente relacionado con cada objeto; esto es, cada objeto está ligado a un código que permanece inmóvil hasta que se dé el suceso que lo activa, por ejemplo, un clic del ratón.

Con este sistema, puede desarrollarse cualquier tipo de aplicación basado en el entorno Windows, incluyendo sistemas administrativos, herramientas, utilidades, aplicaciones *front-end* para el proceso de datos de servidores o procesos locales y hasta productos de *software* comercial, ya que pueden ser creados programas ejecutables.

Las herramientas con las que cuenta Visual Basic son:

- Soporte para intercambio dinámico de datos con otras aplicaciones (*DDE dynamic data exchange*).
- Soporte para establecer enlaces con Windows y con librerías dinámicas escritas en otros lenguajes (*DLL dynamic link libraries*).

- Un sistema de ayuda en línea.
- Un fichero de íconos para trabajos profesionales.

Cuando se combinan estas características, se dispone de un sistema de desarrollo que permite diseñar rápidamente aplicaciones sofisticadas.

Los elementos que conforman Visual Basic son básicamente dos:

- **Formas.**- es una ventana que sirve de fondo para los controles y para los gráficos situados sobre la misma. Se pueden utilizar tantas formas como se necesiten y dependiendo de la utilidad a las que se les destine, éstas serán de diferentes tipos.
- **Controles.**- son objetos que dibujamos sobre una forma con fines de aceptar o visualizar datos. Para dibujar los controles en una forma, Visual Basic pone a disposición del usuario un panel de utilidades (*toolbox*).

La forma de denominar al código empleado que se une a un objeto se le conoce como "procedimiento conducido por un suceso", o simplemente como "procedimiento".

Una de las grandes ventajas de trabajar con Windows y que Visual Basic explota, es que todas las ventanas se comportan de la misma forma y todas las aplicaciones utilizan los mismos métodos básicos (menús descendentes, botones, etc.) para introducir órdenes. Esto por supuesto presenta una forma de programar y de pensar mucho más eficientes.

Cuando se desarrollan aplicaciones bajo este tipo de programación, la secuencia en la que van a ejecutarse las sentencias no puede ser prevista por el programador. Por

lo tanto, para programar una aplicación Windows hay que escribir código separado para cada objeto en general, quedando la aplicación dividida en pequeños procedimientos conducidos cada uno de ellos por un suceso. Esto es a lo que se le conoce como **“Programación orientada a eventos”**.

Dentro de Patronato se ha utilizado Visual Basic como lenguaje de desarrollo final (*front-end*), esto es, las aplicaciones desde el cliente hacen consultas hacia un servidor de bases de datos; éste a su vez, realiza las tareas que le indica el cliente y regresa los resultados para que sean desplegados al usuario final.

La experiencia en el uso de Visual Basic dentro de la dependencia, es principalmente en aplicaciones que interactúan con bases de datos (como el Sistema Contable-Presupuestal), de tal manera que se aproveche al máximo el uso de la arquitectura cliente/servidor, es decir, la aplicación Visual Basic (el cliente) solicita datos al usuario y es generadora de consultas (*queries*) mismas que se envían al manejador de bases de datos (servidor), para que realice el trabajo de validación, proceso y actualización de información sobre las tablas de las bases de datos; quedando como último paso, devolver únicamente resultados al cliente, para que este a su vez lo despliegue en forma gráfica al usuario final.

### 3.4 Conectividad Abierta Para Bases de Datos (ODBC)

La *interface* ODBC permite a las aplicaciones acceder datos en manejadores de bases de datos (*Data Base Management System –DBMS*), usando lenguaje de consulta estructurada (*Structured Query Language –SQL*) como un estándar para acceder información. La interface permite máxima interoperabilidad (una sola aplicación puede acceder diferentes DBMS). Esto permite a un desarrollador crear, compilar y portar una aplicación sin tener como objetivo un DBMS específico. Los usuarios pueden agregar módulos llamados: “*drivers* de bases de datos”, que ligan la aplicación al DBMS de su elección.

En el mundo tradicional de las bases de datos, “aplicación” significa un programa que produce una tarea de bases de datos específica con un DBMS específico, como puede ser nómina, contabilidad, análisis financiero o inventario, por mencionar algunos ejemplos. Tales aplicaciones han sido típicamente escritas usando *embedded SQL*. En tanto que *embedded SQL* es eficiente y es portable entre diferente equipo y ambientes de sistemas operativos, el código fuente debe ser recompilado para cada nuevo ambiente.

ODBC ofrece una nueva propuesta: provee un programa separado para extraer la información de la base de datos, y luego, tener una forma para que las aplicaciones importen los datos. Dado que hay, y probablemente siempre habrá, muchos métodos de comunicación viables, protocolos de datos, y capacidades de los DBMS, la solución ODBC permite el uso de diferentes tecnologías para definir una *interface* estándar. Esta solución conduce a la idea de *drivers* de base de datos, de tal manera que éstos puedan ser utilizados para acceder a un DBMS específico, como cuando se utiliza uno para una impresora corriendo bajo Windows. ODBC

provee la interface estándar que permite a los desarrolladores y a los proveedores de librerías pasar datos entre aplicaciones y fuentes de datos.

La interface ODBC define lo siguiente:

1. Una librería de llamadas a funciones ODBC que permiten a una aplicación conectarse a un DBMS, ejecutar sentencias SQL y regresar resultados.
2. Sintaxis SQL basada en la especificación *X/Open* y *SQL Access Group* (1991).
3. Un conjunto estándar de códigos de error.
4. Una manera estándar para conectarse y obtener el *logon* a un DBMS.
5. Una representación estándar para tipos de datos.

Ahora bien, la flexibilidad de la interface ODBC se puede resumir en lo siguiente:

- *Strings* conteniendo sentencias SQL que pueden ser explícitamente incluidas en código fuente o construidas en tiempo de ejecución.
- El mismo código objeto puede ser usado para acceder diferentes productos y marcas de DBMS.
- Una aplicación puede ignorar protocolos de comunicación subyacentes entre ella y un DBMS.
- Valores de datos pueden ser enviados y regresados en un formato

conveniente a la aplicación.

### 3.4.1 Arquitectura ODBC

La arquitectura ODBC consiste de cuatro componentes, los cuales se describen a continuación:

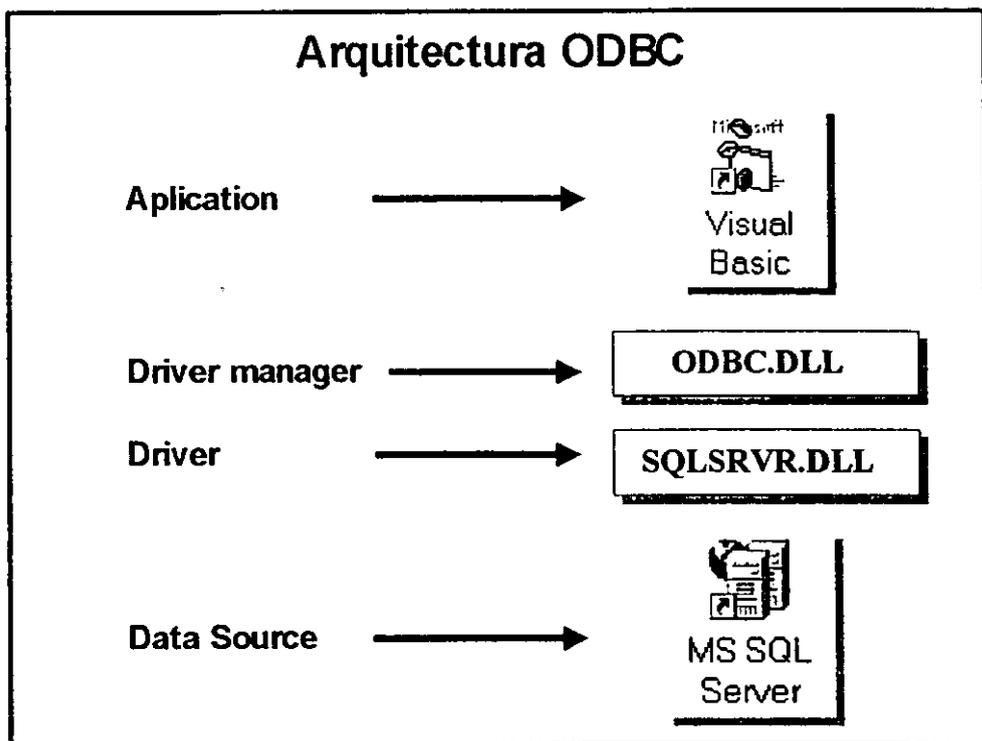


Figura 3.3 Componentes de la arquitectura ODBC.

#### Aplicación.

Realiza el procesamiento y llama a funciones ODBC para presentar sentencias SQL y regresar resultados.

## **Driver Manager.**

Carga *drivers* en la ejecución de una aplicación.

## **Driver.**

Procesa llamadas a funciones ODBC, presenta requisiciones SQL a una fuente de datos (*data source*) específica y regresa resultados a la aplicación. Si es necesario, el *driver* modifica una requisición de la aplicación tal que la requisición se adecue a la sintaxis soportada por el DBMS asociado.

## **Data Source (Fuente de Datos).**

Consiste de los datos que el usuario quiere acceder, del sistema operativo asociado, del DBMS y de la plataforma de red utilizada para acceder el DBMS.

El *driver manager* y el *driver* aparecen en una aplicación como una unidad que procesa llamadas a funciones ODBC.

### **3.4.2 Visual Basic y ODBC**

Porque ODBC es de propósito general, no hay consideraciones especiales para usarlo desde Visual Basic. De hecho, ODBC fue diseñado para aumentar las capacidades de acceso de lenguajes como Visual Basic.

Utilizar los servicios de ODBC es equivalente a llamar funciones en cualquier DLL (*Dynamic Link Library*). Por tal motivo, el programador en Visual Basic debe tener en consideración lo siguiente:

- Aprender las funciones que necesita usar.

- Investigar la sintaxis y convenciones de llamadas a las funciones.
- Declarar las funciones externas y constantes a Visual Basic.
- Llamar a los procedimientos, justo como a cualquier otra función de Visual Basic.
- Procesar los resultados de las funciones.

Debido a que la aplicación ha sido pensada para crecer en un futuro, fue necesario la utilización de ODBC, ya que esto garantiza, que en los años próximos se pueda acceder desde la misma computadora (cliente) a diferentes manejadores de bases de datos, sin importar el tipo de equipo, sistema operativo y base de datos utilizados en el servidor remoto; lo que a la larga redundará en un mejor aprovechamiento de los recursos de cómputo de la Universidad.

Dentro de las dependencias administrativas de la UNAM, lo más utilizado son equipos Sun con Unix como sistema operativo y Sybase como manejador de bases de datos, de tal manera que la aplicación motivo del presente trabajo, garantiza poder comunicarse sin problema alguno con estas dependencias, siempre y cuando se utilice ODBC y se mantenga el protocolo de comunicación TCP/IP.

### 3.5 Herramientas Adicionales

Adicionalmente al lenguaje de programación seleccionado y a los *drivers* de comunicación a bases de datos (ODBC), se utilizaron algunas herramientas complementarias también de *software*: una para ayudar a realizar la programación más rápida y sencilla en cuestión de acceso a bases de datos; otra para acelerar el diseño y presentación de las pantallas dentro del sistema; dos más para el manejo de impresión y de direcciones IP; una diferente para el acceso a tablas con campos de tipo binario, y la última para el manejo de imágenes.

Todas ellas son totalmente compatibles con la versión de Visual Basic seleccionada, de tal manera que se garantiza la ausencia de problemas en la programación e implantación del sistema. Todas ellas se describen a continuación:

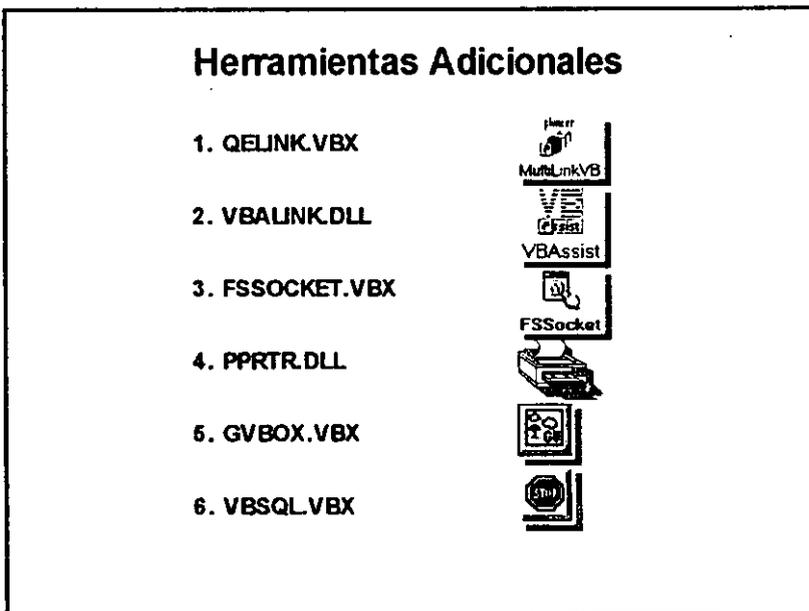


Figura 3.4 Herramientas adicionales al lenguaje Visual Basic.

### 3.5.1 QELINK.VBX

Es un conjunto de controles diseñado para proveer un acceso más rápido y sencillo, a diferentes manejadores de bases de datos, tales como Oracle, Sybase, SQL Server, Progress, Ingres y DB2 entre otros. Además de que permite acceder de la misma manera a archivos de Dbase (\*.DBF), Excel (\*.XLS), Paradox y archivos de texto.

Entre sus principales ventajas se pueden mencionar las siguientes:

- Se reduce en un 90% las líneas de código dentro de un programa para acceso a bases de datos.
- Cuenta con 8 tipos de objetos para desplegar información directamente de la base de datos sin importar su patrón.
- Su curva de aprendizaje es un 50% más rápida que aquella obtenida con métodos tradicionales.
- Debido al manejo de diferentes *drivers*, es posible acceder a más de ocho manejadores de bases de datos simultáneamente desde una misma aplicación, haciendo el acceso totalmente transparente para el usuario, sin importar el *hardware* y *software* donde estén instalados los DBMS.
- El acceso a archivos tipo DBF o de texto es igual que aquel de una base de datos, por lo cual se pueden utilizar SQL como lenguaje, generándose las mismas líneas de código.

### 3.5.2 VBALINK.DLL

Mejor conocida como Visual Basic Assist, es una herramienta que amplía el ambiente de desarrollo de Visual Basic, es decir, mejora la capacidad de poner orden a los objetos dentro de una forma, así como permite sincronizar sus propiedades, agruparlas y ordenarlas de una manera muy rápida y sencilla; lo que se traduce en acortar tiempos para la creación de grandes aplicaciones.

Las herramientas de VBAssist se pueden agrupar en tres categorías que son:

- 1.- Herramientas para la Manipulación de Controles:** son funciones que permiten alinear controles a la izquierda, derecha, arriba y abajo. También sirven para centrarlos dentro de una forma y hacerlos del mismo tamaño. Además, ayudan al manejo de arreglos de controles y permiten ver y modificar el *Tab Order* de los mismos.
- 2.- Herramientas para el Manejo de Propiedades:** permiten sincronizar una o varias propiedades de uno o más controles al mismo tiempo, tales como color, letra, imagen, etc. Además, permiten crear un tipo de control *customizado*, para ser utilizado en diferentes formas del proyecto.
- 3.- Herramientas Adicionales:** brindan la oportunidad de editar formas y módulos de otros proyectos, lo que ayuda a copiar código de un proyecto a otro. También permite crear formas más grandes que las normales y ayuda a la creación de menús y cajas de diálogo.

### 3.5.3 FSSOCKET.VBX

Debido a que la información que el sistema contiene es básicamente de carácter financiero, se requería tener un alto grado de seguridad con respecto al acceso de los usuarios, de tal manera que se debía eliminar la posibilidad de que accasaran usuarios no autorizados o no deseados. Por tal motivo se utilizó esta herramienta para controlar el acceso a los usuarios del sistema, ya que permite conocer la dirección IP, el nombre de la PC, su dominio y algunos parámetros más de cada cliente conectado al mismo. Cabe señalar que el tema de la seguridad se tocará a detalle en el siguiente capítulo.

### 3.5.4 PPRTR.DLL

Dado que Visual Basic en su versión 3.0 tiene algunas limitaciones en el manejo y uso de comandos de impresión, la librería PPRTR.DLL elimina esas desventajas, ya que dentro de sus características principales, permite saber si existe una impresora en red o conectada directamente al equipo cliente; si ésta se encuentra en línea; si tiene papel; permite además cambiar su orientación, los tamaños y tipos de letra, etc. Cabe señalar que esta librería fue obtenida a través de Internet, en un sitio Web donde existe *software* de dominio público; de tal manera que no tuvo que ser erogado gasto adicional por usar este conjunto de funciones.

### 3.5.5 GVBOX.VBX

Al igual que el control anterior, éste fue adquirido via Internet, sólo que en este caso se tuvo que adquirir el *software* con un valor de 20 dólares. Con esta librería se pueden manejar imágenes de una gran variedad de formatos: GIF, JPG, BMP, PCX, TIF y WMF, entre otros. Además, tiene implantado un método de impresión que permite imprimir directamente lo que se ve en pantalla sin agregar código. Otra

ventaja es que maneja el zoom de las imágenes, permitiendo agrandar o disminuir el tamaño en el porcentaje que el usuario desee.

### **3.5.6 VBSQL.VBX**

Esta librería es parte del MS SQL Server, viene junto con el paquete original y tiene la ventaja de no utilizar ODBC, lo que permite hacer conexiones más rápidas al servidor de base de datos, pudiendo con ello, importar y exportar información de una manera muy eficiente. Tiene la desventaja de que se debe crear mucho más código que utilizando Multilink, sin embargo, es muy segura y sirve para la transferencia de archivos binarios, tales como imágenes, programas, archivos de audio y/o video, entre otros.

## **CAPITULO 4 DESARROLLO DEL SISTEMA**

A lo largo de este capítulo, se hace referencia al enfoque sistemático que el desarrollador debe aplicar en el análisis y el diseño de los sistemas de información. Una gran parte de este enfoque, se encuentra en lo que se denomina “ciclo de desarrollo de los sistemas” o “ciclo de vida de los sistemas” (SDLC; Systems Development Life Cycle). El SDLC es un enfoque por etapas de análisis y de diseño, que postula que el desarrollo de los sistemas es más productivo cuando existe un ciclo específico de actividades del analista y de los usuarios.

Actualmente, no existe un acuerdo respecto al número exacto de etapas que conforman el ciclo de desarrollo de los sistemas; sin embargo, por lo general se reconoce la importancia de su enfoque sistemático. Para fines de explicación del presente trabajo, este ciclo se dividirá en seis etapas que inician con el análisis, diseño y desarrollo del sistema; continúan con las pruebas y la implantación del mismo, y finalizan con su mantenimiento.

Cabe mencionar que a pesar de que cada etapa puede identificarse de manera discreta, nunca se lleva a cabo como un evento independiente. En lugar de ello, el ciclo implica la realización simultánea de diversas actividades y, es posible que alguna de ellas se repita en etapas posteriores.

### **4.1 Análisis**

En esta primera etapa, el analista se involucra con la identificación de la situación problema, de las áreas de oportunidad y de los objetivos del proyecto. Esta fase es crucial para el éxito del resto del trabajo, pues nadie estará dispuesto a desperdiciar su tiempo dedicándolo al problema equivocado. Además, se requiere que el analista

observe de forma objetiva lo que ocurre en una empresa, para luego, en conjunto con otros miembros de la organización, haga notar los problemas detectados.

La fase de análisis se centra sobre el *qué*. Es decir, durante el análisis, la persona que desarrolla el software intenta identificar qué información ha de ser procesada; qué función y rendimiento se desea; qué comportamiento deberá tener el sistema; qué interfaces van a ser establecidas; qué restricciones de diseño existen y qué criterios de validación se necesitan para definir un sistema en forma correcta. Por lo tanto, han de identificarse los requisitos clave del sistema y del software.

#### 4.1.1 Planteamiento del Problema

En el primer capítulo, se explicó de manera general cuál era la problemática que vivía Patronato para la distribución de información hacia las dependencias. En esta sección se describe, de acuerdo con el cuadro que se muestra a continuación, cada uno de los problemas observados, considerando la forma de trabajo de un sistema que fue desarrollado en la década de los 70's y con el cual se operaba hasta el momento en que fue iniciado el sistema motivo del presente trabajo.

No.	Problema
1	Tecnología Obsoleta
2	Demora en el Proceso
3	Excesivo Tiempo de Impresión y Gasto de Papel
4	Amplio Espacio Dedicado al Almacenamiento

continuación...

No.	Problema
5	Información Limitada
6	Difícil Explotación de la Información
7	Poca Disponibilidad de Información Histórica
8	Escaso Control y Falta de Seguridad de la Información
9	Imposibilidad de Manejo de Imágenes
10	Imposibilidad para Transferir Archivos

### 1. Tecnología Obsoleta

La tecnología utilizada para el desarrollo y operación del sistema anterior corresponde a equipos de los años 70's, la cual presenta los siguientes inconvenientes. El mantenimiento y operación tanto del equipo principal como de las impresoras resultaba demasiado costoso debido, en primera instancia a que, derivado de su estructura y dimensiones, el equipo principal necesitaba de amplios espacios para su habilitación y requería de instalación y adecuaciones de alto nivel; en segundo término, debido a sus características de funcionamiento, el equipo requería ser operado por personal especializado; por último, a pesar de que en la actualidad ofrece un buen nivel de funcionamiento, el equipo instalado ha sido rebasado tanto en capacidad como en velocidad de operación por los avances

alcanzados por la tecnología del momento. Por todas estas razones, el equipo Unisys A7 en el que operaba el sistema existente fue calificado como insuficiente para los requerimientos planteados por las dependencias universitarias.

## **2. Demora en el Proceso**

El proceso nocturno de consolidación de información se llevaba a cabo en equipo de uso compartido con otras dependencias universitarias. Esto es, además de Patronato, el equipo daba servicio a la Dirección General de Personal, a la Dirección General de Obras, a la Dirección General de Programación Presupuestal y a la Dirección General de Normatividad y Sistemas Administrativos, principalmente. Por tal motivo, para que Patronato pudiera hacer uso del recurso era necesario aguardar turno y, en ocasiones, el tiempo de espera demoraba considerablemente el proceso.

## **3. Excesivo Tiempo de Impresión y Gasto de Papel**

Dado el gran volumen de reportes que debían generarse después del proceso de la información, se requería dedicar un día completo, cubriendo tres turnos, solamente a la impresión de grandes cantidades de hojas. Para dar una idea de la magnitud de la impresión, a continuación se mencionan las cantidades de papel que era utilizado en el proceso:

17,000 hojas para Estados de Cuenta Presupuestales

2,400 hojas para Estados de Cuenta de Ingresos Extraordinarios

1,200 hojas para Compromisos Previos

2,500 hojas para Gastos a Reserva de Comprobar

Estas cantidades sumadas dan un total de 23,100 hojas mensuales, las cuales además de representar un difícil manejo, constituían un gasto muy elevado para la Institución.

#### **4. Amplio Espacio Dedicado al Almacenamiento**

En el área de Contaduría General, encargada de la concentración de información impresa y conciliación con las dependencias, fue necesario destinar un espacio físico adicional para almacenar una copia de los reportes generados mensualmente y para disponer de un archivo de consulta de información. Tomando en consideración que en la Institución el espacio disponible es sumamente reducido, el lugar que ocupaban los reportes bien podría ser utilizado para la ampliación o creación de oficinas, mejorando en la medida de lo posible el ambiente físico de los trabajadores.

#### **5. Información Limitada**

Bajo el esquema propuesto por el sistema anterior, las dependencias estaban sujetas a recibir información únicamente relacionada con Compromisos Previos, Gastos a Reserva de Comprobar, Estados de Cuenta Presupuestales y Estados de Cuenta de Ingresos Extraordinarios, siendo imposible consultar información adicional que podría ser de mucha utilidad, tal es el caso de las Pólizas Contables, las Transferencias Presupuestales y el Estado de los Folios tramitados en las Unidades de Proceso Administrativo, entre otros. Esta limitación, ocasionaba que los usuarios tuvieran que acudir a las oficinas del Patronato para solicitar información adicional, con el consecuente consumo de tiempo que repercutía directamente en la oportunidad de la toma de decisiones.

## **6. Difícil Explotación de la Información**

Debido a que el usuario final únicamente recibía reportes impresos, el poder generar cuadros concentrados, gráficas o algunos cálculos con base en los reportes, le era prácticamente imposible a menos que capturara nuevamente la información. Por otra parte, en caso de requerir la localización de un dato en específico, la búsqueda debía realizarse en forma manual, revisando el reporte hoja por hoja, con la consecuente fatiga del personal y el deterioro paulatino de los ejemplares impresos. Otro aspecto que cabe mencionar es la poca disponibilidad de información de primera línea con la que contaban los directores y mandos superiores de cada dependencia, debido a que los reportes se entregaban a los mandos medios, delegando en ellos la difusión de la información, responsabilidad que en muchas ocasiones no se cumplía.

## **7. Poca Disponibilidad de Información Histórica**

En caso de que alguna dependencia requiriera información específica de meses o años anteriores, era necesario volver a imprimir los reportes correspondientes, proceso que se alargaba por más de una jornada laboral, demorándose días y en ocasiones semanas. Esta demora se debía a que los equipos A7 tienen una capacidad de almacenamiento muy limitada, por lo que resultaba necesario resguardar en cintas y discos intercambiables la información anterior, a fin de tener disponible espacio para la operación actual. Por tanto, para poder recuperar la información anterior, era necesario localizar el disco o cinta correspondientes, cargar la información en el equipo y generar nuevamente el reporte, proceso que en sí mismo consumía cierto tiempo. Por otra parte, considerando que la operación diaria debía cumplir con una secuencia específica y guardar turno para el proceso, el volver a cargar información anterior implicaba hacer todo un trámite especial

para insertarlo dentro de la operación de los procesos, debiendo solicitar un espacio adicional de tiempo máquina.

#### **8. Escaso Control y Falta de Seguridad de la Información**

Dado que el procedimiento para la generación de reportes requería la intervención de diferentes grupos de trabajo para su consolidación existía un alto riesgo de inseguridad. Es decir, el proceso de impresión, en primer lugar, necesitaba del grupo de operadores del programa de impresión; en segundo lugar, ya impresos los reportes pasaban al grupo de corte y clasificación; en tercer lugar, los reportes eran entregados al grupo de distribuidores y hasta el final, llegaban al usuario. Durante este proceso era muy fácil que personal no autorizado consultara los datos de diferentes dependencias, riesgo que anulaba el carácter confidencial de la información.

#### **9. Imposibilidad de manejo de imágenes**

Debido a que no existía la tecnología para transferir imágenes digitalizadas, el poder enviar oficios y/o circulares a las dependencias, representaba tener que sacar múltiples copias fotostáticas de los documentos originales, guardarlos en sobres y, posteriormente, hacer la distribución física a las 143 dependencias universitarias. Este proceso, además de tener un alto costo, ocasionaba importantes retrasos en la difusión de la información institucional.

#### **10. Imposibilidad para transferir archivos**

No obstante que desde hace algunos años se encuentra en funcionamiento el correo electrónico, el poder enviar archivos de grandes dimensiones de manera electrónica representaba dos problemas: el primero, que no en todas las dependencias se

disponía de una cuenta de correo electrónico y no todo el personal tenía la experiencia suficiente para usar este servicio. El segundo era que por limitantes propias de las aplicaciones actuales, era muy difícil mandar correos que contuvieran archivos mayores a 1 Mb, para lo cual era necesario utilizar el programa de transferencia de archivos FTP (File Transfer Program), con los inconvenientes que esto representaba, principalmente en lo que se refiere al manejo de la herramienta por usuarios con poca experiencia.

#### **4.1.2 Propuesta de Solución**

En esta fase el analista también, prepara una propuesta del sistema que resume todo lo que ha encontrado, presenta un análisis de las alternativas y plantea las recomendaciones de lo que deberá realizarse. En el área de sistemas, cada problema es único y en consecuencia nunca habrá sólo una solución correcta. La manera en como se plantea una solución depende de las características individuales de cada analista aunadas a su formación profesional.

Tomando en cuenta cada una de las situaciones descritas en la sección anterior, se consideró indispensable crear una solución integral que, por una parte, diera respuesta de manera eficiente y ágil a las necesidades observadas, y por otra, fuera acorde con las nuevas tendencias y tecnologías incorporadas recientemente a la Universidad.

Por tal motivo, surgió el proyecto de crear un sistema para el manejo de información por computadora, a través del cual todas las dependencias universitarias pudieran consultar de manera ágil y sencilla información presupuestal y de gestión administrativa, desde la comodidad de su computadora personal, teniendo la capacidad de visualizar los datos en pantalla, guardarlos en

archivos de su computadora personal e imprimirlos de manera selectiva; aprovechando en su totalidad la infraestructura de la red de cómputo y comunicaciones con la que cuenta la Universidad (Red UNAM). La figura 4.1 muestra un enfoque global de lo que pretende el sistema.

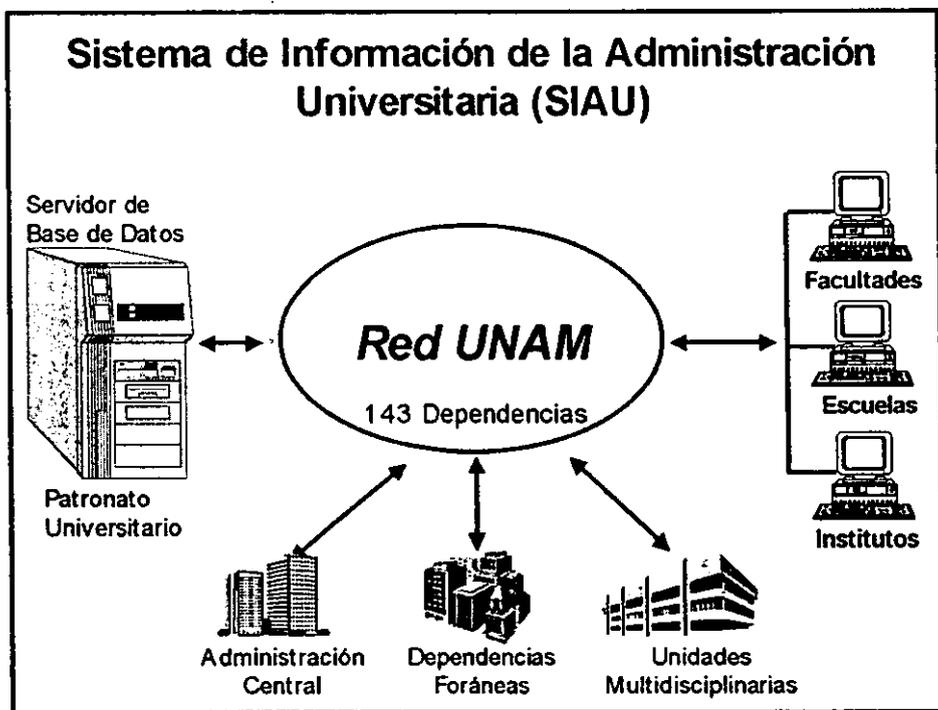


Figura 4.1 Propuesta de Solución

Cabe mencionar que una primera propuesta del sistema tuvo por nombre “Sistema Integral de Consulta” (SIC), el cual originalmente fue creado para agilizar la difusión de la información emitida por Patronato Universitario hacia el resto de las dependencias. Sin embargo, al cabo de diez meses de operación, la aplicación adquirió un carácter institucional, para lo cual se le agregaron algunos módulos y se le asignó el nuevo nombre de “Sistema de Información de la Administración

Universitaria" (SIAU).

Ahora bien, el SIAU debía contar con una serie de características para dar solución a la problemática observada. El cuadro que aparece a continuación presenta cada una de las características que se requiere tenga el sistema y, en párrafos posteriores, se hace la descripción correspondiente.

No.	Característica
1.	Arquitectura Cliente/Servidor
2.	Soporte para 400 a 600 Usuarios
3.	Diferentes Niveles de Acceso
4.	Seguridad de la Información
5.	Fácil Manejo
6.	Impresión Total o Selectiva.
7.	Respaldo de Información en Disco Duro o Disquetes
8.	Conexión Vía Red UNAM
9.	Manejo de Imágenes
10.	Transferencia de Archivos y/o Aplicaciones
11.	Información General de Patronato y Secretaría Administrativa
12	Diseño Modular

## **1. Arquitectura Cliente/Servidor**

El sistema debía ser diseñado bajo una plataforma cliente/servidor de tal manera que se lograra la correcta distribución de las funciones y aprovechara al máximo la potencialidad del servidor y la interfaz gráfica de cada cliente. Esto es, todos los datos debían residir en el servidor, de tal manera que cuando un cliente hiciera una solicitud de información, el servidor, de capacidad mayor, debía realizar la operación, mandando al cliente únicamente los datos seleccionados de acuerdo con la solicitud. Una vez que los datos se mostraran en la pantalla del cliente, el usuario tendría la posibilidad de guardarlos en un archivo propio, o bien, imprimirlos de acuerdo con sus necesidades.

## **2. Soporte para 400 a 600 usuarios**

Otra variable que debía ser considerada, es la cantidad de usuarios a los que daría servicio el sistema. Para este cálculo, se debía considerar, por una parte, que el número de dependencias y subdependencias que existen en la UNAM es de aproximadamente 253 y, por otra, que de cada dependencia podrían tener acceso al sistema: el director, el secretario administrativo y los jefes de contabilidad, presupuesto e ingresos extraordinarios. Con base en dicho cálculo, se estimó que la cantidad de personal que tendría acceso al sistema debía fluctuar entre los 400 y 600 usuarios. Por tanto, el hardware y software utilizado debía contar con la suficiente capacidad para dar servicio a esta cantidad de usuarios.

## **3. Diferentes Niveles de Acceso**

El Sistema debería contar con los suficientes mecanismos de control de acceso a la información, resguardando en la medida de lo posible el nivel de confidencialidad que requieren los datos manejados. Esto es, se debía contar con una clasificación de

usuarios de acuerdo con los niveles de autoridad y alcances del puesto que desempeñan. De esta manera, los mandos superiores (Rector, Tesorero, Secretario Administrativo, Secretario de Planeación y algunos directores de dependencias centralizadoras, como las Direcciones Generales de Programación Presupuestal y de Control e Informática, entre otros), podrían tener acceso a la información a nivel global, es decir, aquella que corresponde a todas las dependencias y/o subdependencias de la Institución. Un siguiente grupo (directores de facultades, escuelas e institutos), sólo podría consultar los datos correspondientes a su dependencia y subdependencias que la conforman. Un tercer grupo (subdirectores y secretarios administrativos), tendría acceso únicamente a la información de su subdependencia; y, por último, un grupo especial formado por dependencias centralizadoras (como las direcciones generales de Bibliotecas, Obras, Personal y Cómputo Académico entre otras), que podrían consultar ciertas partidas a nivel institucional. Lo anterior se representa en la Figura 4.2.

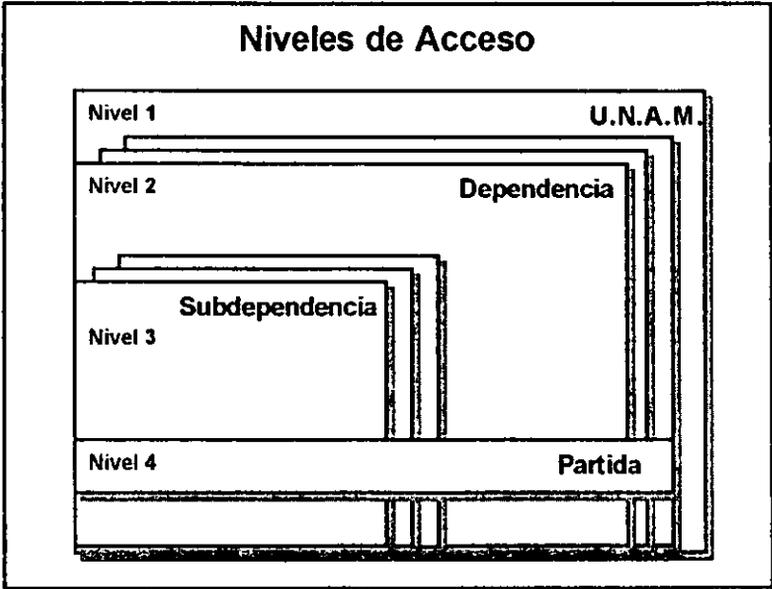


Figura 4.2 Diferentes Niveles de Acceso a la información

#### **4. Seguridad de la Información**

Otro aspecto relevante que se consideró dentro del diseño corresponde a la seguridad de la información manejada por el sistema. Tomando en cuenta que el sistema debía correr sobre la Red-UNAM, la información de la base de datos tendría el riesgo de quedar al alcance de profesores, alumnos e investigadores conectados a la red, e incluso algunos otros usuarios que trabajaran sobre Internet. Por tal motivo, el diseño debía incluir suficientes mecanismos de seguridad que permitieran el acceso a la información, exclusivamente a usuarios autorizados, evitando la intromisión de personas ajenas.

#### **5. Fácil Manejo**

La aplicación debía ser de fácil manejo para el usuario, ya que en la mayoría de los casos el personal que tendría derecho a su utilización, no cuenta con los suficientes conceptos informáticos para manejar una herramienta complicada. Por tanto, se consideró como opción utilizar una interfaz gráfica (ambiente Windows) basada en menús de acceso rápido y una serie de mensajes de ayuda en línea, para que con el uso de un conjunto reducido de teclas y del ratón, los usuarios pudieran familiarizarse rápidamente con el manejo del sistema.

#### **6. Impresión Total o Selectiva**

La aplicación debía brindar la opción de impresión total o selectiva de la información consultada en pantalla, permitiendo al usuario que así lo requiriera, contar con el soporte documental, que sería la equivalencia de los reportes que recibía con el sistema anterior. Adicionalmente, el diseño debía contener diferentes criterios de búsqueda de los datos, para que el usuario selectivamente eligiera el que más se apegara a sus requerimientos, sin la necesidad de recurrir a grandes

volúmenes de información como lo realizaba anteriormente.

## **7. Respaldo de Información en Disco Duro o Disquetes**

Como ventaja adicional, la aplicación debía permitir guardar en archivos propios la información visualizada en pantalla, teniendo la posibilidad de almacenarlos en el disco duro de la computadora del usuario, o bien, en disquetes. Básicamente, se pensó en utilizar dos tipos de formato estándares para todas las aplicaciones: (\*.TXT) para archivos ASCII, que pueden ser manejados en cualquier procesador de texto; y, (\*.DBF) para archivos de bases de datos, que pueden ser utilizados con los paquetes de uso común dentro de la Universidad como el Dbase, Clipper, FoxPro, Excel, Paradox, etc.

## **8. Conexión vía Red UNAM**

Una característica especial, y quizá la más importante, es que el sistema debía correr sobre todas las microcomputadoras cliente conectadas a Red UNAM, independientemente del software que se tuviera cargado sobre cada una de ellas, siempre y cuando utilizara el ambiente gráfico Windows en cualquiera de sus versiones (3.1, 3.11, Windows 95/98 ó Windows NT). Por tal motivo, tanto en el cliente como en el servidor se debía utilizar el protocolo de comunicación estándar dentro de la UNAM (TCP/IP) de tal manera que garantizara la conexión de usuarios en Ciudad Universitaria, en la Zona Metropolitana, en el interior del país e incluso mas allá de sus fronteras. Y para aquellas dependencias que aún no contaran con conexión directa a Red UNAM, también tendrían que disponerse los mecanismos para que se pudieran conectar vía Módem.

## **9. Manejo de Imágenes**

Adicionalmente, el sistema debía contar con la posibilidad de consultar imágenes de oficios y circulares emitidos por la Administración Central, Patronato Universitario o cualquier otra dependencia que quisiera difundir a toda la Institución información relevante, con lo cual se pretendía constituir un medio de transmisión de información ágil, sencillo y eficiente

## **10. Transferencia de Archivos y/o Aplicaciones**

El Sistema debía brindar la posibilidad de transferir aplicaciones completas, es decir, hacer una especie de FTP, sin la complejidad que un verdadero FTP implica; con objeto de realizar actualizaciones más sencillas y facilitar la copia de archivos de manera remota.

## **11. Información General de Patronato y Secretaría Administrativa**

La aplicación debía proporcionar información relevante de las dependencias emisoras, tal como sus antecedentes, funciones y estructura, entre otros datos, además, debía contener un directorio de funcionarios, que comprendiera información de identificación como: cargo, teléfono, fax, correo electrónico, ubicación de oficina, etc.

## **12. Diseño Modular**

Se consideró conveniente que el sistema tuviera una estructura modular, a fin de que se dejara abierta la posibilidad de incrementar paulatinamente la aplicación: es decir, el diseño modular permitiría evitar el estancamiento y obsolescencia de la aplicación, favoreciendo la integración de nuevas funciones conforme fueran surgiendo más necesidades de información del Patronato o de la Administración

Central, incrementando sus opciones sin importando incluso el hardware y software disponibles y el lugar en donde residan las bases de datos de la información a compartir.

Sin lugar a dudas, el sistema propuesto no era una tarea fácil, pero definitivamente correspondía a lo que se necesitaba para que el Patronato y otras dependencias, mejoraran el servicio ofrecido a la comunidad universitaria en todos sus niveles. Tomando en consideración su relevancia, el desarrollo de la aplicación constituyó el objetivo del presente trabajo.

#### **4.1.3 Metodología**

Existen diferentes métodos que se pueden aplicar para el desarrollo de software, sin embargo, estos varían dependiendo del modelo de ingeniería que se aplique. Entre los más conocidos se pueden mencionar los siguientes:

- Modelo Lineal Secuencial
- Modelo de Construcción de Prototipos
- Modelo DRA (Desarrollo Rápido de Aplicaciones)
- Modelo Incremental
- Modelo en Espiral

Sin duda alguna, cada uno de ellos tiene sus ventajas y desventajas de acuerdo con los factores que intervienen para el desarrollo de un sistema (los recursos materiales y humanos, tiempos de entrega, dominio de las herramientas, etc.). Para el desarrollo del presente trabajo, se utilizó el Modelo en Espiral, ya que por sus características fue el que mejor se adaptó a las necesidades propias de Patronato,

principalmente porque permite la creación de nuevas versiones del software en periodos cortos y por su independencia en la creación de nuevos módulos.

#### 4.1.3.1 Modelo en Espiral

El Modelo en Espiral, propuesto por Boehm, es un modelo de proceso de software evolutivo que compagina la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. A través de dicho modelo se proporciona el potencial para el desarrollo rápido de versiones incrementales del software. (Ver Figura 4.3)

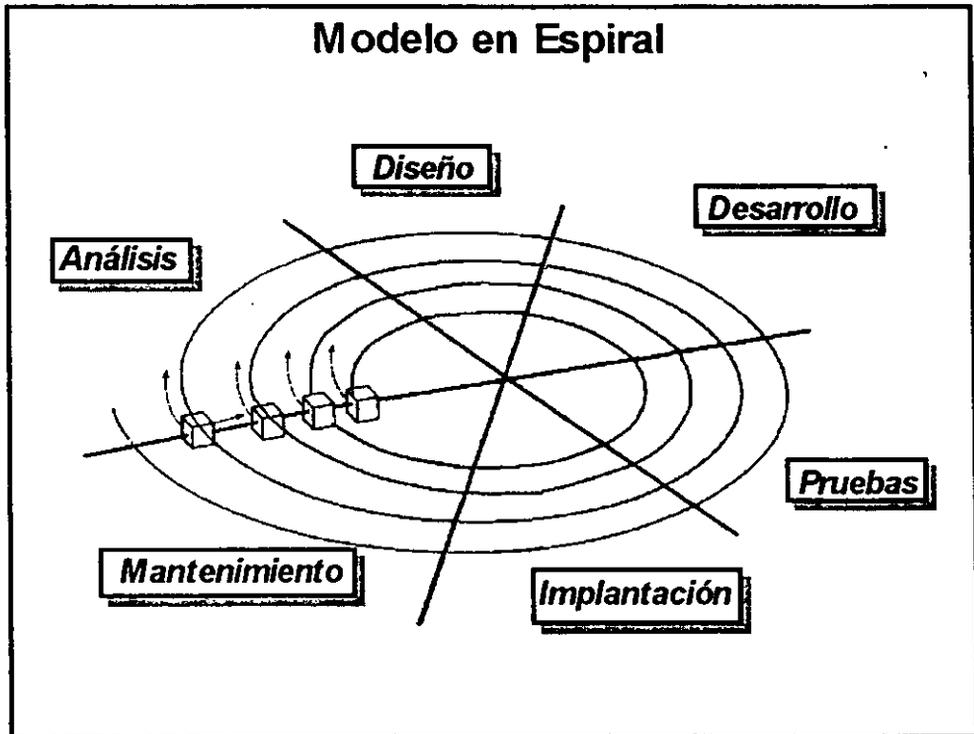


Figura 4.3 Modelo Seleccionado para el Desarrollo del Sistema

De acuerdo con este modelo, el software se desarrolla en una serie de versiones

cada una desarrollada con mayor amplitud de funciones y de capacidades. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas de ingeniería del sistema. El modelo en espiral como muestra la Figura 4.3 se divide en seis regiones.

Cuando empieza este proceso evolutivo, el equipo de ingeniería de software gira alrededor de la espiral en la dirección de las manecillas del reloj, comenzando por el centro. El primer circuito de la espiral produce el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones mas sofisticadas del software.

A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. En esencia la espiral permanece operativa hasta que el software se retira. Hay veces que el software esta inactivo, pero siempre que se inicie un cambio, el proceso arranca en el punto de entrada adecuado (por ejemplo, una mejora del producto).

El modelo en espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Como el software evoluciona, a medida que progresa el proceso, el desarrollador y el cliente comprenden y reaccionan mejor ante riesgos en cada uno de los niveles evolutivos. Este modelo utiliza la construcción de prototipos como mecanismo de reducción de riesgos, pero lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto. Esta opción mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de vida clásico, pero lo incorpora al marco de trabajo

interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Pero al igual que otros paradigmas, el modelo en espiral no es la panacea. Puede resultar difícil convencer a grandes clientes (particularmente en situaciones bajo contrato) de que el enfoque evolutivo es controlable. Requiere una considerable habilidad para la evaluación del riesgo, y cuenta con esta habilidad para el éxito. Si un riesgo importante no es descubierto y gestionado, indudablemente surgirán problemas. Finalmente, el modelo en sí mismo es relativamente nuevo y no se ha utilizado tanto como los paradigmas lineales secuenciales o de construcción de prototipos. Todavía tendrán que pasar muchos años antes de que se determine con absoluta certeza la eficacia de este nuevo e importante paradigma.

#### **4.1.4 Flujo de Datos**

A medida que la información se mueve a través del software, es modificada por una serie de transformaciones. El *diagrama de flujo de datos* (DFD) es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. En la figura 4.4 se muestra la forma básica de un diagrama de flujo de datos.

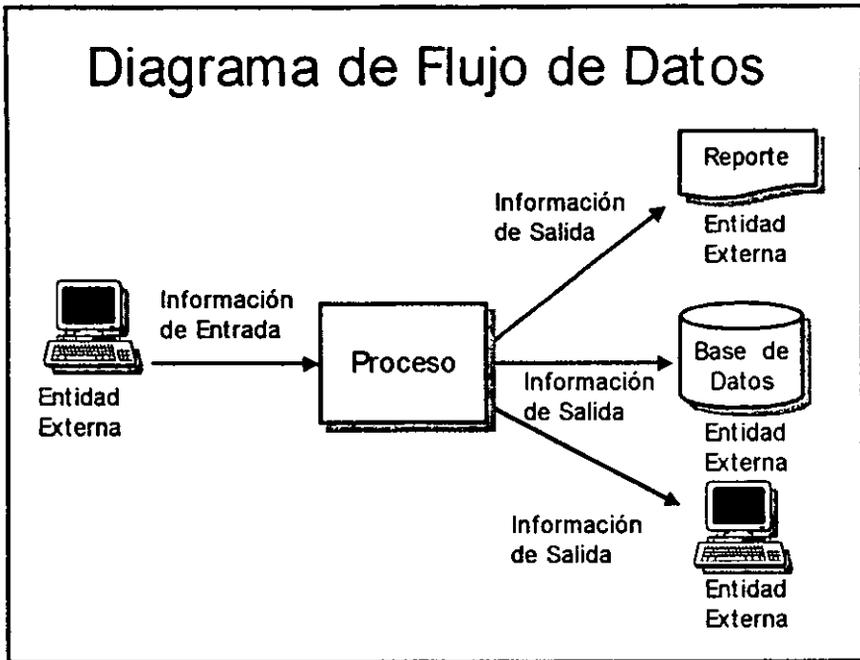


Figura 4.4 Forma Básica de un Diagrama de Flujo de Datos

Existe una notación básica que se usa para crear un DFD, en la figura 4.5 se muestra un ejemplo de ella. La **computadora** o el **símbolo de reporte** se usan para representar una entidad externa, es decir, un elemento del sistema (por ejemplo, el hardware, una persona, un reporte, otro programa), u otro sistema que produzca información a ser transformada por el software o que reciba información producida por el software. Un **rectángulo** representa un proceso o transformación que se aplica a los datos (o a control) y los cambia de alguna forma. Todas las flechas de un diagrama de flujo de datos indican la dirección de los datos desde su origen hasta su destino, y deben ser etiquetadas. El **cilindro** representa un almacén de datos (información almacenada que es utilizada por el software). La sencillez de la notación DFD es una de las razones por la que las técnicas de análisis estructurado son ampliamente utilizadas.

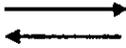
## Notación Básica



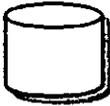
Representan el origen o destino de los datos; pueden ser pantallas, reportes o archivos.



Los procesos representan procedimientos o dispositivos que utilizan o producen datos.



Indican la dirección de los datos, desde su origen hasta su destino.



Representa un almacén de datos para guardarlos o hacer referencia a ellos a través de un proceso.

Figura 4.5 Notación Básica de un Diagrama de Flujo de Datos

Se puede usar el diagrama de flujo de datos para representar un sistema o un software a cualquier nivel de abstracción. De hecho, los DFD's pueden ser refinados en niveles que representen un mayor flujo de información y un mayor detalle funcional. Por consiguiente el DFD proporciona un mecanismo para el modelado funcional así como el modelado del flujo de información.

Un DFD de nivel 0 también es denominado *modelo de contexto*, y representa al elemento de software completo como un sólo proceso con datos de entrada y de salida representados por flechas de entrada y salida, respectivamente. Al particionar el DFD de nivel 0 para mostrar más detalles, aparecen representados procesos y caminos de flujo de información adicionales. Por ejemplo, un DFD de nivel 1 puede contener cinco o seis procesos con flechas interconectadas. Cada uno de los procesos representados en el nivel 1 es una subfunción del sistema general en el modelo de contexto.

En la Figura 4.6 se muestra el diagrama de contexto (nivel 0) del Sistema de Información de la Administración Universitaria (SIAU), donde se cuenta con tres módulos principales: el primero, relacionado con la carga de información, el cual importa información de diferentes bases de datos y archivos, le da formato y la inserta en las diferentes tablas de la base de datos central para su posterior consulta; en segundo lugar, se tiene el módulo de acceso y validación de usuarios, el cual se encarga de verificar los permisos y nivel de consulta de cada usuario al entrar al sistema; por último, el módulo más grande e importante de todos llamado SIAU, se encarga de proporcionar información dentro de los diferentes estratos que se manejan en el sistema para, finalmente, mostrar la información consultada bajo tres opciones: en pantalla, en reportes impresos y/o almacenamiento en disco duro o disquetes.

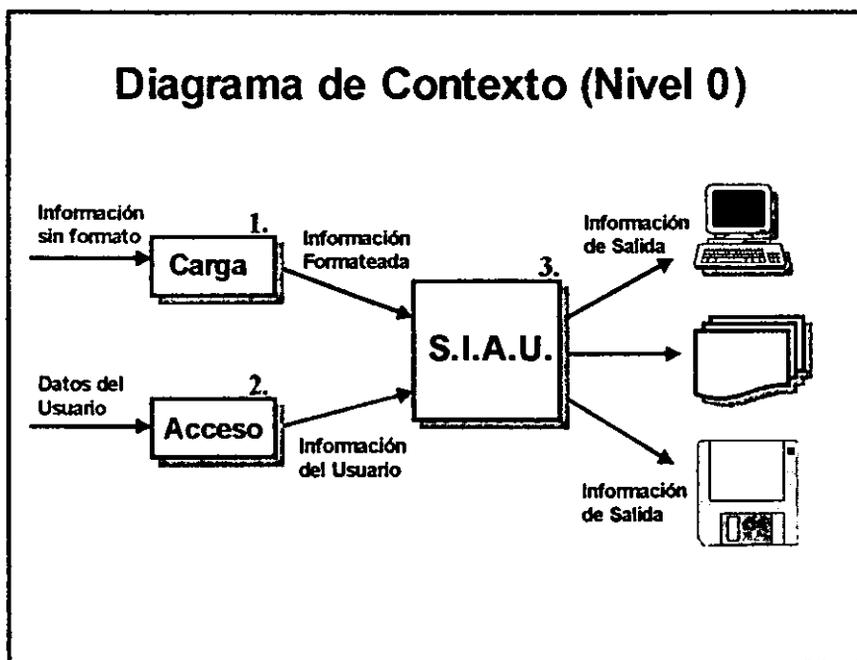


Figura 4.6 Diagrama de Contexto, Nivel 0 del S.I.A.U.

Como ya se mencionó, se puede refinar cada uno de los procesos en distintos niveles para mostrar un mayor detalle. Sin embargo, es importante mantener la continuidad del flujo de información, es decir, la entrada y la salida de cada refinamiento debe ser la misma. Este concepto es denominado *balanceo*, y es esencial para el desarrollo de modelos consistentes.

En los párrafos siguientes se presenta una breve descripción de los módulos que conforman el SIAU, con sus respectivas ilustraciones.

El módulo de *Carga*, ilustrado en la figura 4.7, tiene por función recopilar información de 4 fuentes diferentes, las primeras tres son tablas de bases de datos, mismas que tienen que pasar por un proceso de formateo y ordenamiento, para posteriormente ser insertados en la base de datos central del SIAU. La siguiente tiene que ver con archivos de texto e imágenes, los cuales tienen que pasar por un proceso de validación, para posteriormente ser almacenados.

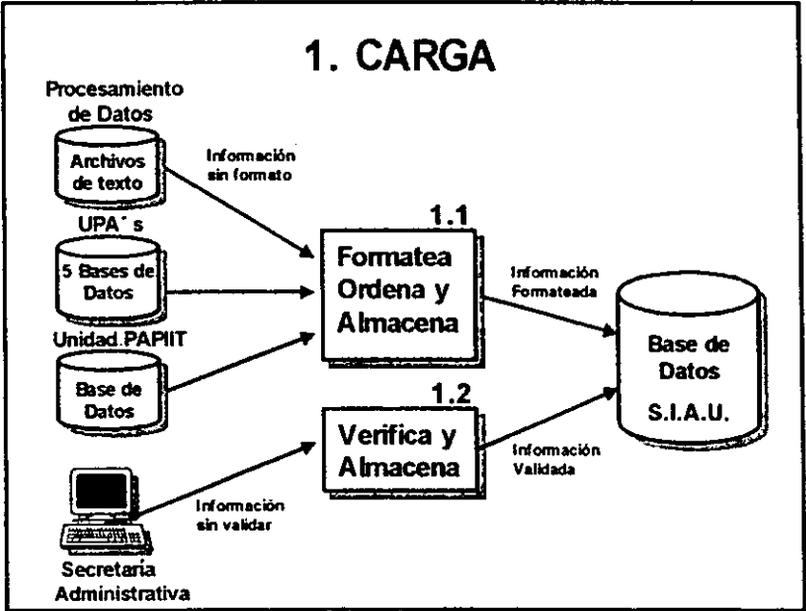


Figura 4.7 Módulo de Carga de Información

El módulo de *Acceso*, ilustrado en la figura 4.8, es el encargado de verificar que los datos de identificación proporcionados por el usuario al entrar al sistema sean válidos; además, genera el nivel de acceso a la información dentro del sistema.

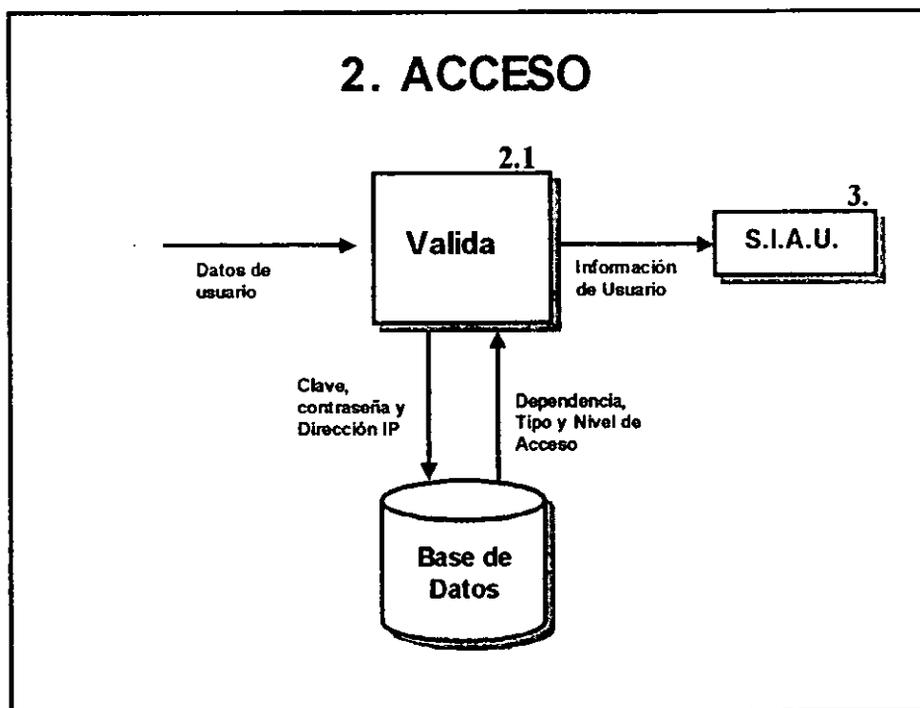


Figura 4.8 Módulo de Acceso al Sistema

Ahora bien, como se comentó anteriormente, el módulo principal y más grande del sistema es el llamado *SIAU*, este en su primera etapa se divide en cinco apartados, dentro de los que destaca el concerniente al *Patronato Universitario*, ya que ahí se concentra el 80% de la información del sistema, los apartados restantes contienen información de carácter general (ver Figura 4.9 de la siguiente página).

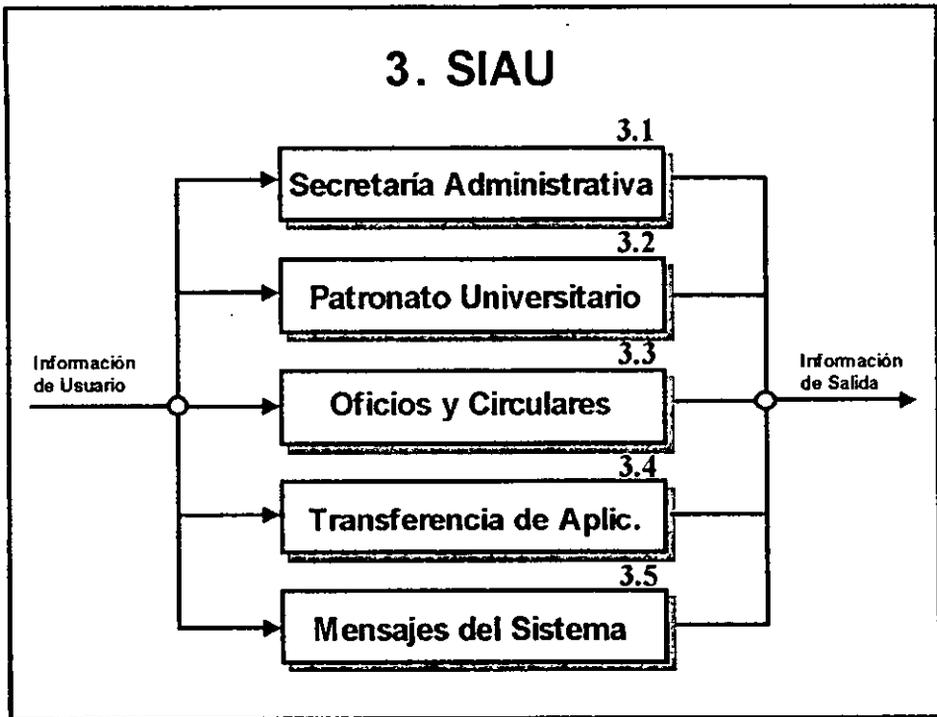


Figura 4.9 Módulo Principal del Sistema

El módulo de *Secretaría Administrativa* muestra información con carácter general referente a dicha dependencia y todas las direcciones que la integran, por ejemplo, incluye los antecedentes, funciones, organigramas y el directorio de funcionarios correspondiente.

El módulo de *Patronato Universitario*, que de hecho fue el que dio origen al sistema anterior conocido como Sistema Integral de Consulta (SIC), concentra información presupuestal y de gestión administrativa concerniente a toda la Institución.

El módulo de *Oficios y Circulares*, permite a los usuarios consultar imágenes digitalizadas de los oficios y circulares generados por Patronato y/o la Secretaría Administrativa; además, está dotado de un algoritmo de búsqueda por diversos criterios, de tal manera que el oficio o circular desplegará, además de su contenido, la firma del funcionario correspondiente.

El módulo de *Transferencia de Aplicaciones*, permite la transportación electrónica de todo tipo de archivos. En esta opción las dependencias recibirán documentos oficiales de extensión considerable y programas de cómputo, tales como futuras actualizaciones del Sistema de Información de la Administración Universitaria (SIAU), o bien, programas correspondientes a aplicaciones administrativas desarrolladas por la Dirección General de Normatividad y Sistemas Administrativos.

El módulo de *Mensajes del Sistema*, permite al usuario conocer información sobre la emisión de nuevos oficios y/o circulares, así como la disposición de nuevos archivos para su transferencia, la generación de nuevas versiones, así como la suspensión del servicio ya sea por problemas de red, o por mantenimiento del sistema, entre otros importantes mensajes.

Para efectos de presentación y tomando en consideración su prominente importancia en el análisis realizado, a partir de este punto la explicación se enfoca en el Módulo correspondiente al *Patronato Universitario*.

En una primera aproximación, el Módulo de Patronato Universitario se divide en 13 submódulos, los cuales se muestran en la figura 4.10 de la siguiente página.



Figura 4.10 Módulo de Patronato Universitario

Los *Compromisos Previos* son apartados de dinero para la adquisición de bienes, éstos se manejan de manera independiente al presupuesto ya que están comprometidos, un ejemplo puede ser el pago de luz o cualquier otro servicio de alguna dependencia. En éste módulo se puede consultar bajo los criterios de número de previo, cuenta, código, partida, grupo, programa, dependencia y subdependencia. La información solicitada se envía a través de un procedimiento almacenado a la base de datos del servidor, y sólo se muestran en pantalla los registros que cumplieron con el criterio de búsqueda seleccionado. Posteriormente la información puede ser impresa o guardada en disco.

Los *Gastos a Reserva de Comprobar* son erogaciones que se registran sin soporte documental, siendo posible presentar la comprobación correspondiente después de un periodo determinado. Estos gastos se utilizan básicamente para amparar viajes a eventos educativos y culturales, que realizan profesores, investigadores, alumnos y personal administrativo. Los criterios de búsqueda bajo los cuales se puede consultar en este módulo son: Viáticos, Trabajos de Campo, Profesores Visitantes, Gastos Generales, R.F.C., número de Folio y Consulta General. Una vez identificada, la información solicitada se envía a través de un procedimiento almacenado a la base de datos del servidor, y sólo se muestran en pantalla los registros que cumplieron con el criterio de búsqueda seleccionado. Si el usuario así lo desea, una vez que el sistema presenta los datos en su pantalla, puede imprimir o guardar la información en el disco duro de su PC.

La *Consulta Presupuestal* permite seleccionar los datos puntuales del presupuesto al cierre del día anterior, ofreciendo para ello, diferentes tipos de consulta, que pueden ir desde la más sencilla, hasta otra con varios niveles de agregación. A su vez, este módulo consta de otros ocho submódulos: los primeros tres, corresponden al Presupuesto Operativo; los siguientes tres son de los Programas Especiales; mientras que los últimos dos, permiten consultar por separado información referente a Partidas de Ejercicio Directas y Partidas Centralizadas. Al igual que en el resto de los módulos, la información consultada puede ser impresa o guardada en el disco duro de la PC, si así se desea.

El *Ejercicio Presupuestal* muestra los saldos al cierre del mes anterior, los movimientos realizados a la fecha de consulta y los saldos actuales según dichos movimientos.

Los *Apoyos* son recursos generados por aquellas dependencias que obtienen ingresos extraordinarios a través de diversas actividades y/o proyectos de su quehacer cotidiano, y que ofrecen en apoyo a las dependencias que no los obtienen. Los criterios por los cuales se puede consultar en éste módulo son: Código, Grupo, Partida, Dependencia y Subdependencia. Como en los otros módulos, la información consultada puede ser impresa o guardada en disco duro o disquete.

El módulo de *Transferencias Presupuestales* se refiere a los movimientos de cambio de asignación presupuestal, es decir, transferencias de fondos entre códigos programáticos efectuadas por las dependencias y que ya han sido aceptados y/o rechazados por la Dirección General de Programación Presupuestal. En este módulo sólo se pueden consultar las operaciones realizadas de enero a diciembre del año en curso, existiendo la posibilidad de posteriormente imprimir o guardar los resultados.

Los *Ingresos Extraordinarios* son recursos generados durante el mes inmediato anterior por cada una de las dependencias, de acuerdo con la tasa de interés aplicada en el periodo. La búsqueda de información bajo este módulo sólo se puede realizar por Dependencia y Subdependencia, y los resultados mostrados en pantalla pueden también ser guardados o impresos.

En el módulo de *Estado de Cuenta de Ingresos Extraordinarios* se muestran los saldos al cierre del mes anterior, los movimientos realizados a la fecha de consulta y los saldos actuales según dichos movimientos. Los criterios por los cuales se puede buscar información en este módulo son: Cuenta, Folio, Fecha, Dependencia y Subdependencia. De igual manera que en los otros módulos, la información consultada puede ser impresa o guardada en disco.

El módulo de *Pólizas* se refiere a los ajustes que la Contaduría General ha realizado al gasto de cada una de las dependencias. Las pólizas son consideradas como documentos contabilizadores para el registro. Este módulo es de gran ayuda para las dependencias, ya que les permite conocer movimientos que no fueron realizados por ellas mismas, sino desde la Contaduría General de la UNAM. La búsqueda de información se puede hacer por: Folio, Fecha, Previo, Dependencia y Subdependencia. Los datos obtenidos derivados de las consultas pueden ser guardados o impresos.

Para el caso de *Folios en UPA's* (Unidades de Proceso Administrativo), la dependencia podrá consultar el estado que guardan los trámites ingresados a las UPA's incluyendo aquellos del día inmediato anterior, identificando si el movimiento fue rechazado y las causas por las que no fue aceptado, o bien, si el movimiento fue aceptado, se muestra el número de cheque que se generó y el nombre del beneficiario con el que se expidió. Los resultados obtenidos, también pueden ser almacenados o impresos.

El módulo de *Boletos de Avión* muestra los gastos totales generados por concepto de viajes y viáticos de investigadores, profesores y demás personal académico y administrativo durante el último mes. Las opciones por las cuales se puede buscar información son: Número de Folio, Boleto, Dependencia y Subdependencia, o bien, por R.F.C., Línea/Agencia y Tipo de Servicio, así como una Consulta General. Al igual que en el resto de los módulos, la información puede ser impresa o guardada en el disco duro de la PC.

El módulo correspondiente a *Proyectos Académicos para la Investigación e Innovación Tecnológica (PAPIIT)*, se descompone en 9 submódulos que son: Resumen de Proyectos, Presupuesto, Super Cray (Ministraciones Entregadas),

Movimientos Presupuestales, Honorarios Proyectos, Becarios Proyectos, Boletos de Avión, Otros Servicios y Compromisos Previos. En todos ellos se muestra información para los titulares de dependencias, unidades administrativas y para los investigadores responsables de proyectos, referente a datos generales del proyecto; presupuesto asignado, movimientos presupuestales, honorarios, becarios, boletos de avión y compromisos previos creados con recursos P.A.P.I.I.T. en los últimos cinco años, pudiendo observar a detalle, información diaria de la situación general de todos los movimientos realizados durante la vida de cada proyecto. Para todos los submódulos se tiene la posibilidad de imprimir la información consultada en pantalla, así como el guardado en archivos.

Por último, el módulo de *Depósitos del Consejo Nacional de Ciencia y Tecnología (CONACYT)* permite a las dependencias mantener informados a sus investigadores o alumnos de las cuentas, montos y fechas en que se realizaron los depósitos correspondientes durante la duración de la beca. En éste módulo se puede consultar bajo los criterios de: Número de Oficio, Fecha de Oficio, Importe, Fecha de Depósito, Dependencia y Subdependencia, además de una Consulta General. Los resultados de las búsquedas tienen el mismo tratamiento que en los demás módulos.

El siguiente módulo a detallar dentro de este análisis, es el referente a *Ejercido Presupuestal*, debido a su importancia y a su gran similitud con el resto de todos los módulos del sistema. La Figura 4.11 muestra a detalle como está constituido dicho módulo, y al mismo tiempo sirve como referencia general para saber cómo están contruidos el resto de los módulos dentro del sistema.

### 3.2.4 Ejercicio Presupuestal

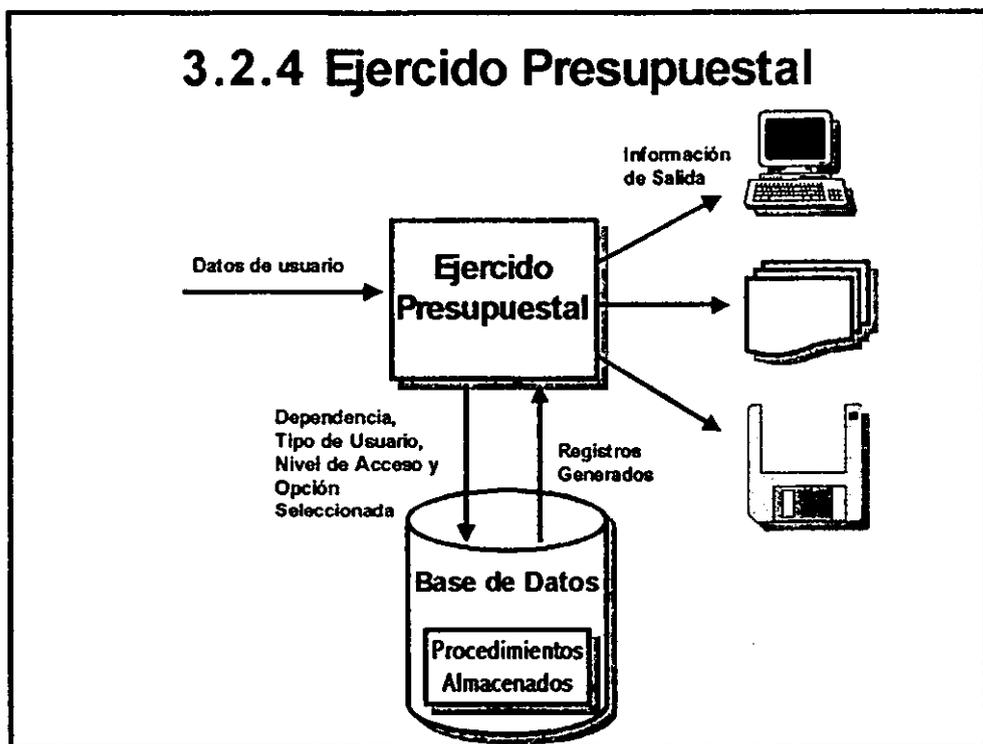


Figura 4.11 Módulo de Ejercicio Presupuestal

En este módulo una vez recibidos los datos generales del usuario, se podrá consultar información presupuestal bajo alguno de los siguientes criterios:

- Número de Código Presupuestal
- Número de Dependencia
- Número de Subdependencia
- Número de Grupo de Gasto
- Número de Partida de Gasto
- Número de Subprograma

Posteriormente, la opción seleccionada, junto con la información de tipo de usuario, clave de dependencia y nivel de acceso, es enviada al servidor de base de datos donde se ejecuta un procedimiento almacenado bajo los parámetros indicados; este último procesa la búsqueda y regresa al cliente sólo los registros necesarios de acuerdo a la consulta realizada, mismos que son desplegados en pantalla, para finalmente ser almacenados o impresos de acuerdo con las necesidades del usuario.

El diagrama de flujo de datos es una herramienta gráfica que puede ser muy valiosa durante el análisis de requisitos del software, representa el flujo de la información sin representación explícita de la lógica de procesamiento (por ejemplo: condiciones o bucles). La notación básica que se usa para desarrollar un DFD no es en sí misma suficiente para describir los requisitos del software. Por ejemplo, una flecha de un DFD representa un objeto de dato que entra o sale de un proceso. Un almacén de datos representa alguna colección organizada de datos. Pero, ¿Cuál es el contenido de los datos implicados en las flechas o en el almacén? Si la flecha (o el almacén) representa una colección de objetos, ¿Cuáles son? Para responder a estas preguntas, aplicamos otro componente de la notación básica del análisis estructurado, este componente es el *diccionario de datos*, el cual describiremos posteriormente.

Por otro lado, es importante señalar que el diagrama no proporciona ninguna indicación explícita de la secuencia del procesamiento. El procedimiento o la secuencia pueden estar implícitamente en el diagrama, pero la representación procedimental explícita generalmente queda pendiente hasta el diseño del software.

#### 4.1.5 Diagrama Entidad-Relación

El Diagrama Entidad-Relación (DER) fue propuesto originalmente por Peter Chen para el diseño de sistemas de bases de datos relacionales y ha sido ampliado por otros autores. En éste, se identifica un conjunto de componentes primarios que son: objetos de datos (entidades), relaciones y atributos. El propósito principal del DER es representar entidades y sus relaciones. La notación DER se maneja de la siguiente manera:

**Entidades** Se representan por medio de un rectángulo en el diagrama. Las entidades son una colección o conjunto de objetos reales o abstractos (personas, lugares, eventos, etc.) que tienen atributos comunes y que juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única.

**Relaciones** Una relación es una asociación entre dos entidades y representa un conjunto de conexiones entre objetos, instancias o entidades. Las relaciones se representan a través de líneas. Una vez establecidas las relaciones se evalúa cada extremo para determinar la cardinalidad, la cual puede ser: Uno a Uno (1:1), Uno a Muchos (1:M) ó Muchos a Muchos (M:M).

Para el sistema en cuestión, la gran mayoría de las entidades son independientes debido a que cada módulo puede ser tratado como un pequeño sistema y la unión de todos ellos forman el sistema principal. De tal manera que el diagrama entidad correspondiente al SIAU se encuentra dividido en tres secciones, mismas que se describen a continuación.

La primera de ellas, representada en la figura 4.12, engloba a todas las entidades que conforman el módulo de Patronato Universitario.

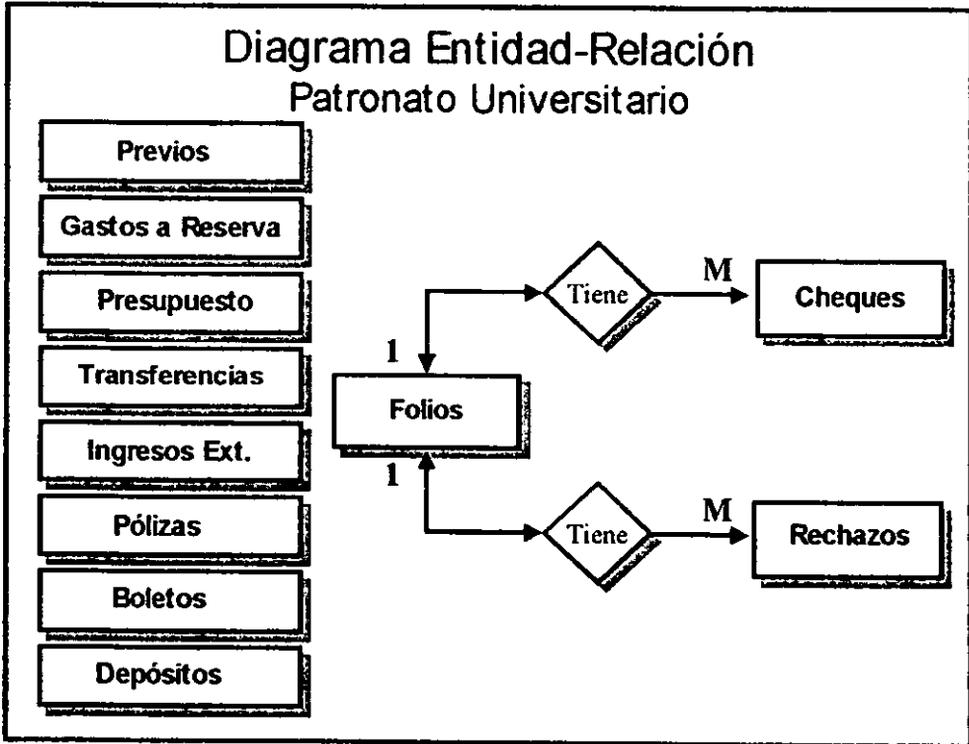


Figura 4.12 Diagrama Entidad-Relación (Primera Parte)

Como se puede observar en esta figura, la única entidad que presenta relaciones es *Folios* con *Cheques* y *Rechazos*.

Para el caso de Proyectos PAPIIT ilustrados en la figura 4.13 no existe ninguna relación entre todas sus entidades, por lo que este módulo se puede considerar como un subsistema del sistema principal.

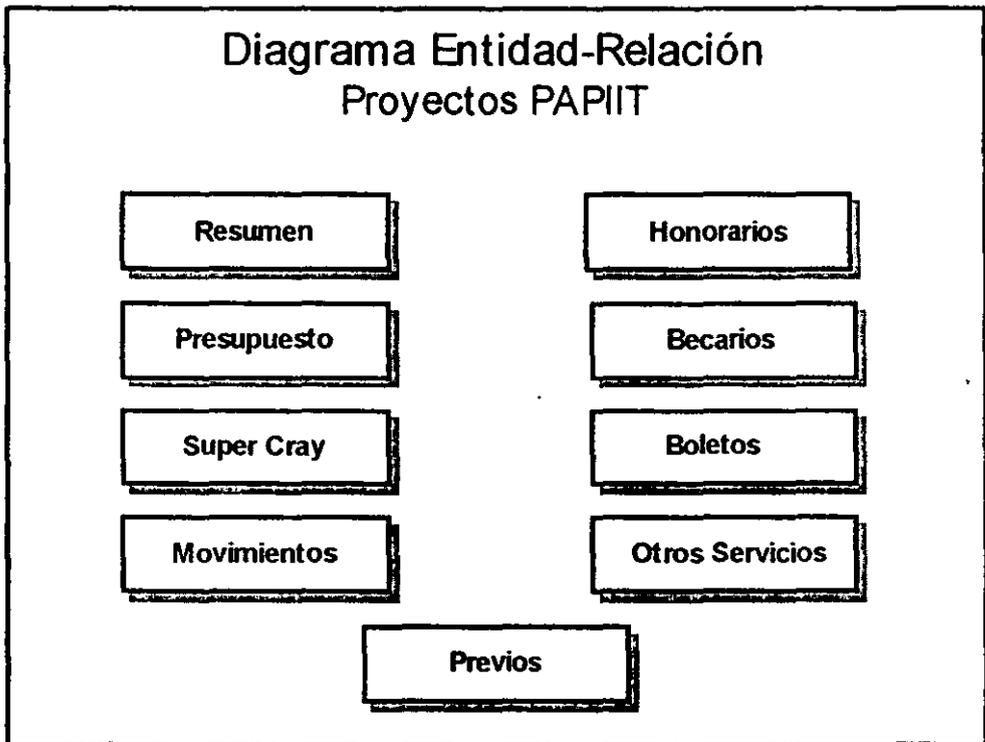


Figura 4.13 Diagrama Entidad-Relación (Segunda Parte)

Por último, en lo referente al módulo de Secretaría Administrativa ilustrado en la figura 4.14, las únicas entidades que tienen relaciones son *Usuarios* y *Circulares* con *Mensajes* e *Imágenes*, respectivamente.

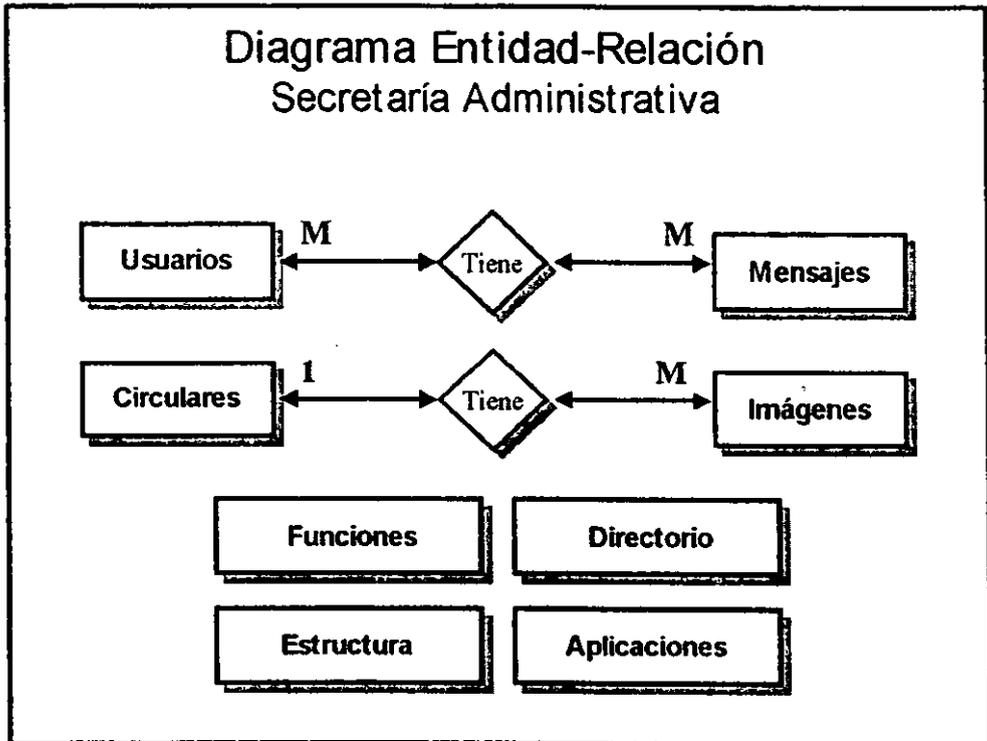


Figura 4.14 Diagrama Entidad-Relación (Tercera Parte)

El modelado de datos y el diagrama entidad-relación, proporcionan al analista una notación concisa para examinar datos dentro del contexto de una aplicación de procesamiento de datos. En la mayoría de los casos, el enfoque del modelado de datos se utiliza para crear una parte del modelo de análisis, pero también se puede utilizar para el diseño de bases de datos y para soportar cualquier otro método de análisis de requisitos.

#### 4.1.6 Diccionario de Datos

El modelo de análisis acompaña representaciones de objetos de datos, funciones y control. En cada representación los objetos de datos y/o elementos de control desempeñan un papel importante.

Por consiguiente, es necesario proporcionar un enfoque organizado para representar las características de cada objeto de datos y elemento de control. Esto se realiza con el diccionario de datos.

Se ha propuesto el diccionario de datos como gramática casi formal para describir el contenido de los objetos definidos durante el análisis estructurado. Esta importante notación de modelado ha sido definida como especialidad dentro de los diccionarios de referencia que utilizamos en la vida diaria. El diccionario es una referencia de “datos acerca de los datos” (esto es, metadatos) recopilados por el analista de sistemas para guiarse durante el análisis y el diseño. Como documento, recopila, coordina y confirma lo que un término específico significa para la gente de la organización.

Los analistas de sistemas deben estar conscientes y catalogar los diversos términos que se refieren al mismo dato. Esto evitará duplicar esfuerzos, favoreciendo una mejor comunicación entre los departamentos de la organización, que comparten una misma base de datos y, a la vez, facilitan el mantenimiento. El diccionario de datos sirve también como estándar consciente de los datos elementales.

Para el analista de sistemas es muy importante conocer qué tipos de datos componen el diccionario de datos, las convenciones utilizadas en ellos y la manera en que se desarrollan, de tal manera que le permita concebir el sistema y cómo es

que éste trabaja.

### **Datos que contiene el diccionario de datos**

Una manera de saber lo que debe contener el diccionario de datos, es visualizar cómo llegará a utilizarse. Es el elemento básico de referencia para localizar los nombres y atributos de los datos utilizados en todo el sistema de la organización. De hecho, deberá actualizarse cada vez que se hagan cambios o adiciones al sistema.

Con el fin de ser de utilidad, los registros del diccionario de datos deben contener información referente a las categorías siguientes:

- El nombre y el sinónimo del dato
- La descripción del dato
- El tipo del dato
- La longitud disponible en caracteres o bytes

Cuando el diccionario de datos se integra de manera correcta, es útil para el desarrollo, modificación y mantenimiento del sistema.

Para el caso del SIAU, el diccionario de datos está constituido principalmente por los atributos de las entidades (tablas) definidas en la sección previa. En las páginas siguientes, se presentan a manera de ejemplo las tablas que conforman el sistema:

## Previos

Campo	Nombre	Tipo	Longitud	Descripción
1	Previo	Char	6	Número de previo asignado (PK)
2	Dependencia	Char	3	Clave de la dependencia (PK)
3	SubDep	Char	2	Clave de la subdependencia (PK)
4	Grupo	Char	1	Clave del grupo de gasto (PK)
5	ComPart	Char	2	Clave del complemento de partida (PK)
6	Programa	Char	2	Clave del programa de gasto (PK)
7	SubProg	Char	2	Clave del subprograma de gasto (PK)
8	ImpPrev	Money	8	Importe del previo
9	SaldoPrev	Money	8	Saldo actual del previo
10	PolCrea	Char	13	Póliza de creación del previo
11	FechaCrea	Smalldate time	4	Fecha de creación del previo
12	PolAfec	Char	13	Póliza de afectación del previo
13	FechaAfec	Smalldate time	4	Fecha de la última afectación del previo

## GastosReserva

Campo	Nombre	Tipo	Longitud	Descripción
1	CtaGas	Char	3	Tipo de Cuenta de gasto (PK)
2	Dependencia	Char	3	Clave de la dependencia (PK)

### GastosReserva (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
3	SubDep	Char	2	Clave de la subdependencia (PK)
4	RfcResp	Char	10	RFC del responsable del gasto (PK)
5	PolCrea	Char	13	Póliza de creación del gasto
6	FechaCrea	Smalldate time	4	Fecha de creación del gasto
7	NomResp	Char	30	Nombre del responsable del gasto
8	ImpCrea	Money	8	Importe de creación
9	ImpComp	Money	8	Importe de comprobación
10	PolComp	Char	13	Póliza de comprobación
11	FechaComp	Smalldate time	4	Fecha de comprobación
12	FechaLim	Smalldate time	4	Fecha límite de comprobación
13	Mes	Char	2	Mes de proceso

**CodigosMen:** utilizada para la consulta del Presupuesto

Campo	Nombre	Tipo	Longitud	Descripción
1	Dependencia	Char	3	Clave de la dependencia (PK)
2	SubDep	Char	2	Clave de la subdependencia (PK)
3	Grupo	Char	1	Clave del grupo de gasto (PK)

## CodigosMen (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
4	ComPart	Char	2	Clave del complemento de partida (PK)
5	Programa	Char	2	Clave del programa de gasto (PK)
6	SubProg	Char	2	Clave del subprograma de gasto (PK)
7	AsigOrig	Money	8	Importe de la asignación original
8	Autorizado	Money	8	Importe del autorizado
9	Disponible	Money	8	Importe del disponible
10	ComPrev	Money	8	Importe del compromiso previo
11	PagEjer	Money	8	Importe ejercido
12	Acum811	Money	8	Importe de la partida 811
13	Acum911	Money	8	Importe de partida 911
14	IngExt	Money	8	Importe de ingresos extraordinarios
15	AntAsigOrig	Money	8	Importe de la asignación original del mes inmediato anterior
16	AntAutorizado	Money	8	Importe del autorizado del mes inmediato anterior
17	AntDisponible	Money	8	Importe del disponible del mes inmediato anterior
18	AntComPrev	Money	8	Importe del compromiso previo del mes inmediato anterior
19	AntPagEjer	Money	8	Importe ejercido del mes inmediato anterior
20	AntAcum811	Money	8	Importe de la partida 811 del mes inmediato anterior

## CodigosMen (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
21	AntAcum911	Money	8	Importe de partida 911 del mes inmediato anterior
22	AntIngExt	Money	8	Importe de ingresos extraordinarios del mes inmediato anterior

## Transferencias

Campo	Nombre	Tipo	Longitud	Descripción
1	NoTransf	Char	4	Número de la transferencia (PK)
2	FechOper	Smalldatetime	4	Fecha en que se generó la transferencia
3	MarcaE	Char	2	Estado en el que se encuentra
4	CodigoD	Char	13	Código programático de disminución
5	TipEjerD	Char	1	
6	CodigoA	Char	13	Código programático de aumento
7	TipEjerA	Char	1	
8	Importe	Money	8	Importe de la transferencia
9	NoOficio	Char	7	Número de oficio
10	FechOficio	Smalldatetime	4	Fecha del oficio
11	MarcaCrea	Char	1	Marca de creación
12	Mes	Char	2	Mes de proceso

## Ingresos: utilizada para Ingresos Extraordinarios

Campo	Nombre	Tipo	Longitud	Descripción
1	Fecha	Smalldatetime	4	Fecha de la operación (PK)
2	TipoDoc	Char	2	Tipo de documento (PK)
3	Numero	Char	5	Número asignado (PK)
4	Cuenta	Char	9	Número de cuenta contable
5	Cargos	Money	8	Importe del cargo
6	Abonos	Money	8	Importe del abono

## Pólizas

Campo	Nombre	Tipo	Longitud	Descripción
1	TipoDoc	Char	2	Tipo de documento
2	Folio	Char	6	Número de folio (PK)
3	Fecha	Smalldatetime		Fecha de creación de la póliza (PK)
4	Previo	Char	6	Número de previo
5	Cuenta	Char	13	Número de cuenta contable
6	CodGasto	Char	14	Código programático de gasto
7	CodApoyo	Char	14	Código programático de apoyo
8	Cargo	Money	8	Importe del cargo

## Pólizas (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
9	Abono	Money	8	Importe del abono
10	SubDep	Char	5	Clave de la dependencia+subdependencia
11	Comentario	Char	255	Comentario de la póliza

## Boletos

Campo	Nombre	Tipo	Longitud	Descripción
1	Numero	Char	6	Número del documento (PK)
2	Renglon	Char	3	Número de renglón del documento (PK)
3	Dependencia	Char	3	Clave de la dependencia
4	SudDep	Char	2	Clave de la subdependencia
5	Codigo	Char	14	Código programático
6	CtaContable	Char	12	Cuenta contable
7	Previo	Char	9	No. de previo que se afecta
8	Agencia	Char	2	Clave de la Agencia de Viajes
9	TotalMn	Money	8	Importe en moneda nacional
10	Fecha	Smalldate time	4	Fecha de compra del boleto
11	Periodo	Char	4	Periodo de compra

## Boletos (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
12	Emision	Smalldatetime	4	Fecha de emisión del boleto
13	Boleto	Char	12	Número del boleto
14	Pasajero	Char	20	Nombre del pasajero
15	Rfc	Char	13	RFC del pasajero
16	Puesto	Char	20	Puesto del pasajero en la Universidad
17	ClvAgencia	Char	10	Clave de la agencia de viajes
18	Itinerario	Char	9	Itinerario del vuelo
19	Servicio	Char	2	Tipo de servicio contratado

## Depósitos

Campo	Nombre	Tipo	Longitud	Descripción
1	IDDep	Int	4	Identificador del depósito (PK)
2	Mes	Tinyint	1	Mes en que se generó el depósito
3	FechOfi	Smalldatetime	4	Fecha de generación del oficio
4	NoOfi	Char	7	Número del oficio
5	Coordi	Char	12	Nombre de la coordinación
6	CveDrp	Char	5	Clave de la dependencia+subdependencia

## Depósitos (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
7	CtaDep	Char	17	Cuenta bancaria de depósito
8	Importe	Money	8	Importe total del depósito
9	FechDep	Smalldate time	2	Fecha de realización del depósito

## Folios

Campo	Nombre	Tipo	Longitud	Descripción
1	ClaveUpa	Char	3	Clave de la unidad de proceso (PK)
2	TipoDoc	Char	2	Tipo de documento (PK)
3	Numero	Char	6	Número del folio (PK)
4	TipoOper	Char	2	Tipo de operación
5	Dependencia	Char	3	Clave de la dependencia
6	SubDep	Char	2	Clave de la subdependencia
7	TotalMe	Money	8	Importe en moneda extranjera
8	TotalMn	Money	8	Importe en moneda nacional
9	Fecha	Smalldate time	4	Fecha en que entro el folio a proceso
10	Status	Char	1	Estado en que se encuentra el folio
11	Beneficiario	Char	40	Nombre del beneficiario
12	Rfc	Char	13	RFC del beneficiario

## Cheques

Campo	Nombre	Tipo	Longitud	Descripción
1	ClaveUpa	Char	3	Clave de la unidad de proceso (PK)
2	TipoDoc	Char	2	Tipo de documento (PK)
3	Numero	Char	6	Número de folio (PK)
4	Renglon	Char	3	Renglón del folio
5	Rfc	Char	13	RFC del beneficiario del cheque
6	Benefic	Char	40	Nombre del beneficiario del cheque
7	Impuesto	Money	8	Importe del impuesto retenido
8	Neto	Money	8	Importe neto del cheque
9	NumChq	Char	7	Número del cheque
10	Tipo	Char	1	Tipo del cheque
11	Fecha	Smalldate time	4	Fecha de emisión del cheque
12	AutCan	Char	5	Estado en que se encuentra el cheque

## Rechazos

Campo	Nombre	Tipo	Longitud	Descripción
1	ClaveUpa	Char	3	Clave de la unidad de proceso (PK)
2	TipoDoc	Char	2	Tipo de documento (PK)
3	Numero	Char	6	Número de folio (PK)

## Rechazos (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
4	Renglon	Char	3	Renglón del folio (PK)
5	Fecha	Smalldatetime	4	Fecha en que se generó el rechazo
6	Descripcion	Char	80	Descripción de los errores

## PapResumen

Campo	Nombre	Tipo	Longitud	Descripción
1	Clave	Char	8	Clave del proyecto
2	Codigo	Char	12	Código programático (PK)
3	Cta_Ban	Char	12	Número de la cuenta de banco
4	Imp_Proj	Money	8	Importe total del proyecto
5	Mini_Entrega	Money	8	Importe de las ministraciones entregadas
6	MinixEnt	Money	8	Importe de las ministraciones por entregar
7	MinixComp	Money	8	Importe de las ministraciones por comprobar
8	Mini_Comp	Money	8	Importe de las ministraciones comprobadas
9	Nomb_Proj	Varchar	255	Nombre del proyecto
10	Nomb_Respons	Varchar	50	Nombre del responsable del proyecto
11	RFC	Char	11	RFC del responsable del proyecto
12	CoResp	Varchar	50	Nombre del co-responsable del proyecto

### PapResumen (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
13	SecAdm	Varchar	50	Nombre del secretario administrativo
14	Tipo_Camb	Money	8	Tipo de cambio utilizado para el proyecto
15	Clave_Banc	Char	2	Clave del banco asignado
16	Importe_Financ	Money	8	Importe total de financiamiento
17	Fech_Financ	Smalldatetime		Fecha del financiamiento

### PapPresupuesto

Campo	Nombre	Tipo	Longitud	Descripción
1	Codigo	Char	12	Código programático (PK)
2	Partida	Char	3	Clave de la partida de gasto (PK)
3	Monto	Money	8	Importe total asociado al código
4	Monto_Mn	Money	8	Importe total en moneda nacional
5	Disponible	Money	8	Importe total disponible
6	Previo	Money	8	Importe total previo
7	Pagado	Money	8	Importe total pagado

## PapSuperCray

Campo	Nombre	Tipo	Longitud	Descripción
1	Codigo	Char	12	Código programático (PK)
2	Consecutivo	Char	2	Número consecutivo (PK)
3	Forma_Libera	Varchar	30	Forma de liberación
4	Monto	Money	8	Importe total
5	Tipo_Cambio	Money	8	Tipo de cambio utilizado para el proyecto
6	Monto_Mn	Money	8	Importe total en moneda nacional

## PapMovimientos

Campo	Nombre	Tipo	Longitud	Descripción
1	Codigo	Char	12	Código programático (PK)
2	Partida	Char	3	Clave de la partida de gasto (PK)
3	Poliza	Char	10	Número de póliza del movimiento
4	Fecha	Smalldatetime	4	Fecha de proceso del movimiento
5	Folio	Char	6	Número de folio asociado
6	Previo	Money	8	Número de previo asociado
7	Imp_Carg	Money	8	Importe total del cargo
8	Imp_Abono	Money	8	Importe total del abono

## PapMovimientos (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
9	Operacion	Char	3	Tipo de operación realizada

## PapHonorarios

Campo	Nombre	Tipo	Longitud	Descripción
1	Codigo	Char	12	Código programático (PK)
2	Rfc	Char	15	RFC del beneficiario
3	Nombre	Varchar	50	Nombre del beneficiario
4	Tipo_Contrato	Char	2	Tipo de contrato
5	Contrato	Char	8	Número de contrato
6	Fecha_R	Smalldatetime		Fecha de registro de la comprobación
7	F_Pago	Smalldatetime		Fecha en que recibió el pago
8	Cantrarecibo	Char	6	Número de contra-recibo
9	Imp_Pago	Money	8	Importe del pago
10	Iva_Pago	Money	8	Importe del IVA aplicado al pago
11	Isr_Ret	Money	8	Importe del ISR retenido
12	Total	Money	8	Importe neto a pagar

## PapBecarios

Campo	Nombre	Tipo	Longitud	Descripción
1	Numero	Int	4	Número de identificación (PK)
2	Código	Char	12	Código programático (PK)
3	Rfc	Char	15	RFC del becario
4	Nombre	Varchar	50	Nombre del becario
5	Tipo_Contrato	Char	2	Tipo de contrato
6	Contrato	Char	8	Número de contrato
7	F_Pago	Smalldatetime		Fecha de pago
8	Contrarecibo	Char	6	Número del contra-recibo
9	Imp_Pago	Money	8	Importe total del pago

## PapBoletos

Campo	Nombre	Tipo	Longitud	Descripción
1	Código	Char	11	Código programático (PK)
2	Año	Char	4	Año en que se realiza la compra
3	Contrarecibo	Char	6	Número del contra-recibo
4	Cve_Comp_Aerea	Char	2	Clave de la línea aérea
5	Tipo	Char	2	Tipo línea/agencia
6	Destino	Char	30	Destino del vuelo

### PapBoletos (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
7	Fecha	Smalldatetime		Fecha de salida
8	Usuario	Char	30	Nombre del pasajero
9	Total	Money	8	Importe total del boleto

### PapOtros

Campo	Nombre	Tipo	Longitud	Descripción
1	Codigo	Char	12	Código programático (PK)
2	Año	Char	4	Año en que se realiza la compra
3	Contrarecibo	Char	6	Número del contra-recibo
4	Cve_Agencia	Char	2	Clave de la agencia de viajes
5	Usuario	Varchar	50	Nombre del usuario del servicio
6	Cargo	Varchar	20	Cargo del usuario
7	Importe	Money	8	Importe total del servicio

## PapPrevios

Campo	Nombre	Tipo	Longitud	Descripción
1	Previo	Char	6	Número de previo (PK)
2	Proyecto	Char	12	Número del proyecto (PK)
3	Partida	Char	3	Clave de la partida de gasto
4	Contrarecibo	Char	6	Número de contra-recibo
5	Fecha	Smalldate time		Fecha de creación del previo
6	Imp_Mn	Money	8	Importe total de creación

## Usuarios

Campo	Nombre	Tipo	Longitud	Descripción
1	Clave	Char	8	Clave de acceso al sistema (PK)
2	Pass	Char	8	Contraseña de acceso al sistema
3	Nombre	Char	15	Nombre del usuario
4	Paterno	Char	15	Apellido paterno
5	Materno	Char	15	Apellido materno
6	Telefono	Char	12	Número telefónico
7	Status	Char	1	Bandera de estado actual
8	Mensaje	Char	1	Bandera de mensaje
9	Dep	Char	3	Clave de dependencia a la que pertenece

## Usuarios (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
10	SubDep	Char	2	Clave de subdependencia a la que pertenece
11	Entrada	Smalldatetime	4	Fecha y hora de última entrada al sistema
12	Salida	Smalldatetime	4	Fecha y hora de última salida del sistema
13	NoVeces	Smallint	2	Número de accesos al sistema
14	Tiempo	Smallint	2	Tiempo de acceso al sistema
15	CveDep	Char	3	Clave de la dependencia que tiene alcance para consultar
16	CveSubDep	Char	2	Clave de la subdependencia que tiene alcance para consultar
17	DirIP	Char	15	Dirección IP de la PC de consulta
18	Tipo	Char	1	Tipo de usuario en el sistema

## Mensajes

Campo	Nombre	Tipo	Longitud	Descripción
1	Numero	Smallint	2	Número de mensaje (PK)
2	Fecha	Smalldatetime	4	Fecha de emisión
3	Mensaje	Char	255	Descripción del mensaje

## MensajeUsuario

Campo	Nombre	Tipo	Longitud	Descripción
1	Numero	Int	4	Número de mensaje (PK)
2	Clave	Char	8	Clave del usuario (PK)

## Circulares

Campo	Nombre	Tipo	Longitud	Descripción
1	IdDocu	Int	4	Identificador de documento (PK)
2	TipoDoc	Char	1	Tipo de documento
3	Numero	Char	20	Número del documento
4	Asunto	Char	120	Resumen del documento
5	DepEmi	Char	5	Clave de dependencia emisora
4	FechEmi	Smalldate time	4	Fecha de emisión
5	Dirigido	Varchar	255	A quien va dirigido el documento
6	Texto	Text	16	Texto del documento
7	Pclave	Char	255	Palabras clave

## Imágenes

Campo	Longitud	Tipo	Longitud	Descripción
1	IdDocu	Int	4	Identificador de documento (PK)
2	Pagina	Int	4	Número de página
3	Imagen	Image	16	Archivo de la imagen

## Funciones

Campo	Nombre	Tipo	Longitud	Descripción
1	CveFuncion	Char	2	Clave de la función (PK)
2	Titulo	Char	50	Nombre de la función
3	Texto	Text	16	Texto asociado
4	CveDep	Char	5	Clave de dependencia+subdependencia

## Estructura

Campo	Nombre	Tipo	Longitud	Descripción
1	IdImagen	Char	2	Clave de la imagen (PK)
2	Titulo	Char	50	Nombre de la función
3	Imagen	Image	16	Imagen asociada
4	CveDep	Char	5	Clave de dependencia+subdependencia

## Directorio

Campo	Nombre	Tipo	Longitud	Descripción
1	Id	Tinyint	1	Identificador de persona (PK)
2	Dependencia	Char	50	Nombre de la dependencia
3	Paterno	Char	15	Apellido paterno
4	Materno	Char	15	Apellido materno
5	Nombre	Char	20	Nombre de la persona
6	Titulo	Char	12	Titulo de la persona
7	Cargo	Char	85	Puesto dentro de la Universidad
8	Ubicacion	Char	40	Ubicación física de la oficina
9	Telefono	Char	35	Número telefónico
10	Fax	Char	10	Número de fax
11	Correo	Char	36	Clave de correo electrónico
12	CveDep	Char	5	Clave de dependencia+subdependencia (PK)

## Aplicaciones

Campo	Nombre	Tipo	Longitud	Descripción
1	IdAplic	Int	4	Identificador de la aplicación (PK)
2	NomAplic	Char	255	Nombre de la aplicación

### Aplicaciones (continuación)

Campo	Nombre	Tipo	Longitud	Descripción
3	Descrip	Text	16	Texto asociado
5	NomArch	Char	25	Nombre del archivo
6	ContAplic	Image	16	Contenido de la aplicación

El diccionario de datos es de gran utilidad si cada entrada se registra de manera consistente, incluyendo el nombre del dato, el sinónimo, su descripción, el rango, la longitud y cualquier otra información necesaria para su edición.

## 4.2 Diseño y Desarrollo

En esta etapa del ciclo de desarrollo de los sistemas, el analista usa la información que recolectó con anterioridad y elabora el diseño lógico de la aplicación. El analista diseña procedimientos precisos de captura de datos, con el fin de que los datos que se introduzcan al sistema sean los correctos. También genera accesos efectivos al sistema de información, mediante el uso de las técnicas de diseño de formas y de pantallas.

Una parte del diseño lógico del sistema de información es el diseño de la interfaz con el usuario. La interfaz conecta al usuario con el sistema, y evidentemente, es de suma importancia. Algunos ejemplos de interfaces para el usuario son: el uso del teclado para introducir preguntas o respuestas; la aplicación de menús en la pantalla, con las opciones que brinda; el empleo de dispositivos como el ratón (mouse), y muchos otros.

La etapa del diseño también incluye la estructuración de los archivos o la base de datos que almacenará aquella información requerida por el personal que toma las decisiones en la organización. Una base de datos bien organizada es fundamental para cualquier sistema de información. De manera similar, en esta etapa el analista diseña la salida (en pantalla o impresa) hacia el usuario, de acuerdo con las necesidades de información.

La fase de desarrollo se centra en el *cómo*. Es decir, durante el desarrollo un ingeniero intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función de acuerdo a la arquitectura del software, cómo han de atenderse los detalles procedimentales, cómo han de caracterizarse las interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de

realizarse la prueba. Los métodos aplicados durante la fase de desarrollo pueden ser variados, sin embargo, las tres técnicas específicas que permanecen como una constante son: el diseño del software, la generación del código y la prueba del software.

#### 4.2.1 Diseño Descendente

Existen dos enfoques tradicionales para el diseño de sistemas: el diseño ascendente (bottom-up) y el diseño descendente (top-down). Para el caso particular del SIAU, se decidió utilizar el diseño descendente por ser el que mejor se adaptaba a los recursos disponibles y necesidades observadas.

El enfoque descendente implica observar la gran imagen del sistema y luego, explosionarlo o desglosarlo en partes más pequeñas o subsistemas, tal y como se muestra en la figura 4.14.

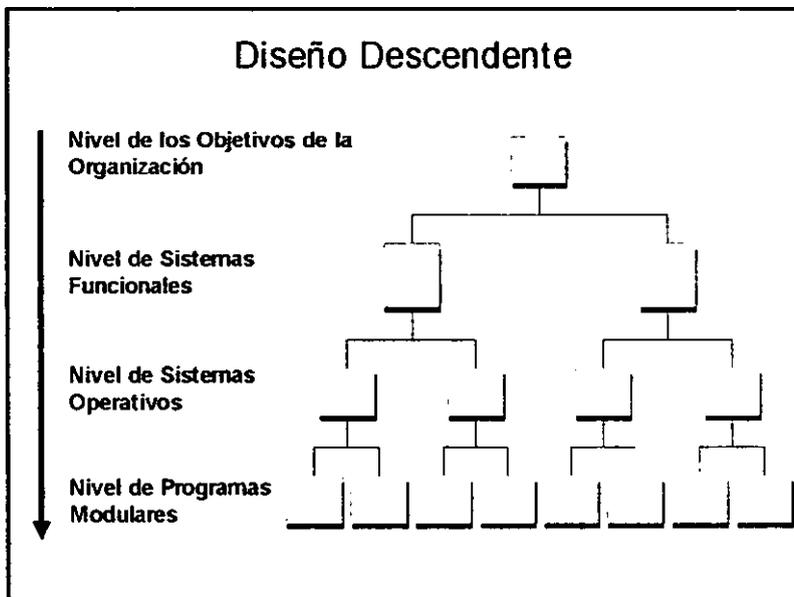


Figura 4.14 Enfoque de Diseño Descendente (top-down)

El diseño descendente obliga a que los analistas de sistemas se enteren primero de los objetivos globales de organización, así como el establecimiento de la mejor manera de satisfacerlos dentro de un sistema integral. Luego, el analista se dirigirá a dividir tal sistema en sus subsistemas y sus requerimientos.

Cuando el analista de sistemas emplea un enfoque descendente, está empleando las interrelaciones e interdependencias de los subsistemas, para apearse lo mejor posible a las necesidades existentes en la organización.

El enfoque descendente otorga importancia a la sinergia o las interfaces requeridas por el sistema y los subsistemas; los cuales no existen en el enfoque ascendente.

Dentro de las ventajas de la utilización de un enfoque descendente en el diseño de los sistemas, se encuentra el evitar el caos originado al tratar de diseñar el sistema "en un sólo paso". Como es bien conocido, la planeación y la implantación de sistemas de información es increíblemente compleja; por lo que tratar de integrar a todos los subsistemas haciendo que funcionen al unísono, es buscar el fracaso.

La segunda ventaja de hacer el uso del enfoque descendente en el diseño, es la posibilidad de contar con grupos de analistas de sistemas trabajando por separado pero simultáneamente en subsistemas independientes, pero necesarios. Esto puede ahorrar una gran cantidad de tiempo. El trabajo de grupos integrados para el diseño de subsistemas es particularmente conveniente cuando se busca asegurar la calidad total.

La tercera ventaja estriba en evitar el gran problema asociado con un enfoque ascendente. Esto es, la utilización de un enfoque descendente, previene que el analista de sistemas se adentre en los detalles y dé la pauta para que se pierdan los

objetivos centrales del sistema.

Existen ciertos inconvenientes en el diseño descendente que el analista de sistemas debe mantener en mente. El primero es que existe el riesgo de que el sistema se divida en subsistemas "incorrectos". Se debe prestar atención a la necesidad de la superposición y la distribución de los recursos, de tal forma que una participación de los subsistemas tenga sentido en el esquema global del sistema. Además, es importante que cada subsistema se integre de manera correcta al sistema.

El segundo inconveniente es que una vez que se realizan las divisiones en subsistemas, sus interfaces pueden descuidarse o simplemente ignorarse. La responsabilidad para lograr la adecuada interrelación debe quedar bien detallada.

Una tercera precaución que debe acompañar al uso del diseño descendente, es que los subsistemas deberán reintegrarse eventualmente. Los mecanismos para la integración deben plantearse desde un principio. Una sugerencia es intercambiar, de manera regular, la información entre los equipos de subsistemas, y otra, consiste en la utilización de instrumentos que permitan la flexibilidad adecuada en el caso de que se requirieran cambios de los sistemas interrelacionados.

El enfoque descendente proporciona al grupo de sistemas, una clara división de los usuarios en fuerzas de tareas para los subsistemas. Los grupos asignados a las tareas, establecidos de esta manera, pueden tener una doble función, similar a los círculos de control de calidad de los sistemas informáticos para la administración. De esta manera se tiene una estructura necesaria para el aseguramiento de la calidad, así como una motivación apropiada para incorporar el subsistema al cumplimiento de los objetivos de los departamentos involucrados.

#### **4.2.2 Desarrollo modular**

Aunado a la decisión de desarrollar un diseño con enfoque descendente, también se consideró pertinente adoptar para la programación la concepción modular.

El enfoque de programación modular significa descomponer la programación en fracciones lógicas y manejables. Este tipo de programación se apega bien al diseño descendente porque enfatiza las interfaces entre los módulos, en lugar de mantenerlas ignoradas hasta el final del desarrollo del sistema. De manera ideal, cada módulo debe ser funcionalmente cohesivo, de tal manera que satisfaga sólo una función.

El diseño de programas modulares tiene tres ventajas básicas.

Los módulos son más fáciles de escribir y de revisar, ya que están virtualmente autocontenidos. La detección de un error dentro de un módulo es menos complicada, ya que los problemas asociados a un módulo no llegarán a trascender a los otros.

El mantenimiento de los módulos es más fácil. Las modificaciones pueden limitarse a unos cuantos módulos y no al programa completo.

La problemática de los módulos es más fácil de entender, ya que son sistemas autocontenidos. Eso significa que un lector entenderá la función de un módulo específico con sólo tomar su listado de código.

Algunos lineamientos para la programación incluyen:

- Mantener cada módulo de un tamaño manejable (de manera ideal incluyendo sólo una función).

- Prestar atención particular a las interfaces críticas (esto es, observar detenidamente los datos y las variables de control que pasan entre los módulos).
- Minimizar el número de módulos que el usuario necesite modificar cuando haga cambios.
- Mantener las relaciones jerárquicas establecidas en las etapas de descenso.

#### **4.2.3 Diseño de la Base de Datos**

Algunas personas consideran el almacenamiento de los datos como la esencia del sistema de información. Los objetivos generales del diseño de la organización del almacenamiento de los datos son que se logre:

- Disponibilidad de los datos
- Integración de los datos
- Almacenamiento eficiente de los datos
- Actualización eficiente
- Recuperación dirigida de la información

Para que el almacenamiento de los datos sea eficiente, debe vigilar el cumplimiento de algunos requisitos. Primero, los datos deben estar disponibles para cuando el usuario desee usarlos. Segundo, los datos deben ser precisos y consistentes (la base de datos debe tener una integridad). Tercero, el almacenamiento, actualización y grabado de los datos debe ser ágil, sencillo y confiable. Finalmente, es necesario que el acceso a la información tenga un propósito. La información obtenida de los datos almacenados debe contar con un formato útil que facilite la administración,

la planeación, el control y/o la toma de decisiones.

En un sistema de información basado en computadora se cuenta con dos enfoques para el almacenamiento de los datos. El primero de ellos consiste en almacenar los datos en archivos individuales, exclusivos para una aplicación en particular. El segundo, involucra la elaboración de una base de datos. Una base de datos es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones.

Las bases de datos no son meramente una colección de archivos. Más bien, una base de datos es una fuente central de datos significativos, los cuales son compartidos por numerosos usuarios para diversas aplicaciones. La esencia de una base de datos es el Sistema Administrador de la Base de Datos (DBMS: Database Management System), el cual permite la creación, modificación y actualización de la base de datos; la recuperación de los datos y la emisión de reportes. A la persona responsable de asegurar que la base de datos satisfaga los objetivos programados se le denomina administrador de la base de datos.

Los objetivos de eficacia de la base de datos son:

- Garantizar que los datos puedan ser compartidos por los usuarios, para una variedad de aplicaciones.
- Asegurar que el mantenimiento de los datos sea preciso y consistente.
- Certificar que todos los datos requeridos para las aplicaciones presentes y futuras se encuentren siempre disponibles.
- Permitir que la base de datos evolucione y se adapte a las necesidades crecientes

de los usuarios.

- Consentir que los usuarios desarrollen su propia visión de los datos, sin preocuparse por la manera en que los datos se encuentren almacenados físicamente.

Esta lista de requisitos advierte de las ventajas y desventajas del enfoque de la base de datos. Primero, el compartir los datos significa que éstos deben almacenarse una sola vez. Esto a su vez apoya la idea de que se mantenga la integridad de los datos, ya que el cambio de los datos se realizará de manera más sencilla y confiable si éstos aparecen una vez y no en varios archivos.

Cuando un usuario necesite un dato particular, una base de datos con un buen diseño se debería anticipar a tal necesidad (y quizás ya hayan sido utilizados en otra aplicación). En consecuencia, los datos tendrán mayor probabilidad de encontrarse disponibles en una base de datos más que en un sistema de archivos convencionales. Una base de datos con un buen diseño también llega a ser más flexible que dos archivos separados, esto significa que una base de datos llega a evolucionar conforme se modifican las necesidades de los usuarios y de sus aplicaciones.

Finalmente, el enfoque de base de datos, tiene la ventaja de permitir que los usuarios expongan sus puntos de vista sobre los datos, sin necesidad de preocuparse de la estructura presente de la base de datos o de su ubicación física.

No obstante los beneficios del enfoque de base de datos, también deben considerarse las desventajas que presenta. Entre estas últimas, las que resaltan son las siguientes: los datos deben almacenarse en un sólo lugar, lo que trae como consecuencia que los datos son más vulnerables a accidentes y, por tanto,

requerirán de un proceso de respaldo completo y constante. Además, también existe el riesgo de que la persona que administra la base de datos pueda convertirse en el único privilegiado o habilitado para estar cerca de los datos. Una más, es que los procedimientos burocráticos requeridos para modificar o para actualizar la base de datos pueden llegar a ser insuperables.

Otra desventaja para la administración de los datos como recurso, se presenta al intentar satisfacer dos objetivos de eficiencia:

- Reducción del tiempo requerido para insertar, actualizar, eliminar y recuperar los datos con demora tolerable.
- Mantenimiento del costo del almacenamiento de datos en un nivel razonable.

Es necesario recordar que una base de datos no puede optimizar la recuperación de los datos para una aplicación especial, ya que deberá compartirse con numerosos usuarios y con varias aplicaciones.

El enfoque de la base de datos es un concepto que se vuelve cada vez más relevante. El uso de base de datos relacionales mediante microcomputadoras indica el grado de difusión que este concepto ha alcanzado entre los usuarios. Con este enfoque, los usuarios toman una parte de la base de datos central y cargan sus computadoras personales. Luego estas pequeñas bases de datos se utilizan para emitir reportes o contestar consultas específicas del usuario final.

#### **4.2.3.1 Organización de la Base de Datos**

Una base de datos, a diferencia de un archivo, la comparten muchos usuarios, y naturalmente, cada usuario verá los datos de manera diferente. Esto se refiere a la forma en que un usuario concibe y describe los datos desde una presentación

personal. Sin embargo, el problema es que los diferentes usuarios tienen diversos enfoques. Estas presentaciones se examinan en el modelo lógico global de la base de datos, que eventualmente deberá desarrollarse. Finalmente, el modelo lógico de la base de datos debe transformarse en su correspondiente diseño físico. El diseño físico considera la forma del almacenamiento de los datos y de sus interrelaciones, así como la mecánica del acceso.

Existen tres tipos básicos de base de datos con una estructuración lógica: el tipo jerárquico, el de red y el de relación. Para SIAU, se aplicó el enfoque relacional, ya que además de ser el más conocido y utilizado, es el que mejor se adapta al software de bases de datos (SQL Server) que se utilizó para el desarrollo del sistema.

#### **4.2.3.2 Bases de Datos Relacionales**

Una base de datos relacional consiste en una o más tablas bidimensionales, las cuales se refieren como relaciones. Los renglones de las tablas representan los registros y las columnas contienen los atributos.

El mantenimiento de las tablas de una base de datos relacional es bastante simple al compararlo con el mantenimiento de una estructura jerárquica o de red. Una de las ventajas principales de la estructura relacional, es que las consultas específicas se manejan de una manera muy eficiente.

Con el fin de que las estructuras relacionales sean útiles y manejables, las tablas relacionales primero deben “normalizarse”. Existen varios niveles de normalización que deben de aplicarse sobre cada una de las tablas de la base de datos. En la siguiente sección se detalla tal proceso de normalización.

### 4.2.3.3 Normalización

La normalización es el proceso de transformación de las complejas presentaciones de usuarios, y de los almacenamientos de datos, en conjuntos estables de estructuras de datos de menor tamaño. Además de ser más sencillas, tales estructuras son más estables. Las estructuras de datos normalizados son más fáciles de mantener.

Al comenzar, ya sea con la presentación del usuario o con el almacenamiento de información diseñado para un diccionario de datos, el analista normaliza una estructura de datos en tres pasos. Cada paso involucra un importante procedimiento de simplificación de dicha estructura. La relación derivada de la presentación del usuario o del almacenamiento de datos, generalmente no se encuentra normalizada, por lo que casi todas las estructuras de datos deberán someterse a este proceso.

La primera etapa del proceso de normalización incluye la eliminación de grupos repetidos y la identificación de la llave que define al criterio primario. Con el fin de hacer esto, la relación necesita desglosarse en dos o más relaciones. En este punto, las relaciones pueden encontrarse en la forma normal tercera, pero quizás sean necesarios más pasos para transformar las relaciones a la forma normal tercera.

El segundo paso asegura que todos los atributos no-llave, sean completamente dependientes de la llave del criterio primario. Todas las dependencias normales se eliminan y se colocarán en otra relación.

El tercer paso elimina cualquier dependencia transitoria. Una dependencia transitoria es aquella en la cual sus atributos no-llave son dependientes de otros atributos no-llave.

Es importante señalar, que para el SIAU, no en todos los casos se cumplió con las tres formas de normalización, ya que al tratarse de tablas muy grandes (mayores a cien mil registros) y considerando que un alto número de usuarios podrían estar accedendo la misma tabla en forma simultánea, se decidió que algunas de las tablas fueran divididas en doce partes, correspondiendo cada una de las partes al mes en que se generó la información. Esto principalmente se realizó con el fin de mejorar la velocidad de respuesta en las consultas realizadas por los usuarios.

De tal manera que la relación de tablas con las que cuenta el sistema, es de la siguiente forma:

Nombre	Número de tablas
Previos01..12	12
GastosReserva01..12	12
CodigosMen01..12	12
Transferencias	1
Ingresos	1
Pólizas	12
Boletos	12
Depósitos	1
Folios	1
Cheques	1
Rechazos	1
PapResumen	12
PapPresupuesto	12

(continuación)

<b>Nombre</b>	<b>Número de tablas</b>
PapSuperCray	12
PapMovimientos	12
PapHonorarios	12
PapBecarios	12
PapBoletos	12
PapOtros	12
PapPrevios	12
Usuarios	1
Mensajes	1
MensajeUsuario	1
Circulares	1
Imágenes	1
Funciones	1
Estructura	1
Directorio	1
Aplicaciones	1

De esta manera se cuenta con 183 tablas de usuario. Adicionalmente, para la creación de la base de datos se consideraron las tablas del sistema, de control y de seguridad, las que sumadas a las de usuario hacen un total de más de 200 tablas.

Con esta información, se puede realizar un estimado del tamaño de la base de datos, por lo que se puede iniciar el proceso de construcción.

#### 4.2.4 Construcción de la Base de Datos

El siguiente paso dentro del proceso de desarrollo del sistema fue construir la base de datos (BD) tomando la información del diccionario de datos, considerando también la información detallada en la sección previa. Para ello, se utilizó una herramienta de SQL Server llamada SQL Enterprise Manager la cual, a través de una interfaz gráfica, sirvió para realizar toda la administración de las bases de datos que contiene el servidor.

Para poder llevar a cabo dicha tarea, fue necesario realizar los siguientes pasos:

No.	Tarea a Realizar
1	Crear los dispositivos (devices) de almacenamiento y respaldo
2	Generar la Base de Datos
3	Establecer las Tablas
4.	Llenar las tablas con información
5	Crear los Indices
6	Definir las Reglas y Defaults
7	Establecer los Procedimientos Almacenados (Stored Procedures)
8	Crear los Disparadores (triggers)
9	Identificar a los Usuarios y definir los Grupos
10	Asignar Permisos a los Objetos

## 1.- Crear los dispositivos (devices) de almacenamiento y respaldo

Los devices son archivos donde va a ser almacenada la base de datos. Existen tres tipos diferentes: el de datos, el de log y el de respaldo. En los dos primeros residen los datos de la BD, mientras que el tercero se utiliza para crear los respaldos.

Con base en el número de tablas que contiene la BD y en la cantidad de información que maneja, para el sistema en cuestión se creó un *device* de datos de 475 Mb y un *device* de log de 25 Mb. El device de log podrá variar de acuerdo con la cantidad de operaciones que se hayan efectuado dentro de la base.

Para realizar esta operación, únicamente se tiene que hacer uso de la herramienta de administración y escribir los datos de acuerdo con las necesidades, tal como lo muestra la Figura 4.15.

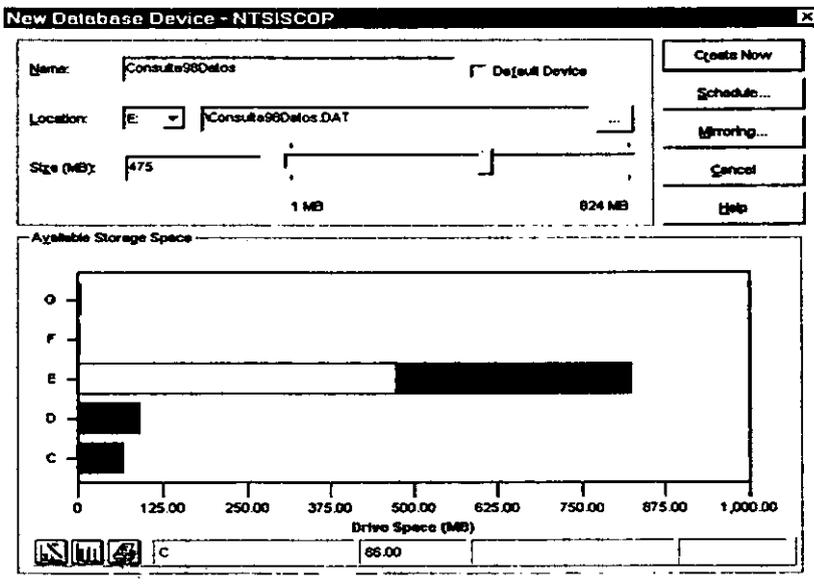


Figura 4.15 Creación de los Devices de Almacenamiento

## 2.- Generar la Base de Datos

Una vez que los devices han sido establecidos, se procede a la creación de la base de datos, donde únicamente se le tiene que indicar qué device se utilizará para guardar los datos y cuál para el log. (Figura 4.16)

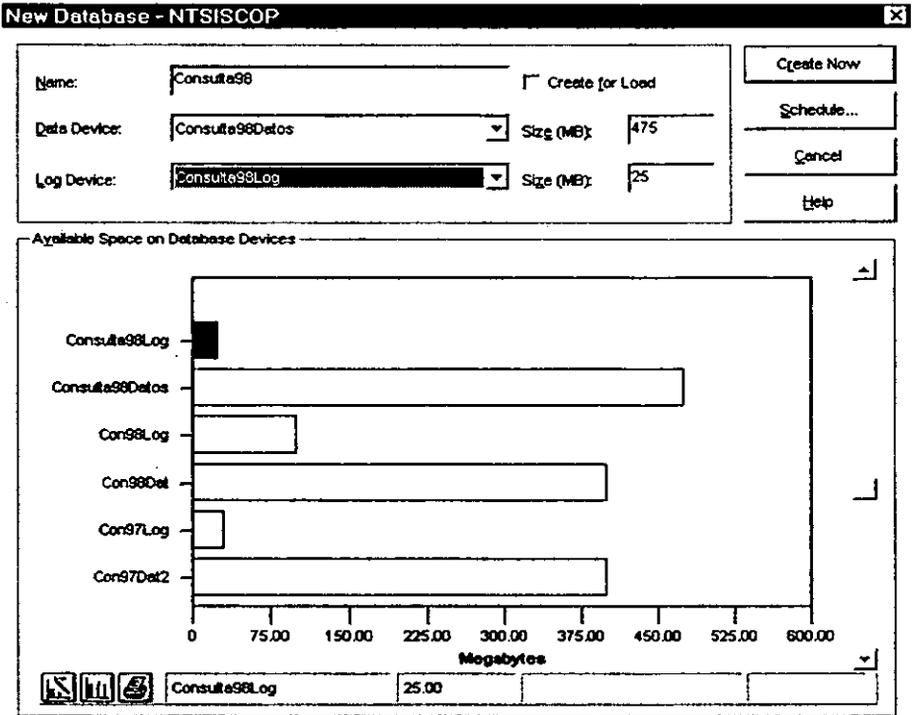


Figura 4.16 Creación de la Base de Datos

La base de datos fue creada sobre el disco E:\ con un tamaño de 500 MB.

### 3.- Establecer las Tablas

Una vez generada la base de datos se tiene que crear, una a una, cada tabla que utiliza el sistema. Este proceso puede llevar mucho tiempo, pero hay que considerar que sólo se realiza en una ocasión.

A manera de ejemplo, a continuación se desglosa el proceso de creación de la Tabla Usuarios (Figura 4.17), resaltando que el mismo procedimiento se llevó a cabo para la creación del resto de las tablas.

Key	Identity	Column Name	Datatype	Size	Nulls	Default
		Clave	char	8		
		Pass	char	8		
		Nombre	char	15		
		Paterno	char	15		
		Materno	char	15	✓	
		Telefono	char	12	✓	
		Status	char	1		
		Mensaje	char	1		
		Dep	char	3		
		SubDep	char	2		

Figura 4.17 Creación de las Tablas

La tabla contiene 22 columnas, cada registro es de 247 bytes.

#### 4.- Llenar las tablas con información

Para el llenado de las tablas existen dos procesos: el normal, basado en el comando Insert, o bien, a través de una herramienta conocida como Bulk Copy Program (BCP), la cual sirve para insertar de manera muy rápida archivos de texto, o archivos delimitados (tipo dbase o excel). El programa se invoca desde la línea de comandos del sistema operativo. Continuando con el ejemplo de la tabla de usuarios, y asumiendo que existe un archivo llamado Usuarios.txt con información idéntica a las columnas de dicha tabla, la instrucción queda como sigue:

```
C:\>Bcp Consulta..Usuarios in Usuarios.txt /c /t @ /r\n /S Ntsiscop /U sa /P pass
```

El parámetro /c indica que se trata de un archivo de tipo texto, /t nos sirve para establecer a @ como separador de columnas, /r \n indica que el delimitador de renglones (/r) es un *return* (\n); por lo último /S /U /P nos sirven para especificar el nombre del servidor, la clave del usuario y su *password* respectivamente.

#### 5.- Crear los Índices

Con base en la información del diccionario de datos, se pueden crear los índices sobre cada una de las tablas. Existen dentro de SQL Server dos tipos de índices: los *clustered* y los *no-clustered*, el primero de ellos se utiliza cuando se requiere que las búsquedas sean más rápidas; el segundo, se usa cuando las actualizaciones y las inserciones deban ser las de mayor velocidad. Para el caso del SIAU, se utilizaron índices del tipo *clustered*, ya que el 95% de las tablas del sistema son utilizadas para realizar consultas. Como ejemplo, a continuación se detalla la creación del índice de la tabla de Previos01 (Figura 4.18), cuyo procedimiento es prácticamente igual al resto.

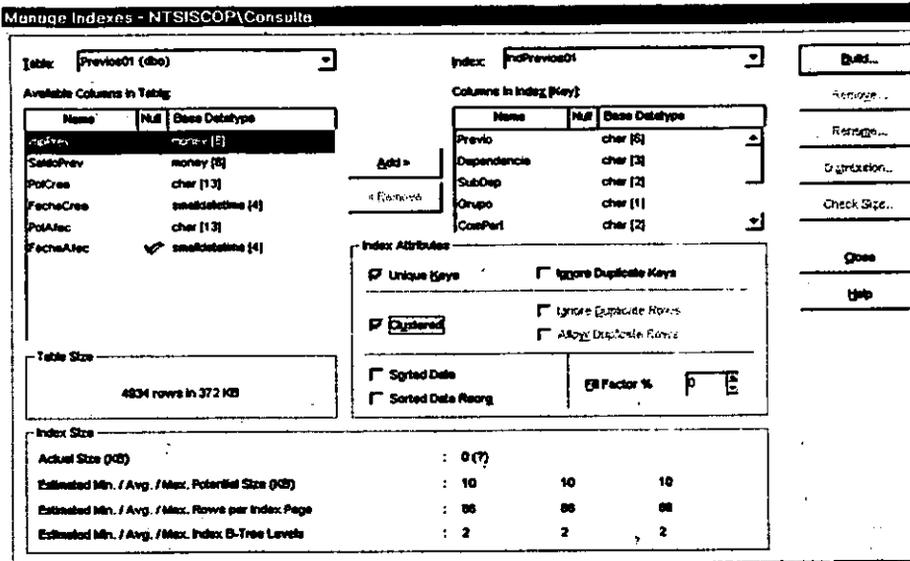


Figura 4.18 Creación de los Índices

## 6.- Definir las Reglas y Defaults

Las reglas son objetos que sirven para delimitar valores dentro de una columna. Por ejemplo, si se define una regla llamada edad entre los números 15 y 25, la columna que tenga asociada dicha regla, sólo permitirá valores que se encuentren en ese rango.

Los *defaults* son objetos que sirven para insertar datos dentro de una columna aunque no se haya especificado el dato durante una operación de Insert.

En el caso del SIAU, no fue necesaria la creación de *reglas y defaults* debido a que se trata de un sistema de consulta donde las actualizaciones son mínimas y las inserciones son nulas.

## **7.- Crear los Procedimientos Almacenados (Stored Procedures)**

Los procedimientos almacenados son consultas precompiladas que ayudan a incrementar el rendimiento y la consistencia de la base de datos cuando se realizan tareas repetitivas porque sólo son compilados la primera vez que se ejecutan. Cuando se vuelven a invocar, el tiempo de ejecución es mucho más corto comparado con el que tardaría un conjunto de sentencias SQL. Los Stored Procedures son objetos de SQL almacenados en un campo de una base de datos. Estos objetos son mantenidos por la misma base de datos y aceptan argumentos.

Los stored procedures pueden ser como un módulo de un programa definido en la base de datos y ejecutado por el programa del mismo. Es una forma simple de llamar a un procedimiento remoto, donde el cliente remoto envía un mensaje al servidor, el cual ejecuta el procedimiento.

Los procedimientos almacenados son escritos en lenguaje Transact-SQL que es una extensión del lenguaje SQL con algunas instrucciones adicionales. Cuando se crea un stored procedure su fuente, el código compilado y el ejecutable son almacenados en un sistema de catálogo de tablas.

### **Características de los Stored Procedures**

Son usados en el diseño modular para encapsular tareas complejas utilizadas por SQL, en un programa ejecutable o como parte de un campo calculado. Suponiendo la existencia de un módulo de uso no muy frecuente, si este módulo se incluye en el programa ejecutable el código se extendería considerablemente, sin embargo, si ese módulo se elabora en un Stored Procedures y se guarda en una tabla, se utiliza menos código y se puede llamar al módulo sólo cuando sea necesario.

Son usados para cambiar el flujo de trabajo de un cliente a un servidor. Por ejemplo, si se necesitan programar múltiples queries (búsquedas) en el cliente que está accediendo constantemente los registros de la base de datos del servidor, el proceso es muy tardado y, además, se incrementa considerablemente el tráfico en la red. En este caso, es mejor crear un Stored Procedures que realice todos los queries en el servidor y sólo mande el resultado al cliente, con lo que se ahorra tiempo y se disminuye el tráfico en la red.

Se pueden otorgar privilegios que los usuarios no tienen sobre una base de datos. Esta ventaja es muy importante, porque a pesar de que un usuario tenga todos los permisos para acceder una tabla y quiera realizar operaciones vía sistema para alterarla, también debe cumplir con las condiciones programadas en un Stored Procedures, de otra manera no tendría acceso a todas las operaciones. Por ejemplo: suponiendo que hay un usuario que tiene el permiso de ingresar personal a una nómina, pero este personal no cumple con las reglas establecidas en el Stored Procedure (como el que sea una persona menor de 18 años), aunque el usuario tenga los permisos necesarios, el Stored Procedures que valida la edad, no permitirá agregarlo a la tabla. En pocas palabras, los Stored Procedures facilitan la lógica del programa para el desarrollador del sistema.

### **Ventajas de los Stored Procedures en un ambiente Cliente/Servidor**

- Proveen un medio para el particionamiento de la aplicación lógica entre el servidor y el cliente.
- Mejoran la ejecución de la aplicación para reducir el tráfico en la red.
- Permiten encapsulación y generación de reglas de transacción en el servidor.

- Proveen una aplicación de mantenimiento. Es mejor tener guardadas las reglas de transacción en un Stored Procedures (en el servidor) que tenerlas que guardar en cada cliente para agilizar posteriores cambios en las reglas de transacción.
- Favorecen un control más robusto de la transacción. Esto permite que los clientes no alteren las transacciones y no choque posteriormente la información o que no alteren la información de las tablas inadecuadamente.
- Mejoran las aplicaciones de seguridad. En ambientes ODBC cualquier usuario puede acceder a la base de datos usando sus herramientas. Los Stored Procedures pueden eliminar la necesidad de permitir acceso directo a todas las tablas.

### **Desventajas de los Stored Procedures**

- El código de los procedimientos almacenados no siempre es portable a otras plataformas de DBMS.
- Es más complicada la depuración de un programa.
- La extradependencia está en las aplicaciones. Es necesario que los programas garanticen la ejecución de versiones correctas de los procedimientos.
- El uso excesivo de Stored Procedures puede degradar el rendimiento.

Dentro del SIAU los stored procedures representan uno de los elementos más importantes debido a que constituyen la parte fundamental de la arquitectura cliente/servidor, esto es, al crear todas las consultas en el servidor de base de datos se aprovechan al máximo las características de alto rendimiento de la computadora central, ya que este equipo será el encargado de ejecutar las consultas y generar los

resultados, a fin de que en la parte del cliente únicamente se presente en pantalla o vía reportes la información obtenida del proceso.

Para generar una consulta (query) se utiliza el lenguaje de programación Transact-SQL, a través del cual se puede realizar desde una consulta muy simple, hasta una que involucre varias tablas y requiera de muchos parámetros. Por tanto, la creación de procedimientos puede estar dentro de tres niveles de acuerdo con su complejidad:

El primer nivel se refiere a consultas muy simples donde no es necesario el uso de parámetros. Por ejemplo, se puede crear un procedure, el cual a través de una consulta a la tabla de Previos, presente algunas de las columnas de la tabla. Este ejemplo se desglosa a continuación:

```
Create Procedure sp_Consulta_Previos
As
Select Previo,
  Codigo=Programa+'.'+Subprog+'.'+Dependencia+'.'
      +SubDep+'.'+Grupo+ComPart,
  Folio=substring(polcrea,9,5) ,
  Fecha=FechaCrea,
  Importe=ImpPrev,
  Saldo=SaldoPrev,
  Ejercido=ImpPrev-SaldoPrev,
  FolioUlt=substring(PolAfec,9,5) , FechaUlt=FechaAfec
From Previos01
Go
```

Al invocar al procedimiento *sp\_Consulta\_Previos*, se presentarán todos los registros de la tabla *Previos01*. Estos procedimientos se utilizan para realizar consultas muy generales dentro de la aplicación.

El segundo nivel se refiere a consultas más elaboradas, es decir, a aquellas que requieren uno o más parámetros para ejecutarse. Por ejemplo, si se desea limitar el

número de dependencia y subdependencia a la cual el usuario pertenece de acuerdo al tipo de clave que tiene, el procedimiento quedaría de la siguiente forma:

```
Create Procedure sp_Consulta_Previos_Dep
/* Parametros utilizados */
@Tipo char(6), @Dep char(3), @Subdep char(2)
As
If substring(@Tipo,2,3) = "000" /* Nivel de Acceso */
  Select Previo,
    Codigo=Programa+'.'+Subprog+'.'+Dependencia+'.'
      +SubDep+'.'+Grupo+ComPart,
    Folio=substring(polcrea,9,5),
    Fecha=FechaCrea,
    Importe=ImpPrev,
    Saldo=SaldoPrev,
    Ejercido=ImpPrev-SaldoPrev,
    FolioUlt=substring(PolAfec,9,5), FechaUlt=FechaAfec
  From Previos01
  Where dependencia=@Dep and SubDep=@Subdep
/*End If */
Go
```

Para ejecutar el procedimiento anterior, se tiene que invocar de la siguiente manera:

```
sp_Consulta_Previos 'D00000','721','03'
```

En este caso ya se involucran sentencias de programación dentro del procedimiento, convirtiéndose en una consulta más elaborada, pero al mismo tiempo más versátil.

Finalmente, el tercer nivel se refiere a consultas en donde se involucran tantos parámetros como sea posible, incluso donde el nombre de las tablas también es considerado como parámetro. Para llevar a cabo esto, se tienen que formar cadenas de strings y ejecutarlas utilizando el comando *Execute*, tal como se muestra en el ejemplo siguiente:

```

/* Descripción: Obtiene información de Honorarios Papiit */
/* Autor: Arturo Jallath C. */
/* Modificado: Humberto Blas */
/* Modulo de VB: FrmPap1 */
/* Tablas Usadas: PapHonora01 a PapHonora12 */

```

```

Create Procedure sp_PapHonora

```

```

/* Parametros utilizados */

```

```

    @Dep char(3), @SDep char(2), @Mes char(2),
    @Tipo char(1), @Cod Char(11)=null, @TC char(2)=null,
    @Clave char(8)=null

```

```

As

```

```

If @Tipo="G" /*Consulta de Tipo General*/
If @Dep="000" /*Nivel Institucional*/
EXECUTE ('select * from PapHonora'+@Mes)
Else
If @SDep="00" /*Nivel Dependencia*/
EXECUTE ('select * from PapHonora'+@Mes+
' Where substring(Codigo,3,3)="'+@Dep+'')
Else /*Nivel SubDependencia*/
EXECUTE ('select * from PapHonora'+@Mes+
' Where substring(Codigo,3,3)="'+@Dep+
' and substring(Codigo,6,2)="'+@SDep+'')
/*Edif*/
/*Endif*/
/*Endif*/

```

```

If @Tipo="C" /*Consulta por Etapa-Código*/
If @Dep="000" /*Nivel Institucional*/
EXECUTE ('select * from PapHonora'+@Mes+
' Where Codigo="'+@Cod+
' and Tipo_Contrato="'+@TC+'')
Else
If @SDep="00" /*Nivel Dependencia*/
EXECUTE ('select * from PapHonora'+@Mes+
' Where substring(Codigo,3,3)="'+@Dep+
' and Codigo="'+@Cod+
' and Tipo_Contrato="'+@TC+'')
Else /*Nivel SubDependencia*/
EXECUTE ('select * from PapHonora'+@Mes+
' Where substring(Codigo,3,3)="'+@Dep+
' and substring(Codigo,6,2)="'+@SDep+
' and Codigo="'+@Cod+
' and Tipo_Contrato="'+@TC+'')
/*Edif*/
/*Endif*/

```

continuación:

```
/*Endif*/
If @Tipo="S" or @Tipo="D" /*Consulta Por Dep o Subdep*/
  If @Sdep="00" /*Nivel Dependencia*/
    EXECUTE ('select * from PapHonora'+@Mes+
      ' Where substring(Codigo,3,3)="'+@Dep+
      '" and Tipo_Contrato="'+@TC+'')
  Else /*Nivel SubDependencia*/
    EXECUTE ('select * from PapHonora'+@Mes+
      ' Where substring(Codigo,3,3)="'+@Dep+
      '" and substring(Codigo,6,2)="'+@SDep+
      '" and Tipo_Contrato="'+@TC+'')
  /*Ednfi*/
/*Endif*/
If @Tipo="I" /*Consulta para investigadores*/
  EXECUTE ('select *
    from PapHonora'+@Mes+ ' a, inves_codigo b
    Where a.codigo = b.codigo and b.clave = "' +@clave +'')
/*Endif*/
GO
```

Para ejecutar este procedimiento hay que pasar 7 parámetros, los primeros cuatro son obligatorios y los siguientes 3 son optativos, de tal manera que la llamada quedaría de la siguiente manera:

*Sp\_Pap\_Honora '416','01','10','G','HS'*

El ejemplo anterior, mostrará los contratos de manera general (G) realizados por la dependencia 41601 (Facultad de Ingeniería) durante el mes de octubre (10) cuyo tipo de contrato sea honorarios a sueldos (HS).

De hecho, como este último ejemplo en el SIAU existen más de 250 procedures, la mayoría de ellos utilizados para realizar las consultas dentro del sistema; algunos otros para el control de la seguridad y para la actualización de información dentro de las tablas de la base de datos. El cuadro siguiente muestra de manera general los procedimientos más utilizados dentro del sistema.

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
1	ConCodigo	Previos01..12	FrmPrevios.Frm	Realiza la consulta por código programático
2	ConCodigoTot	Previos01..12	FrmPrevios.Frm	Genera los totales por código
3	ConCuenta	Previos01..12	FrmPrevios.Frm	Realiza la consulta por cuenta contable
4	ConCuentaTot	Previos01..12	FrmPrevios.Frm	Genera los totales por cuenta
5	ConDependencia	Previos01..12	FrmPrevios.Frm	Realiza la consulta por dependencia
6	ConDependenciaTot	Previos01..12	FrmPrevios.Frm	Genera los totales por dependencia
7	ConGrupo	Previos01..12	FrmPrevios.Frm	Realiza la consulta por grupo
8	ConGrupoTot	Previos01..12	FrmPrevios.Frm	Genera los totales por grupo
9	ConPartida	Previos01..12	FrmPrevios.Frm	Realiza la consulta por partida de gasto
10	ConPartidaTot	Previos01..12	FrmPrevios.Frm	Genera los totales por partida
11	ConPrevio	Previos01..12	FrmPrevios.Frm	Realiza la consulta por número de previo
12	ConPrevioCta	PreviosCta01..12	FrmPrevios.Frm	Realiza la consulta por número de previo cuenta
13	ConPrevioCtaDep	PreviosCta01..12	FrmPrevios.Frm	Realiza la consulta por dependencia
14	ConPrevioCtaDepTot	PreviosCta01..12	FrmPrevios.Frm	Genera los totales por dependencia
15	ConPrevioCtaTot	PreviosCta01..12	FrmPrevios.Frm	Genera los totales por previo cuenta
16	ConPrevioTot	Previos01..12	FrmPrevios.Frm	Genera los totales por previo

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
17	ConPrograma	Previos01..12	FrmPrevios.Frm	Realiza la consulta por programa
18	ConProgramaTot	Previos01..12	FrmPrevios.Frm	Genera los totales por programa
19	ConSubDep	Previos01..12	FrmPrevios.Frm	Realiza la consulta por subdependencia
20	ConSubDepTot	Previos01..12	FrmPrevios.Frm	Genera los totales por subdependencia
21	ConGR6xx	GastosReserva01..12	FrmGastos.Frm	Realiza la consulta por cuenta de gasto
22	ConGr6xxTotCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales por cuenta
23	ConGr6xxTotRfc	GastosReserva01..12	FrmGastos.Frm	Genera los totales por RFC
24	ConGr6xxTSCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales por Subcuenta
25	ConGRFolio	GastosReserva01..12	FrmGastos.Frm	Realiza la consulta por folio
26	ConGRgral	GastosReserva01..12	FrmGastos.Frm	Realiza la consulta general
27	ConGRGralTotCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales generales por cuenta
28	ConGRGralTotRfc	GastosReserva01..12	FrmGastos.Frm	Genera los totales generales por RFC
29	ConGRGralTSCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales generales por subcuenta
30	ConGRRfc	GastosReserva01..12	FrmGastos.Frm	Realiza la consulta por RFC
31	ConGRRfcTotCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales por RFC/Cuenta
32	ConGRRfcTotRfc	GastosReserva01..12	FrmGastos.Frm	Genera los totales por RFC/RFC
33	ConGRRfcTSCta	GastosReserva01..12	FrmGastos.Frm	Genera los totales por RFC/Subcuenta

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
34	AnalmpCodigo	Codigosmen01..12	FrmPresup.Frm	Genera información para imprimir por código
35	AnalmpGrupo	Codigosmen01..12	FrmPresup.Frm	Genera información para imprimir por grupo
36	AnalmpPartida	Codigosmen01..12	FrmPresup.Frm	Genera información para imprimir por partida
37	AnalmpSDep	Codigosmen01..12	FrmPresup.Frm	Genera información para imprimir por Subdep
38	AnaliticoCodigo	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por código
39	AnaliticoDep	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por dependencia
40	AnaliticoGrupo	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por grupo
41	AnaliticoPartida	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por partida
42	AnaliticoSDep	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por subdependencia
43	AnaliticoSubProg	Codigosmen01..12	FrmPresup.Frm	Realiza la consulta por subprograma
44	AnaMoviCodigo	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por código
45	AnaMoviDep	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por dependencia
46	AnaMoviGrupo	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por grupo
47	AnaMoviPartida	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por partida
48	AnaMoviSubDep	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por subdep
49	AnaMoviSubDep	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por subdep

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
50	AnaMoviSubProg	Codigosmen01..12	FrmPresup.Frm	Muestra movimientos del mes por subprograma
51	Catalogo	CatPartidas	FrmPresup.Frm	Busca nombres en el catálogo
52	SP_TransPre	Transferencias	FrmTrans.Frm	Muestra las transferencias
53	MovIngExtCta	MovIngExt01..12	FrmIngEx.Frm	Muestra movimientos del mes por cuenta
54	MovIngExtDep	MovIngExt01..12	FrmIngEx.Frm	Muestra movimientos del mes por dependencia
55	MovIngExtFecha	MovIngExt01..12	FrmIngEx.Frm	Muestra movimientos del mes por fecha
56	MovIngExtFolio	MovIngExt01..12	FrmIngEx.Frm	Muestra movimientos del mes por folio
57	MovIngExtSDep	MovIngExt01..12	FrmIngEx.Frm	Muestra movimientos del mes por subdep
58	SaldoActCta	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos actuales por cuenta
59	SaldoActDep	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos actuales por dependencia
60	SaldoActSDep	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos actuales por subdependencia
61	SaldoAntCta	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos anteriores por cuenta
62	SaldoAntDep	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos anteriores por dependencia
63	SaldoAntSDep	SaldosMen01..12	FrmIngEx.Frm	Muestra saldos anteriores por subdependencia
64	PolizaDep	Polizas01..12	FrmPoliz.Frm	Realiza la consulta por dependencia
65	PolizaFecha	Polizas01..12	FrmPoliz.Frm	Realiza la consulta por fecha
66	PolizaFolio	Polizas01..12	FrmPoliz.Frm	Realiza la consulta por folio

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
67	PolizaPrevio	Polizas01..12	FrmPoliz.Frm	Realiza la consulta por previo
68	PolizaSubDep	Polizas01..12	FrmPoliz.Frm	Realiza la consulta por subdependencia
69	Sp_BoletosDep	Boletos01..12	FrmBolet,Frm	Realiza la consulta por dependencia
70	Sp_BoletosGral	Boletos01..12	FrmBolet,Frm	Realiza la consulta de forma general
71	Sp_BoletosLinea	Boletos01..12	FrmBolet,Frm	Realiza la consulta por línea aérea
72	Sp_BoletosSubDep	Boletos01..12	FrmBolet,Frm	Realiza la consulta por subdependencia
73	sp_DepConacyt	Depositos	FrmDepCo.Frm	Muestra información bajo cualquier criterio
74	Checafolio	Folios	FrmFolio.Frm	Verifica que exista el folio
75	DesErr	CatErrores	FrmFolio.Frm	Muestra descripción de errores
76	NumError	CatErrores	FrmFolio.Frm	Muestra el número de error
77	FoliosDep	Folios	FrmFolio.Frm	Realiza la consulta por dependencia
78	FoliosDepCheqME	Folios, Cheques	FrmFolio.Frm	Muestra cheques por dependencia
79	FoliosFecha	Folios	FrmFolio.Frm	Realiza la consulta por fecha
80	FoliosFechaCheq	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.n. por fecha
81	FoliosFechaCheqME	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.e. por fecha
82	FoliosNumero	Folios	FrmFolio.Frm	Realiza la consulta por numero de folio
83	FoliosNumeroCheq	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.n. por numero

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
84	FoliosNumeroCheqME	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.e. por numero
85	FoliosRechazos	Folios, Rechazos	FrmFolio.Frm	Realiza la consulta por numero de rechazo
86	FoliosSDep	Folios	FrmFolio.Frm	Realiza la consulta por subdependencia
87	FoliosSDepCheq	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.n. por subdependencia
88	FoliosSDepCheqME	Folios, Cheques	FrmFolio.Frm	Muestra cheques m.e. por subdependencia
89	sp_PapBecario	PapBecario01..12	FrmPap1.Frm	Muestra información bajo cualquier criterio
90	sp_PapBoletos	PapBoletos01..12	FrmPap1.Frm	Muestra información bajo cualquier criterio
91	sp_PapHonora	PapHonorarios01..12	FrmPap1.Frm	Muestra información bajo cualquier criterio
92	sp_PapHonoraSum	PapHonorarios01..12	FrmPap1.Frm	Muestra los totales bajo cualquier criterio
93	Sp_PapMovtos	PapMovimientos01..12	FrmPap4.Frm	Muestra información bajo cualquier criterio
94	sp_PapMovtosSum	PapMovimientos01..12	FrmPap4.Frm	Muestra los totales bajo cualquier criterio
95	sp_PapOtros	PapOtros01..12	FrmPap1.Frm	Muestra información bajo cualquier criterio
96	sp_PapPresup	PapPresupuesto01..12	FrmPap1.Frm	Muestra información bajo cualquier criterio
97	sp_PapPresupSum	PapPresupuesto01..12	FrmPap1.Frm	Muestra los totales bajo cualquier criterio
98	Sp_PapPrevio	PapPrevios01..12	FrmPap4.Frm	Muestra información bajo cualquier criterio
99	sp_PapPrevioSum	PapPrevios01..12	FrmPap4.Frm	Muestra los totales bajo cualquier criterio

No.	Nombre	Tablas que Utiliza	Módulo de V.B.	Descripción
100	sp_PapResProy	PapResumen01..12	FrmPapRe.Frm	Muestra información bajo cualquier criterio
101	sp_PapResPSum	PapResumen01..12	FrmPapRe.Frm	Muestra los totales bajo cualquier criterio
102	sp_PapRPxCod	PapResumen01..12	FrmPapRe.Frm	Muestra información bajo cualquier criterio
103	sp_PapSuperC	PapSuperCray01..12	FrmPapI.Frm	Muestra información bajo cualquier criterio
104	sp_PapSuperCSum	PapSuperCray01..12	FrmPapI.Frm	Muestra los totales bajo cualquier criterio
105	CambiaPass	Usuarios	FrmContr.Frm	Cambia la contraseña del usuario
106	ValidaSiau	Usuarios	FrmPrinc.Frm	Realiza la validación de acceso al sistema
107	Sp_HayMensaje	MensajeUsuario	FrmPrinc.Frm	Verifica si existe mensaje nuevo
108	Usu_Mensaje	Usuarios	FrmMens.Frm	Checa cuantos mensajes tiene el usuario
109	Sp_CircProc	Circulares	FrmCirPr.Frm	Muestra circulares bajo cualquier criterio
110	sp_FechaCP	Circulares	FrmCirPr.Frm	Realiza búsquedas por rangos de fechas

Es importante señalar que los stored procedures son herramientas poderosas cuando son usadas apropiadamente. Pueden considerarse cuando se construyen aplicaciones Cliente/Servidor, donde tendrán un impacto significativo en el rendimiento de la aplicación. Además sirven para reforzar la consistencia de los datos entre tablas. Las características de seguridad que confieren, permiten a las tablas ser modificadas únicamente de una forma predeterminada y, aún más, facilitan los cambios del diseño de la base de datos en tiempo de ejecución.

## **8. Crear los Disparadores(triggers)**

El concepto de disparador, tal como lo define SQL Server, es relativamente sencillo. Para cualquier evento que provoca un cambio en el contenido de una tabla, un usuario puede especificar una acción asociada que el manejador de bases de datos debería efectuar. Los tres eventos que pueden disparar una acción son intentos de INSERT, DELETE ó UPDATE sobre un registro de una tabla.

Este tipo de objetos tampoco fue necesario crearlos dentro del sistema, debido a que casi la totalidad de los procesos se refieren a consultas (SELECT).

## **9. Identificar a los Usuarios y definir los Grupos**

Al igual que dentro del sistema operativo, en el manejador de bases de datos, se deben dar de alta los usuarios y los grupos de usuarios que tendrán acceso a los diferentes objetos de las diferentes bases de datos que existan en el servidor.

Para nuestro sistema, sólo se crearon dos usuarios con el objeto de saber qué versión de software cliente estaba usando cada usuario conectado al sistema. Estos usuarios se integraron dentro de un grupo llamado GrpConsulta98, con el objeto de realizar la administración de los permisos de acceso a los objetos utilizando el nombre del grupo, lo cual hace que la administración de las bases de datos sea más sencilla y eficiente.

Para el manejo de claves de usuario dentro del sistema, se creó una tabla llamada *Usuarios*, donde se tiene la información más relevante relacionada a cada usuario como clave, contraseña, dependencia a la que pertenece, nombre, teléfono, etc. La creación de dicha tabla, obedeció fundamentalmente a que no era conveniente crear una cuenta de usuario a nivel SQL para cada usuario que fuera a acceder al sistema.

Esto se debe a que con el uso de ODBC, el cliente puede comunicarse a la base de datos utilizando cualquier herramienta que haga uso de ODBC, lo que afectaría severamente el carácter confidencial de la información.

## **10. Asignar Permisos a los Objetos**

Los permisos son las acciones que un usuario o un grupo de usuarios, tienen permitido efectuar para un determinado objeto de la base de datos. Un usuario puede tener permiso para **SELECT** e **INSERT** sobre registros en una tabla determinada, pero puede carecer de permiso para **DELETE** o **UPDATE**. Un usuario diferente, puede tener un conjunto diferente de privilegios.

Para otorgar permisos a los objetos y establecer un esquema de seguridad en una base de datos, se utiliza el comando **GRANT** con el que se especifica qué usuarios tienen qué permisos sobre qué objetos de la base de datos.

Para el sistema en cuestión, sólo se otorgaron permisos de **SELECT** sobre unas cuantas tablas ya que la mayoría de las consultas se realizan a través de procedimientos almacenados, por lo que al otorgar el permiso de ejecutar sobre un procedimiento, automáticamente se puede acceder a una tabla aún sin que el usuario tuviera permiso de *select* sobre ella. Como ejemplo se presentan las instrucciones para dar permiso de *select* sobre la tabla de Aplicaciones y permiso de *execute* sobre el procedimiento `sp_Hay_Mensaje`.

```
GRANT SELECT ON Aplicaciones to GrpConsulta98
```

```
GRANT EXECUTE ON sp_Hay_Mensaje TO GrpConsulta98
```

#### **4.2.5 Diseño y Construcción del Esquema de Seguridad**

La seguridad dentro del sistema en cuestión juega un papel muy importante, ya que la información residente en la base de datos consta de todo el presupuesto a nivel UNAM.

Debido a que el sistema trabaja sobre Red-UNAM, la información de la base de datos está expuesta a todos los usuarios de Internet, tanto nacionales como internacionales, y puede existir alguien que con malas intenciones intente filtrarse. Por tal motivo, es de suma importancia que ninguna persona ajena, interna o externa a la Institución, pueda consultar, modificar o borrar información de la base de datos del sistema.

Para ello, en el sistema se implantó un mecanismo que abarca importantes aspectos de seguridad (acceso, disponibilidad, privacidad, integridad, consistencia y auditoría), mismos que tendría que conocer y violentar aquel que de manera ilícita quisiera acceder a la información confidencial.

En el nivel de acceso es dónde se decide quién puede acceder a consultar la información. Para el SIAU se implantaron los siguientes mecanismos:

##### **a) Dirección IP del servidor**

Cada computadora conectada a Internet y que trabaja con el protocolo de comunicación TCP/IP, tiene asignada una dirección IP única que la hace diferente a las demás dentro de la red. Para pasar por el nivel de seguridad a) se tiene que conocer la dirección IP del servidor de base de datos, que al igual que todas las direcciones dentro de la red, es del tipo w.x.y.z; donde w,x,y y z pueden tener valores que van desde 1 hasta 254.

### **b) Número de Puerto de Comunicación (Socket)**

Los sistemas que utilizan protocolo TCP/IP, definen *socket* como un punto virtual de comunicación que permite intercambiar datos con otro proceso de usuario. Este último proceso puede estar situado en la misma máquina o en otra conectada a la Red. Al igual que los números de puerto en los protocolos de transporte permiten la utilización del canal de comunicación físico por varios procesos sin entorpecerse, los sockets realizan la misma función a nivel de programación.

Por lo tanto, como idea fundamental un socket se concibe como un canal que puede abrirse con una dirección de destino; a través de él pueden enviarse y recibirse datos, existiendo también la posibilidad de que el canal se cierre.

Los TCP/IP sockets están formados por la unión de la dirección IP más el número de puerto de comunicación. El número de puerto es un número de 16 bits. Existen tres tipos de puertos: los “bien conocidos” que van de 0 a 255, los “privilegiados” que van de 256 a 1,024 y los “definidos por el usuario” que van de 1,025 a 65,536.

Para pasar sobre el nivel de seguridad b), el intruso tendrá que saber el número de puerto que se le asignó a SQL Server cuando fue instalado.

### **c) Nombre del servidor**

Así como las PC's y servidores dentro de la red cuentan con una dirección IP, también tienen un identificador de tipo carácter, a través del cual se puede controlar el uso y acceso a los equipos a nivel de aplicación. El nombre del servidor deberá ser de al menos 8 caracteres y podrá contener mayúsculas, minúsculas y caracteres alfanuméricos para hacer más difícil su identificación.

#### **d) Configuración de ODBC**

Existen dentro de windows algunos archivos de inicialización que deberán ser editados y modificados correctamente para poder hacer uso de la base de datos del servidor a través de las aplicaciones cliente. Estos archivos son el `odbc.ini`, el `win.ini` y el `system.ini`, a los que se les deberán agregar líneas de configuración de acuerdo con la información vertida en los puntos a), b) y c).

#### **e) Software Cliente (Front-End)**

Una vez que han sido bien configurados los archivos para trabajar con ODBC, se requiere de una aplicación cliente (front-end) que trabaje bajo ambiente windows y que tenga la posibilidad de acceder a bases de datos vía ODBC. En el mercado actual existen muchas herramientas que cuentan con estas características, entre las cuales se puede citar MS Word, MS Excel y MS Query que es software que viene incluido con MS Office 97. Por otro lado, entre los lenguajes de programación más populares que cumplen con los requisitos se pueden citar: Visual Basic, Visual Fox, Visual C++, Power Builder y Delphi.

Ahora bien, dada la facilidad con la que cualquier usuario puede conseguir dicho software, y con ello, adquirir la posibilidad de acceder al servidor de base de datos (siempre y cuando logre descifrar los puntos a) al d)), sólo se creó una cuenta dentro de la base de datos para todos los usuarios, con la que se les permite acceder vía el programa de Visual Basic. Adicionalmente, ésta cuenta no tiene permisos de *Select* sobre la mayoría de las tablas. Sin embargo, si permite el acceso a los procedimientos almacenados, con permiso para ejecutar y acceder a la información. Finalmente, aunque un usuario llegara a descubrir esta cuenta, deberá contar con suficiente experiencia en el manejo de SQL para poder consultar la información,

pero nunca podrá modificarla.

Aunado a todo esto, se decidió agregar un mecanismo más de seguridad, mismo que se explica a continuación.

#### **f) Clave y Contraseña Internas y Externas**

Como se explicó en una sección previa, sólo existen dos claves de usuario (internas) dadas de alta dentro de la base de datos del sistema; éstas se encuentran dentro de un grupo de usuarios, también pertenecientes a la base de datos, que en el sistema se identificó como *Consulta*.

Para que los más de 500 usuarios del sistema pudieran acceder la base de datos, se creó dentro de *Consulta*, una tabla llamada *Usuarios* donde se guarda información relevante de cada uno de ellos. Entre los datos más relevantes que se guardan en esta tabla se pueden mencionar la clave y la contraseña (externas) que les fueron asignadas por el administrador del SIAU al momento de darlos de alta.

Una vez que el usuario posee su clave y contraseña externa, envía, a través del software cliente (Visual Basic), tanto las claves y las contraseñas internas, como las externas. Con ello, se permite la validación dentro de SQL y, posteriormente, se ejecuta un procedimiento almacenado para preguntar si el usuario externo existe en la tabla de *Usuarios*. Si el proceso concluye con éxito el usuario está en posibilidad de utilizar el sistema; de lo contrario, el sistema muestra en pantalla un mensaje de error explicando la causa de la negación de acceso.

En cuanto al nivel de disponibilidad de información empleada, para proteger los servicios de cómputo de manera que no se degraden o no dejen de estar disponibles a los usuarios, se realizan varias medidas, tales como: el respaldo de información

diaria, copias en cinta y uso de herramientas de monitoreo del mismo Sistema Operativo y de la Base de Datos.

En el renglón de la privacidad, que consiste en no permitir la consulta de información a personas no autorizadas, se crearon los mecanismos que se explican a continuación:

#### **g) Dirección IP del Cliente**

Este mecanismo de seguridad para evitar intrusos en el sistema, se estableció obteniendo la dirección IP de la computadora cliente. Esta información es almacenada en la tabla de usuarios al momento que el usuario está instalando el sistema por primera vez. De hecho, el programa de instalación es el que se encarga de recoger toda la información del usuario (entre la que se encuentra la dirección IP) y almacenarla en la tabla de *Usuarios* perteneciente a la base de datos *Consulta*.

De tal manera que si un usuario a pesar de tener clave y contraseña válidas dentro del sistema, quiere acceder desde un punto de red diferente al de donde se instaló el sistema por primera vez, el acceso le será denegado. Lo anterior, establece una restricción de acceso a cierto grupo de usuarios.

#### **h) Permiso de Acceso a los Objetos de la Base de Datos**

Como se mencionó anteriormente, sólo existen dos claves de usuario dadas de alta en SQL y únicamente pueden acceder a la base de datos *Consulta*. Sin embargo, cada usuario sólo tiene permiso de consultar (SELECT) y actualizar (UPDATE) en la tabla de *Usuarios*; en el resto de las tablas no se concedió permiso alguno, debido a que éstas se accesan vía procedimientos almacenados, en los cuales el usuario tiene permiso de ejecutar (EXECUTE) sobre cada uno de los

procedimientos.

En los niveles de integridad y consistencia, se protege la información a ser consultada o modificada, además de que se trabaja para que el sistema se comporte en la forma esperada. Estos mecanismos suelen ser transparentes para los usuarios, ya que el origen de la información tiene sus propias medidas de seguridad para evitar datos inconsistentes. Para el caso del SIAU, estas medidas son implantadas por el Departamento de Procesamiento de Datos, instancia encargada de generar la información diariamente, para posteriormente insertarla en la base de datos.

El último nivel de seguridad se refiere a la Auditoría, donde se hace uso de herramientas de monitoreo y uso de bitácoras de acceso, recursos y tiempo de acceso. Por un lado, se usan las mismas del Sistema Operativo, SQL Server y en el mismo sistema se cuenta con algunos campos en la tabla de usuarios donde se actualiza información, tal como tiempo de uso, hora de entrada y salida, entre otros.

En resumen, se puede decir que para que un usuario externo a la Universidad pueda ingresar al SIAU, tendrá que conocer la dirección IP del servidor de base de datos, el número de puerto de comunicación y el nombre del servidor; además, deberá haber configurado adecuadamente su PC para trabajar con ODBC, tener un software cliente para ambiente windows, contar con una clave, contraseña y dirección IP válidas dentro del sistema y, lo más importante, tener experiencia en el uso y manejo de MS SQL Server.

En términos generales se puede comentar que las medidas implantadas garantizan, en un alto porcentaje, la no-intromisión de usuarios internos y externos no deseados. Sin embargo, a pesar de que dichas medidas reducen el riesgo de un ataque o incidente; se sabe que entre más dinero, tiempo y esfuerzo se invierta en

los sistemas de seguridad en función de la trascendencia de la información, ésta se encontrará más protegida.

#### **4.2.6 Diseño y Construcción de la Interfaz del Usuario**

Para la mayoría de los usuarios, la interfaz es el sistema. Aunque esté bien o mal diseñada, la interfaz permanece como una representación de la aplicación y, por lo tanto, es de la competencia del analista de sistemas.

De esta manera, se debe tener como objetivo el diseño de una interfaz que ayude a los usuarios a obtener, o bien, a introducir información al sistema, cumpliendo los siguientes objetivos:

**Eficacia**, al lograr, mediante el diseño de interfaces, que el usuario tenga acceso al sistema, de tal forma que sea congruente con sus necesidades particulares.

**Eficiencia**, demostrada a través de interfaces que mejoren la velocidad de captura de los datos y reduzcan los errores.

**Consideración del usuario**, al demostrar un diseño adecuado de la interfaz que favorezca la retroalimentación del sistema para los usuarios en forma apropiada.

**Productividad**, considerada a través del apego a los principios del diseño ergonómico de las interfaces de los usuarios y de sus áreas de trabajo.

Existen diferentes tipos de interfaces para los usuarios, incluyendo las de lenguaje natural; las consultas de preguntas y respuestas; los menús; las formas de entrada/salida; el lenguaje por comandos y el manejo directo. La interfaz cuenta con dos componentes principales: el lenguaje de presentación, que es parte de la relación computadora-hombre y el lenguaje de acción que caracteriza la parte

hombre-computadora. En conjunto, ambos conceptos cubren la fórmula y el contenido del término interfaz del usuario.

Para el sistema en cuestión, la interfaz de menús y el uso del mouse (ratón) fue la que se consideró más adecuada a las necesidades planteadas, debido a que representaba mayores ventajas en relación con el universo de usuarios del SIAU, tomando en cuenta la poca experiencia que la mayoría de ellos tienen en el uso y manejo de equipos y programas de cómputo.

#### **4.2.6.1 Interfaz de Menús**

Esta interfaz toma su nombre a partir de la lista de platillos que pueden seleccionarse en un restaurante. De manera similar, una interfaz de menú permite que el usuario elija alguna de las posibles opciones ofrecidas dentro de una lista que se presenta en pantalla.

Al responder el usuario al menú, se limita a las opciones que se le presentan. Él no necesita conocer el sistema pero sí requiere saber qué tareas pueden realizarse. Por ejemplo, en un menú de un típico procesador de palabras, los usuarios pueden elegir entre “copiar”, “cortar” o “pegar”. Sin embargo, para sacar el máximo provecho del menú, los usuarios deben estar al tanto de las tareas potenciales del sistema y de lo que realmente desean hacer.

Los menús, considerados como interfaces, no son dependientes del hardware pero existen variantes que derivan del equipo disponible, es así que, puede tenerse acceso a los menús a través del teclado, de lápices ópticos o de ratones. Por otro lado, las secciones pueden identificarse mediante un número, una letra, una clave y en ambiente gráfico (Windows) con un botón o una caja de selección.

En la figura 4.19 se muestra un ejemplo del programa orientado a través de menús, del SIAU. El usuario cuenta con nueve acciones diferentes a realizar para la consulta de información. Con el fin de elegir la opción "Información Presupuestal y de Gestión", el usuario únicamente debe pulsar doble click sobre dicha opción, o bien, seleccionar la opción y después dar click en el botón *Aceptar*. Esto lo guiará a una nueva pantalla, la cual estará esperando la selección de una nueva opción.

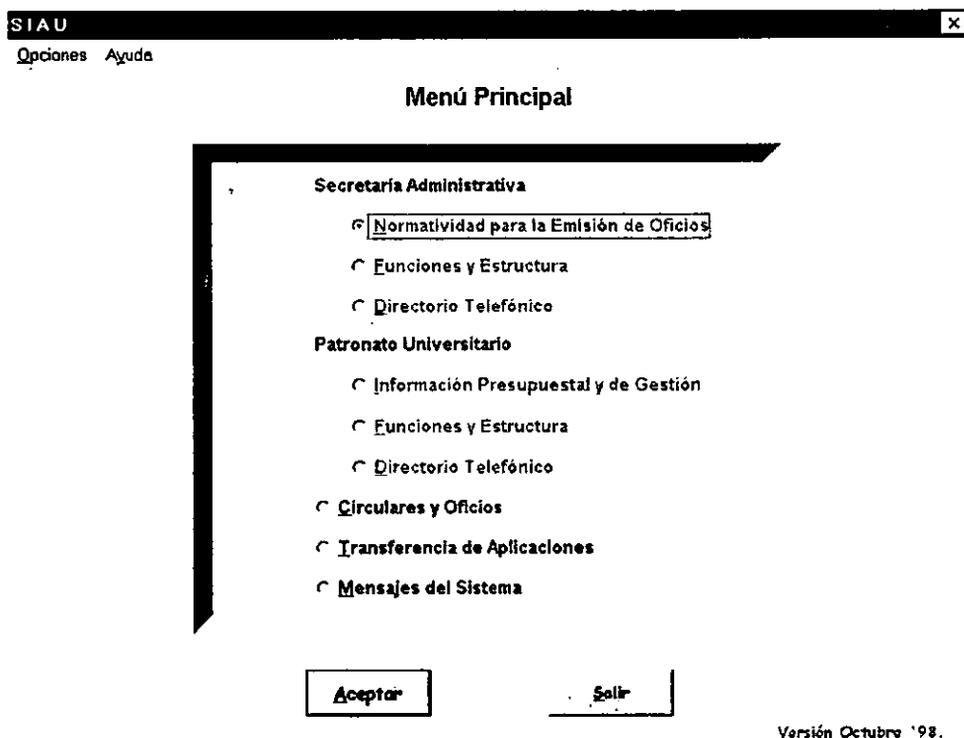


Figura 4.19 Programa Orientado por Menús

Al diseñar una interfaz de menú es importante mantener la consistencia. Con el fin de tener acceso a una sección del menú, el usuario debe presionar las teclas de dirección (flechas) para seleccionar la opción deseada; después presionar la tecla ENTER, o bien, dar doble click con el Mouse sobre la opción a consultar; entonces la computadora se dirigirá a la siguiente rutina.

Los menús pueden encontrarse anidados, uno dentro del otro, de tal forma que el usuario los debe solicitar conforme los va requiriendo, tal y como se muestra en la figura 4.20

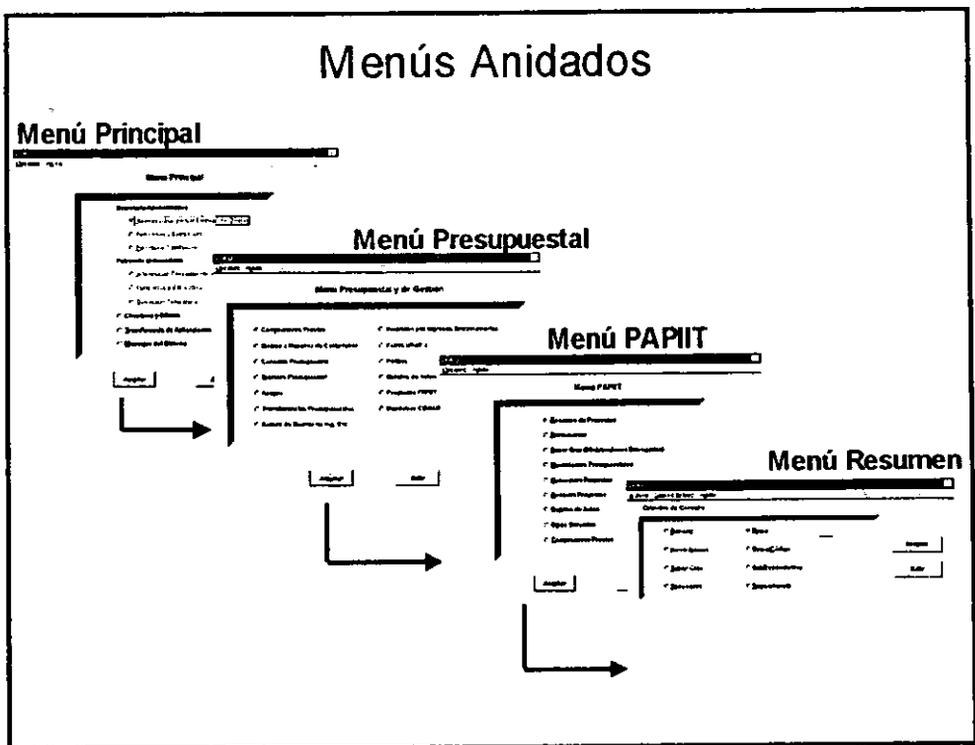


Figura 4.20 Uso de Menús Anidados

Este ejemplo muestra cuatro pantallas sucesivas de menús. El usuario comienza con el menú principal y luego selecciona la opción 4 para ingresar al menú *Presupuestal y de Gestión*. De ahí selecciona la opción 12 para acceder al menú *PAPIIT*; esto lo lleva a un nuevo menú de *Resumen de Proyectos* donde se muestran ocho opciones específicas para consulta de información de proyectos. Si el usuario elige una de las opciones podrá ver en pantalla la información relacionada al rubro de consulta seleccionado, para que finalmente, auxiliado de la barra de menús superior, pueda imprimir o guardar en un archivo los datos mostrados en pantalla.

Los menús anidados tienen ciertas ventajas. Permiten que la pantalla se mantenga despejada, lo cual es consistente con un buen diseño de pantalla. Los menús anidados evitan que el usuario siga viendo aquellas opciones del menú que ya no son de su interés, y por tanto, se reduce la saturación de información irrelevante. Además, los menús anidados permiten un movimiento rápido del usuario a lo largo del programa.

No obstante que, en general, son una bendición para los inexpertos, los menús anidados pueden presentar ciertos problemas para los usuarios experimentados. Los usuarios experimentados pueden impacientarse al tener que elegir diferentes opciones en menús sucesivos antes de llegar a la rutina que desean. Una manera de solucionar esto es proporcionar la alternativa de que todas las opciones de los menús puedan invocarse mediante un sólo comando de línea. Esto hace que el sistema sea atractivo tanto para los usuarios expertos como para los que no lo son.

## 4.2.7 Programación y Desarrollo

En esta etapa del ciclo de desarrollo de los sistemas, el analista trabaja conjuntamente con los programadores, aquí es donde les transmite todos los requerimientos de programación para desarrollar el software original.

### 4.2.7.1 Diagrama estructural

El instrumento recomendado para el desarrollo de un sistema modular descendente se denomina diagrama estructural, el cual muestra la partición del sistema en módulos y la relación jerárquica entre éstos, además de mostrar los flujos de información y control entre los módulos.

Un diagrama estructural cuenta principalmente con los elementos que se muestran en la Figura 4.21 y que se describen a continuación:

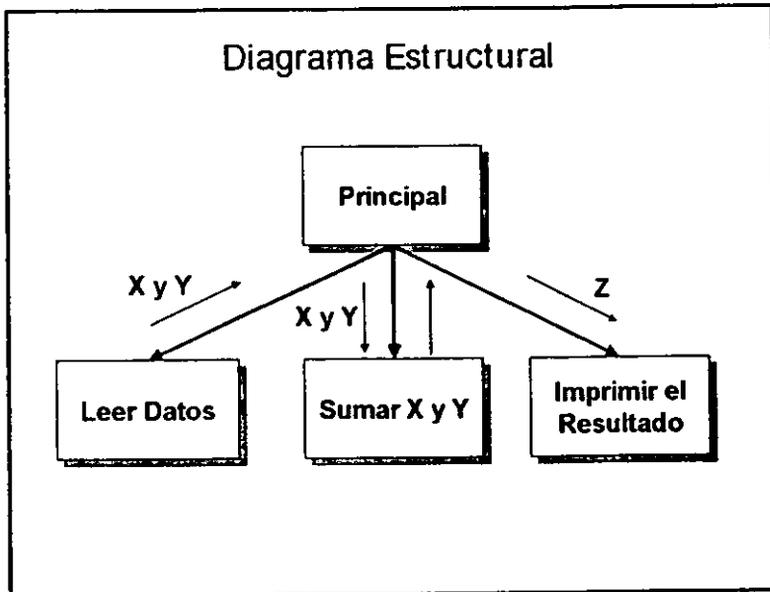


Figura 4.21 Ejemplo de un Diagrama Estructural



Cada uno de los módulos presentados en la figura anterior, se descompone en varios submódulos, sin embargo, la gran mayoría de ellos tienen gran similitud en sus procesos internos, por tal motivo, y para efectos del presente documento, a continuación sólo se detallará el módulo de Ejercicio Presupuestal, que corresponde al que se mostró a detalle durante la fase de análisis.

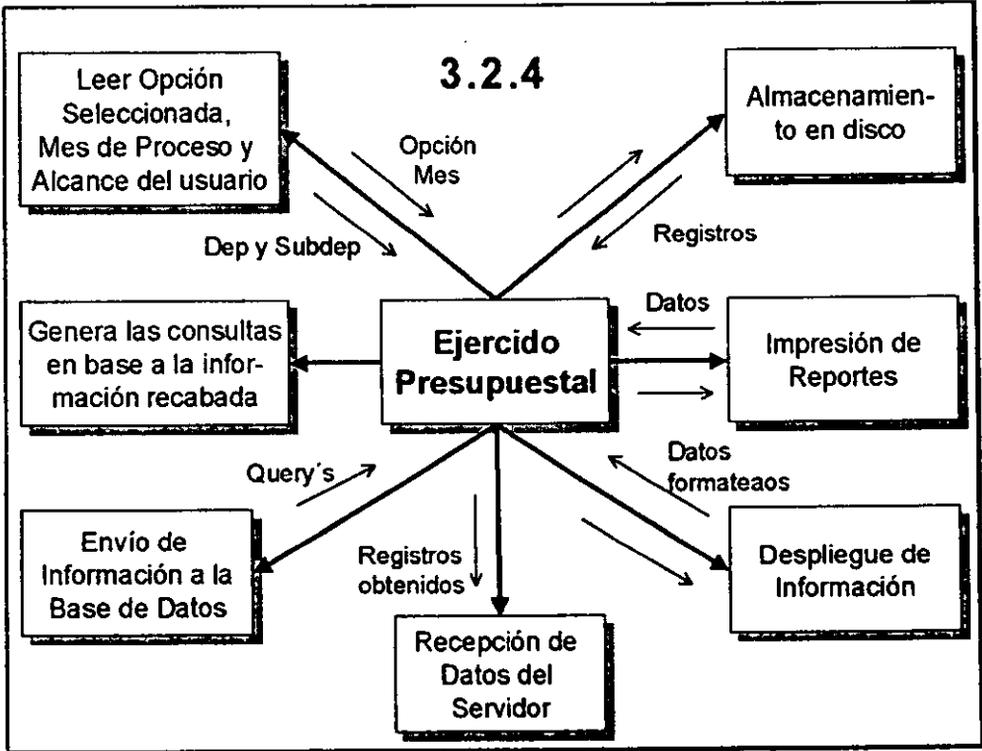


Figura 4.23 Diagrama Estructural del Módulo Ejercicio Presupuestal

El siguiente paso para el desarrollo del SIAU fue iniciar la programación del sistema, para lo cual se utilizó el lenguaje de programación Visual Basic (VB) en su versión 3.0, considerando las ventajas que fueron comentadas a detalle en el tercer Capítulo del presente trabajo.

Durante el desarrollo de aplicaciones con Visual Basic, intervienen cinco etapas. La primera de ellas consiste en planificar lo que el usuario verá, en otras palabras, se diseñan las pantallas. Explotando los diferentes controles de VB se puede crear la parte Visual del desarrollo de la aplicación, es decir, sólo se define la parte de los elementos contenidos en las ventanas (denominadas comúnmente formas). A continuación en la figura 4.24 se muestra el diseño inicial de la forma correspondiente a *Ejercicio Presupuestal*.

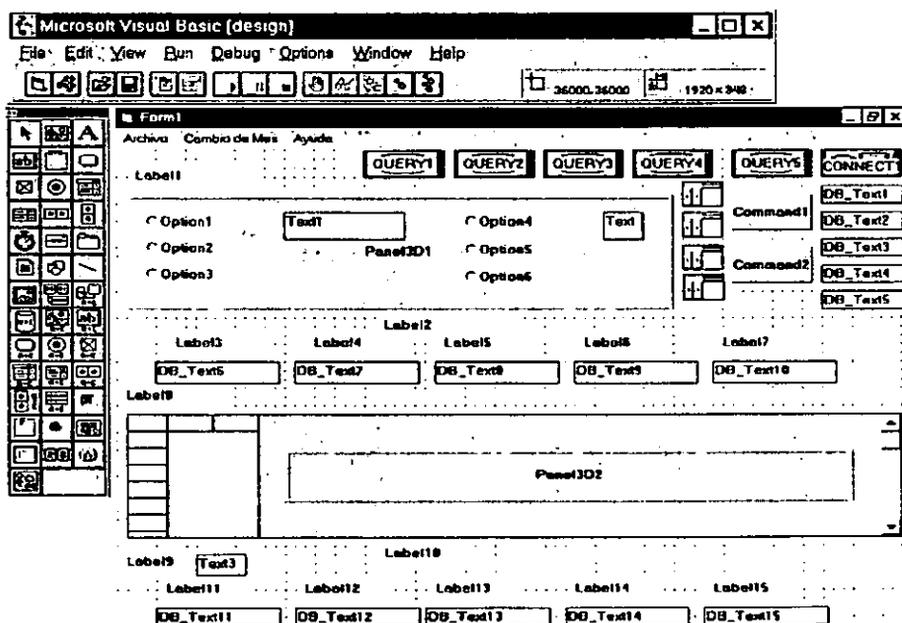


Figura 4.24 Diseño de la Forma de Ejercicio Presupuestal

El número de controles que se pueden usar dentro de una forma depende de la versión de Visual Basic que se utilice, de hecho para el desarrollo del presente trabajo se emplearon controles comerciales y controles de dominio público, además de los que vienen integrados en la versión 3.0 de Visual Basic.

Por otra parte, el segundo paso consiste en configurar las propiedades de los objetos dibujados sobre las formas para darles la apariencia deseada. Entre las propiedades se encuentran: el nombre, título, dimensiones, color, tipo y tamaño de letra, ubicación dentro de la forma, etc. Para configurar las propiedades de los objetos se utiliza la ventana de propiedades de Visual Basic, tal y como lo muestra la figura 4.25

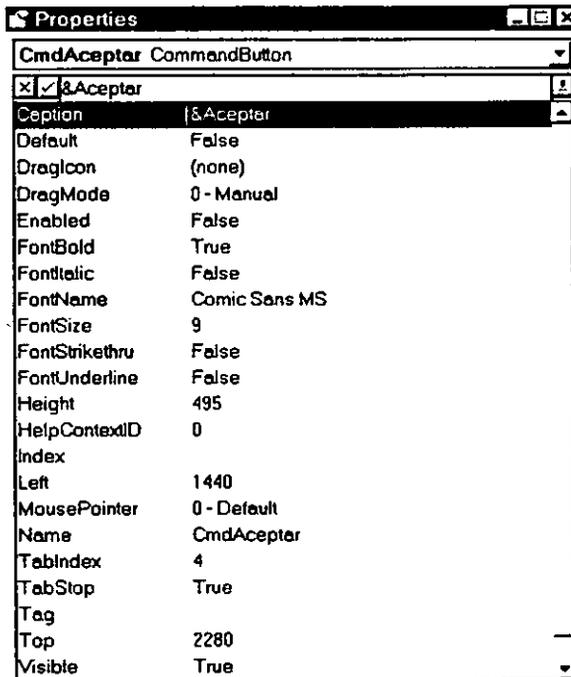


Figura 4.25 Propiedades del botón Aceptar

Para la forma de *Ejercido Presupuestal* y en general para el resto de las formas, se utilizaron diversos tipos de controles; entre los que se pueden mencionar: botones de comando, labels, grids y paneles principalmente, además de que a todas se les agregaron tres menús de opciones en la parte superior izquierda.

El primer menú llamado *Archivo* contiene cuatro opciones, la primera de ellas *Guardar*, permite al usuario almacenar los datos consultados en pantalla, en un archivo de tipo texto (\*.TXT) o de base de datos (\*.DBF). La segunda opción *Imprimir*, genera un reporte impreso de los datos consultados. La opción *Terminar Aplicación*, permite salir del sistema sin tener que regresar por todas las pantallas; finalmente, la opción *Salir*, regresa al usuario a la pantalla anterior.

El menú *Cambio de Mes* ofrece doce opciones, correspondiendo cada una de ellas al mes que se desea consultar.

El menú *Ayuda* ofrece tres diferentes opciones para buscar información acerca de cualquier rubro del sistema.

Cabe señalar que se estableció un estándar a fin de definir los nombres de los objetos, de tal manera que fuera más fácil utilizarlos durante la programación y que sirvieran para que cuando un programador externo requiriera de hacer un cambio en algún módulo, pudiera identificarse rápidamente con la lógica del programa. De esta manera, al nombre de los objetos le antecede un *Frm* si se trata de una forma, un *Txt* si es una caja de texto, un *Lbl* para una etiqueta, un *Mnu* para un menú, un *Glb* para una variable global, un *Cmd* si es un botón de comando y un *Opt* para seleccionar una opción.

La figura 4.26 muestra la misma forma de *Ejercido Presupuestal*, pero con las propiedades de los objetos modificadas, de hecho es muy similar a la forma que el

usuario observa en pantalla cuando está ejecutando el programa.

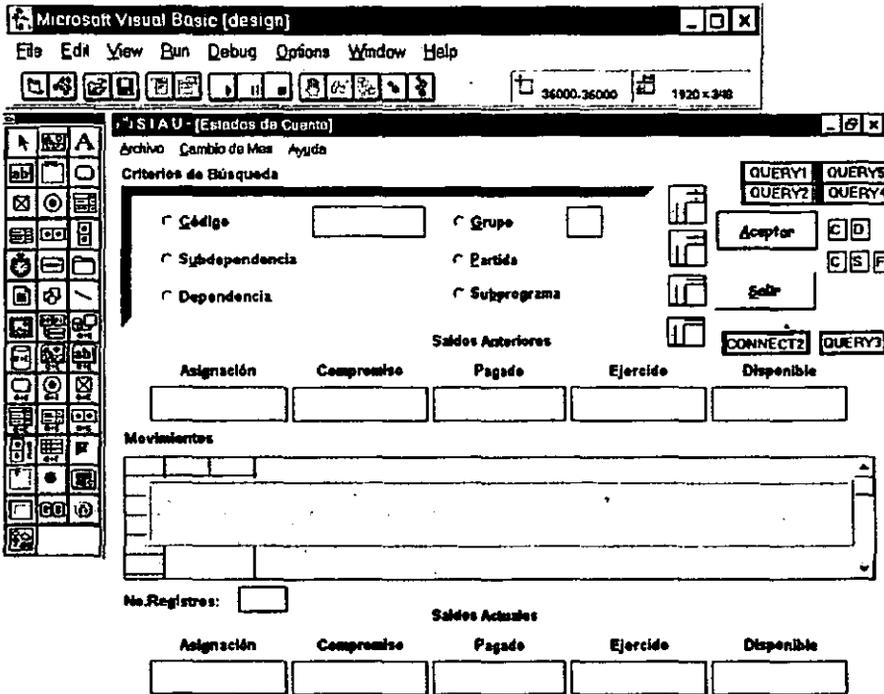


Figura 4.26 Forma de Ejecido Presupuesta! configurada

Una vez diseñada la interfaz y configuradas las propiedades de todos los objetos, es posible trabajar como en la programación tradicional. Esto lleva al tercer paso en la programación de Visual Basic: la escritura del código para activar la interfaz visual construida en los primeros dos pasos.

La clave de este paso consiste en que los objetos en Visual Basic reconocerán eventos tales como pulsaciones de ratón; la forma en que los objetos respondan a ellos depende del código que se escriba. Casi siempre será necesario escribir código para hacer que los controles respondan a los eventos. Esto hace que la programación en Visual Basic sea fundamentalmente diferente de la programación

convencional.

Los programas desarrollados en lenguajes de programación convencionales se ejecutan de arriba hacia abajo, es decir, la ejecución comienza en la primera línea y se mueve con el flujo del programa a las diferentes partes del programa cuando es necesario. Un programa Visual Basic trabaja de forma completamente distinta. En el fondo, un programa Visual Basic es un conjunto de fragmentos independientes de código que son activados mediante y, por tanto, responden sólo a los eventos que se ha indicado que reconocerán. Dependiendo de la complejidad de nuestros programas, se tendrá cierto código, que será ejecutado cuando se presente el evento al que se encuentra asociado.

La mayoría del código de programación en Visual Basic que indica al programa cómo responder a eventos, tales como pulsaciones del ratón, suceden en lo que Visual Basic denomina *procedimientos de evento*, mejor conocidos como *procedimientos*.

Un procedimiento no es más que el código necesario para indicar a Visual Basic cómo responder a un evento. Esencialmente todo lo que se ejecuta en un programa Visual Basic, es un procedimiento o es utilizado por un procedimiento para llevar a cabo su trabajo.

Dentro de cada uno de los módulos del SIAU se desarrollaron numerosos procedimientos para llevar el control de cada módulo y de todo el programa. Se definieron variables para cada forma, estas variables son locales y pierden su valor al cerrar la forma, por lo que para contar con variables globales se adicionó un módulo independiente al proyecto llamado GLOBS.BAS en donde se encuentran todas las variables, procedimientos y funciones que son utilizados por cualquier

procedimiento perteneciente a cualquier módulo del sistema. Como ejemplo, a continuación se muestra el código que conforma el procedimiento *Consulta* de la forma de *Ejercido Presupuestal*.

```
Sub Consulta (Proc1, Proc2 As String)
  Query4.pTables = Proc1 'procedure que solicita saldos
  Res% = FDoQuery(Query4)
  NumError = Query4.pErrorDB, RutError NumError
  If NumError = 156 Or NumError = 208 Then 'Si hay regs.
    Res% = FEndQuery(Query4), Exit Sub
  End If
  If Query4.pRecCount < 1 Or Txt_Act_Asig = "" Then
    Aparece 2 'Desaparece los objetos de la pantalla
    MsgBox "No hay Registros", 64, "; A V I S O !"
    If TxtCodigo.Visible = True Then
      TxtCodigo.SetFocus
    ElseIf TxtCodigo2.Visible = True Then
      TxtCodigo2.SetFocus
    End If
  Else
    Aparece 0 'Muestra los objetos en pantalla
    'Se configuran las propiedades del Grid
    QGridM.pColHeader = ", Código, Tipo, Folio, , _
                        Compromiso, Pagado, Disponible"
    QGridM.pColWidth = "600,1700,600,900,1000,1350,1350,1350"
    QGridM.pFormatString = ",, ,, ,dd/mm/yy, '#,##0.00;_
                          -#,##0.00', '#,##0.00;#,##0.00'"
    QGridM.pQueryExpr = "1,2,3,4,5,6,7"
    QGridM.pTables = Proc2
    Res% = FDoQuery(QGridM)
    NumError = QGridM.pErrorDB, RutError NumError
    If QGridM.pRecCount = 0 Then
      MsgBox "No hay Movimientos", 64, "; A V I S O !"
      If TxtCodigo2.Visible = True Then
        TxtCodigo2.SetFocus
      ElseIf TxtCodigo.Visible = True Then
        TxtCodigo.SetFocus
      End If
    Else
      TxtRegs = QGridM.pRecCount
      Aparece 1
    End If
  End If
  Res% = FEndQuery(Query4)
End Sub
```

De acuerdo a la opción que el usuario haya seleccionado, se manda llamar al procedimiento *Consulta* al cual se le pasan dos parámetros correspondientes a los procedimientos que mostrarán en pantalla los saldos anteriores del mes, los movimientos realizados y los saldos actuales al día de consulta.

Una vez realizada la consulta y mostrados los datos en la pantalla del usuario, éste último podrá elegir entre guardar la información obtenida en el disco duro de su PC o imprimirla a través de un reporte, para ello se utilizan los procedimientos *Guarda* e *Imprime* que se encuentran también dentro de la forma Ejercido Presupuestal, y que se encargan de realizar la lectura de los datos en pantalla para posteriormente enviarla ordenada y organizada a la impresora, o bien a un archivo de texto (.TXT) o de base de datos (.DBF).

El cuarto paso en la creación de programas con Visual Basic, consiste en la búsqueda de errores en el código (depuración del programa) y su posterior corrección. Cabe destacar que la búsqueda de errores en desarrollos cliente/servidor es más compleja, debido a que los errores pueden provenir del servidor (procedimientos almacenados) y no únicamente del elemento cliente.

Una vez que se han cumplido satisfactoriamente los cuatro pasos anteriores, se procede a la etapa final que es la creación del programa ejecutable (en el caso del SIAU, el ejecutable es el SIAU.EXE), para su posterior distribución a todas las PC's que trabajarán como cliente.

### **4.3 Pruebas del Sistema**

La etapa de pruebas durante el ciclo de desarrollo del software, conforma una serie de pasos esenciales que ayudan a garantizar la calidad del sistema. Es menos grave evaluar de antemano, que tener un sistema pobremente evaluado y que falle una vez instalado. La evaluación se lleva a cabo conforme progresa el trabajo en los módulos del programa; ésta se realiza a diferentes niveles y con varios intervalos, aún antes de que el sistema entre en operación, todos los programas deben examinarse en cuanto a su diseño con datos de prueba, a fin de verificar si los módulos se enlazan entre sí, tal y como fue planeado.

El programador realiza algunas pruebas por su cuenta y otras se llevan a cabo en colaboración con el analista de sistemas. En un principio, se realiza una serie de pruebas con datos tipo para identificar las posibles fallas del sistema; más adelante, se utilizarán los datos del sistema real.

También debe probarse el sistema trabajando como una unidad. Esto incluye la evaluación para las interfaces entre los subsistemas, la operación adecuada de la salida, la utilidad y comprensión de la documentación del sistema y de la salida. Los programadores, analistas, operadores y usuarios, todos ellos, juegan diferentes papeles que abarcan diversos aspectos de la evaluación.

Este es un importante periodo para evaluar la manera en que los usuarios finales interactúan con el sistema. Aunque se haya pensado mucho sobre tal interacción usuario-sistema, nunca se podrá predecir completamente el amplio espectro de diferencias sobre la manera real de las interacciones entre los usuarios y el sistema.

Los elementos a observar son: la facilidad de aprendizaje del sistema; el ajuste de factores ergonómicos; las reacciones de los usuarios a la retroalimentación del

sistema, incluyendo lo que ocurra cuando se presente en pantalla un mensaje de error y lo que ocurre cuando el usuario se entera de que el sistema está ejecutando sus comandos.

La etapa de pruebas del sistema se dividió en cinco fases que fueron las siguientes:

- Prueba Unitaria de Procedimiento
- Pruebas Integrales por Módulo
- Pruebas Integrales del Sistema
- Pruebas de Desempeño
- Pruebas de Recuperación

### **Prueba Unitaria de Procedimiento**

En primer lugar, la prueba unitaria se realiza sobre los procedimientos contenidos dentro de un módulo (forma), cada uno de manera independiente, a fin de localizar posibles errores de codificación y lógica que estén contenidos en su código. En este momento es cuando se detecta si está bien generada una llamada (cadena) a un procedimiento almacenado del servidor, así como los datos que éste último regresa.

En el caso del SIAU, los primeros procedimientos que se probaron fueron los de consulta, es decir, todos aquellos que mostraban información directamente en la pantalla del usuario; posteriormente, se trabajó con los que generaban las impresiones y, por último, con aquellos que guardaban en disco la información consultada.

## **Pruebas Integrales por Módulo**

En esta etapa se evalúa el conjunto de procedimientos que conforman un módulo, por ejemplo, el de Compromisos Previos. Este módulo está integrado por varios procedimientos que realizan una tarea específica, pero que interactúan dependiendo del buen funcionamiento de los otros procedimientos.

Para llevar a cabo esta evaluación, se realizó una matriz de pruebas para ensayar el ciclo completo (consulta, impresión y almacenamiento) de cada una de las operaciones relacionadas con todos los módulos.

En esta etapa fue donde el SIAU presentó más errores, principalmente en el proceso de impresión, ya que al probar el ciclo completo del módulo surgieron opciones que no se habían contemplado anteriormente.

## **Pruebas Integrales del Sistema**

En esta evaluación se observó cómo funcionaba el sistema completo en un ambiente de producción. Una vez que todos los módulos estaban listos y probados individualmente se podía pensar que el sistema estaba listo, sin embargo, después de este periodo de pruebas se detectó que ciertos módulos tenían errores cuando trabajaban en interacción con otros. Por ejemplo, el módulo GLOBS.BAS, en donde se almacenan los procedimientos, funciones y variables globales para todo el sistema, presentó varios problemas de incompatibilidad con el resto de los módulos.

Los problemas detectados se encontraron dentro del grupo de usuarios que tenían acceso a nivel Institucional, ya que una vez que buscaban información a nivel dependencia, les era imposible regresar a consultar de forma general. Este y otros

errores observados fueron solucionados oportunamente.

### **Pruebas de Desempeño**

Estas pruebas fueron realizadas en días y horas pico, ejecutando el sistema simultáneamente en 15 equipos y entrando a los diversos módulos para realizar consultas. En los casos de las dependencias que se encuentran en el interior de la república o que se conectan vía Módem, el tiempo máximo de respuesta fue de 5 segundos; mientras que para las que se encuentran dentro de ciudad universitaria y zona conurbada, el tiempo de respuesta fue de sólo un segundo, lo que hace un promedio de 3 segundos como tiempo de respuesta.

Esta característica, representa la ventaja principal del sistema, si se considera que anteriormente las dependencias recibían sus reportes 30 días después de haber generado sus operaciones en las unidades de proceso. Cabe mencionar, que el tiempo de respuesta también depende de las características de hardware y software con las que cuente la computadora cliente.

### **Pruebas de Recuperación**

En esta última etapa se hizo un planteamiento de los posibles eventos que en algún momento podrían alterar el buen funcionamiento del sistema entre las computadoras y el servidor central, tal como el rompimiento de un enlace a mitad de una transmisión, la desconexión accidental de una PC o la interrupción de uno de los equipos que dan servicio en Red-UNAM, entre otros.

Para dar solución a lo anterior, se generó una rutina que manejara los posibles errores de comunicación, a fin de que, por una parte, la máquina no se le quedara trabada al usuario, y por otra, de que en caso de que se restableciera la

comunicación, el programa continuara con su ejecución normal. A esta rutina de errores también se le integró programación referente a la impresión y al almacenamiento de datos en disco duro o flexible.

Las áreas del Patronato que participaron en la revisión del sistema fueron: el Departamento de Procesamiento de Datos (por ser el generador de reportes en el esquema de servicio anterior), el área de Contaduría General y el Departamento de Control Presupuestal (por ser los principales usuarios de esta información dentro de toda la Universidad).

Durante el periodo de pruebas realizado por dichas áreas, el Departamento de Procesamiento de Datos detectó varios errores principalmente en lo referente a la generación de los reportes de Previos, Gastos a Reserva de Comprobar, Estados de Cuenta Presupuestales y de Ingresos Extraordinarios. Dichos errores, fueron solucionados de tal manera que los usuarios pudieran imprimir en su computadora una copia exacta de los reportes que recibían bajo el esquema de trabajo anterior, además de tener la ventaja de imprimir por criterios generales o por consultas muy específicas.

Las modificaciones solicitadas por Contaduría y Control Presupuestal se debieron básicamente a nuevos requerimientos de información, generación de criterios de búsqueda adicionales y a la modificación del orden en que se presenta la información en pantalla, todos los cuales fueron atendidos y solucionados oportunamente.

#### 4.4 Implantación

En esta etapa del desarrollo del sistema, el analista ayuda a implantar el sistema de información para el manejo directo del usuario. Esto incluye el adiestramiento que el usuario requerirá para conocer a detalle el funcionamiento del sistema.

La estrategia utilizada para la implantación del SIAU, estuvo basada en el *Enfoque Piloto*, mismo que se utiliza cuando los sistemas nuevos implican cambios drásticos en el desempeño de una empresa. De tal manera que, en una primera etapa, se eligieron al azar cinco dependencias que representan cuatro de los diferentes sectores de la universidad; estas dependencias fueron:

Sector Central: la Secretaría Administrativa y la Dirección General de Programación y Presupuesto.

Sector Administrativo: la Dirección General de Incorporación y Revalidación de Estudios.

Sector Educativo: Facultad de Medicina Veterinaria y Zootecnia.

Sector de Investigación: Instituto de Biología.

En esta etapa se trabajó directamente con los titulares y representantes de cada una de las dependencias a fin de explicarles el funcionamiento del sistema y apoyarles durante sus primeros intentos de manejo y operación de cada módulo. En esta primera etapa, se obtuvieron resultados favorables, a partir de los cuales se planearon las siguientes etapas.

Posteriormente, se realizaron sesiones inductivas a las que se invitó a todos los directores y secretarios administrativos de todas las dependencias de la UNAM. En

cada sesión, además de hacer una presentación del Sistema, se brindó una demostración de su funcionamiento. Al término de cada presentación a cada participante se le entregó el manual de instalación y una clave de usuario para que al llegar a su dependencia pudiera instalar y comenzar a hacer uso del sistema.

#### **4.4.1 Proceso de Instalación**

Los requerimientos de hardware y software que deben cumplir los equipos de cómputo de los usuarios para estar en condiciones de explotar el sistema, son los siguientes:

##### **Hardware**

- PC con procesador 486 o Pentium, con un mínimo de 8MB en RAM (se recomendaban 16 MB) y monitor VGA o superior.
- Una impresora, preferentemente con tecnología láser.
- Infraestructura para la conexión a Red-UNAM en la dependencia o enlace vía módem.

##### **Software**

- Sistema Operativo Windows 3.11 o superior, se recomienda Windows 95/98.
- Protocolo TCP/IP.

Una vez que se cuenta con lo anterior, se debe realizar la transferencia de archivos vía el programa FTP, copiando del servidor de Patronato los archivos necesarios hacia la estación de cada usuario, tal como lo ilustra la Figura 4.27.

## Proceso de Instalación del SIAU

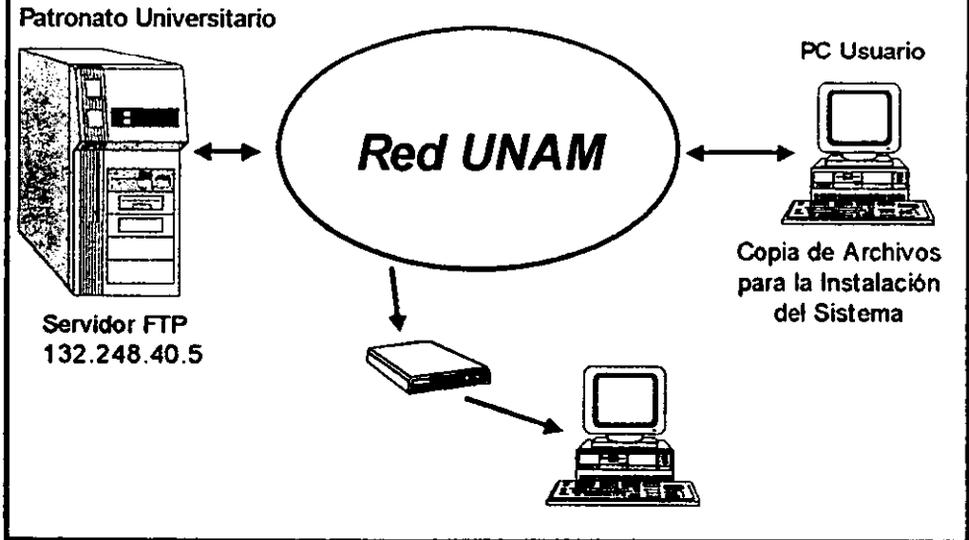


Figura 4.27 Proceso de Instalación del SIAU

El manual de instalación fue ideado de tal manera, que incluso los usuarios que no tuvieran experiencia en cómputo pudieran realizar la instalación rápida y sencillamente. Las instrucciones para la instalación son las siguientes:

- Cree un directorio en raíz (*C:\*) de su disco duro con el nombre *INSTALA*.
- Invoque el programa *ftp.exe* (dar doble click al icono de MS-DOS estando en *C:\* escribimos *ftp.exe*).
- Estando en *ftp>*, ubíquese en el directorio *INSTALA* con el comando:  
*lcd c:\instala.*

- Inicie la conexión al servidor mediante la instrucción **open 132.248.40.5**
- Introduzca la clave de usuario: **siau** y el password: **ftpsiau**
- Teclee **binary** para una transferencia adecuada
- Inicie la copia con el comando **mget \*.\***
- Por cada archivo a copiar, presione la tecla **enter**
- Salga de la sesión ftp con el comando **bye**
- Invoque el programa **C:\INSTALA\INSTALAR.EXE** y siga las instrucciones que aparecerán en la pantalla.

Una vez que se ejecuta el programa **INSTALAR.EXE** sobre la computadora cliente, se solicita al usuario su clave de acceso (previamente asignada), su contraseña, la dependencia a la que pertenece y algunos datos personales. Posteriormente, se genera un subdirectorío llamado **SIAU** en donde se copian algunos archivos que sirven para el correcto funcionamiento del sistema. Más adelante se realizan las modificaciones necesarias a los archivos *win.ini* y *odbc.ini*, para finalmente generar un grupo de programas denominado **SIAU**.

El sistema se puso a disposición de la comunidad universitaria a partir del mes de enero de 1997. Como era de esperarse, en un principio fueron pocas las dependencias que lo utilizaron, debido principalmente a dos factores: el primero, de tipo personal, consistió en una fuerte resistencia al cambio por parte de los usuarios; por su parte, el segundo, de tipo estructural, se derivó de la barrera de comunicación derivada de que en ese entonces sólo el 85% de las dependencias contaba con conexión a Red-UNAM. Con el paso del tiempo y gracias a una

apertura de los usuarios y a que la red universitaria se ha ido expandiendo (o bien, se han multiplicado las conexiones vía Módem), paulatinamente se han integrando cada vez más dependencias al uso cotidiano del sistema.

#### 4.4.2 Tipos de Ayuda

El sistema ofrece diferentes tipos de ayuda, la primera de ellas está incluida dentro del mismo programa, es decir, bastará con que el usuario oprima la tecla *FI* para que inmediatamente le aparezca una ventana de ayuda donde podrá consultar cualquier tema referente al SIAU, tal y como lo muestra la siguiente figura:

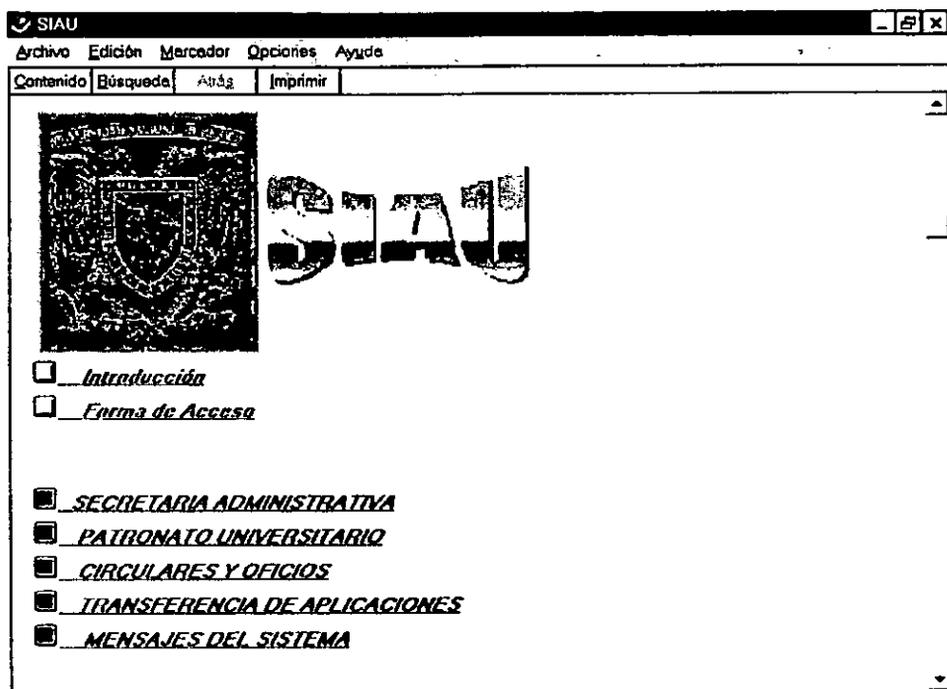


Figura 4.28 Ayuda en Línea del SIAU

Además de la ayuda general, también se cuenta con una ventana de ayuda dentro de cada uno de los módulos del sistema, por ejemplo, para el caso de que un usuario tenga duda acerca del número de grupo de gasto que desea consultar, al teclear *F1* dentro de la casilla de grupo, aparecerá la pantalla que se muestra a continuación (Figura 4.29).

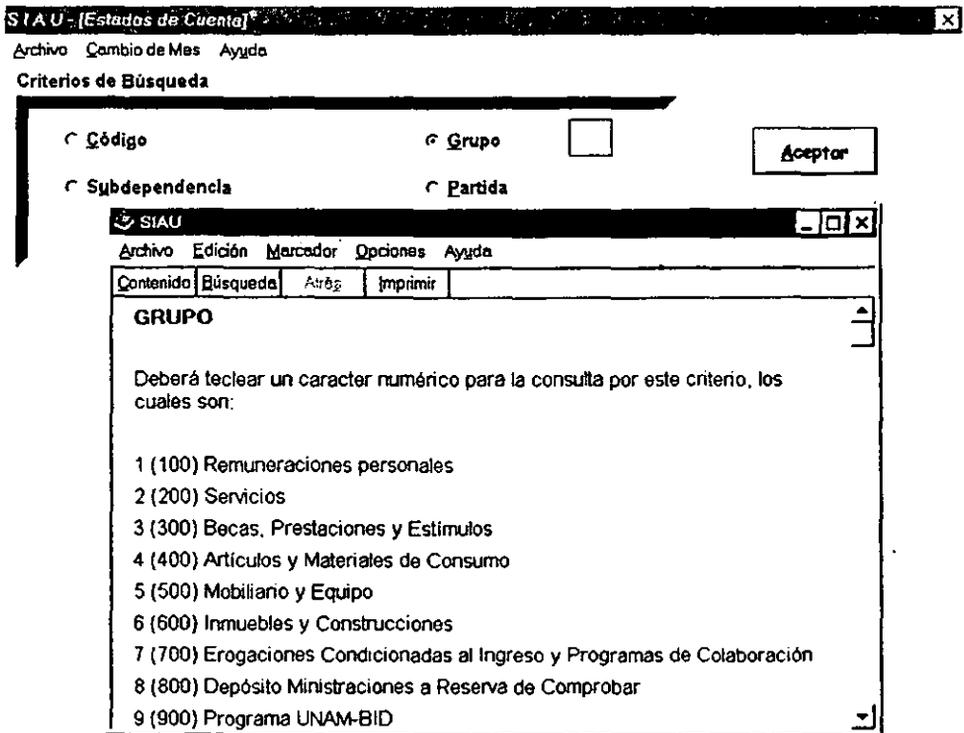


Figura 4.29 Ayuda de un Tema en Particular

Adicionalmente, los usuarios tienen la posibilidad de consultar el *Manual de Usuario* y navegar a través de las diferentes opciones que ofrece el sistema, a través de la página web del SIAU, cuya dirección es: <http://www.patronato.unam.mx/siau>.

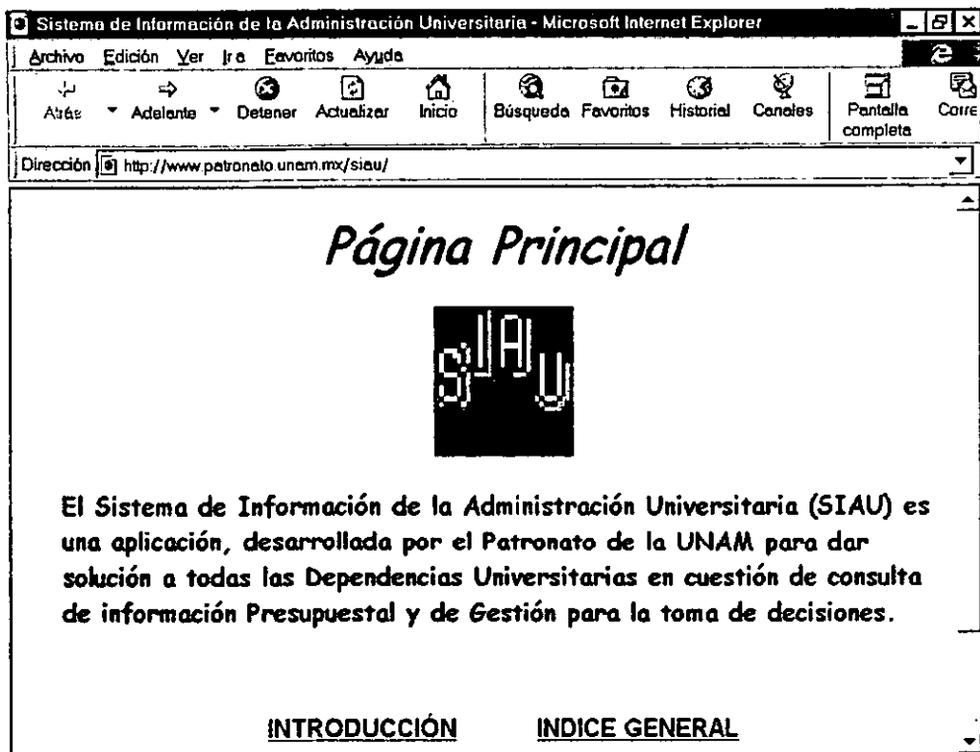


Figura 4.30 Página Web del SIAU

Una modalidad más de ayuda se da a través de una cuenta de correo electrónico conocida como [siau@patronato.unam.mx](mailto:siau@patronato.unam.mx) en donde los usuarios pueden enviar correos para reportar problemas con el sistema, solicitar nuevas claves de usuario o bien, proponer mejoras en la aplicación.

La última alternativa, que hasta el momento ha presentado mayor frecuencia de uso, es la asistencia vía telefónica, servicio para el que Patronato Universitario puso 3 líneas telefónicas a disposición de todos los usuarios, a fin de brindar atención expedita a las solicitudes y requerimientos manifestados por los usuarios.

### 4.4.3 Monitoreo del Sistema

Existen diversas herramientas para monitorear el comportamiento del sistema, algunas de ellas pueden ser adquiridas por terceros, otras vienen incluidas dentro del sistema operativo de red o del manejador de base de datos y unas más que pueden ser diseñadas por el administrador del sistema.

La figura 4.31 muestra la funcionalidad de una herramienta denominada Windows NT Monitor que viene integrada dentro del mismo sistema operativo de red. A través de ella se pueden monitorear elementos tan importantes como el uso del procesador, la memoria, el número de usuarios conectados simultáneamente, el número de transacciones por segundo, el tamaño del *Log* de la base de datos y el número de bytes enviados/recibidos por segundo, entre otros.

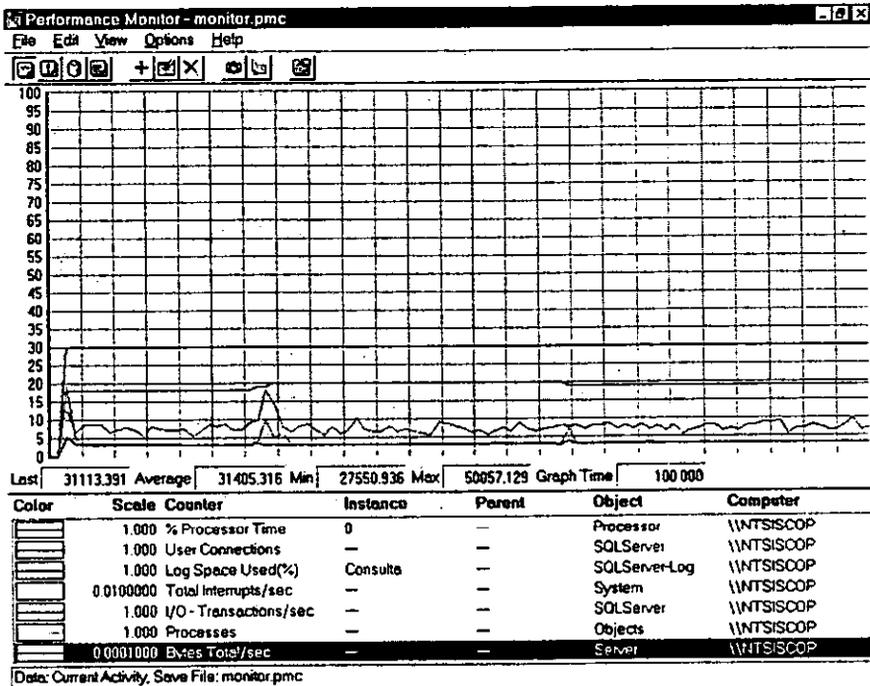


Figura 4.31 Monitoreo del Sistema a través de Windows NT Monitor

Esta herramienta es muy importante, ya que a través de ella se puede saber cuándo es necesario agregar más procesadores, discos duros o memoria al servidor central, a fin de mantener el mejor tiempo de respuesta para los usuarios.

Otra herramienta con la que se puede monitorear el comportamiento de todos y cada uno de los usuarios dentro del sistema de base de datos es el SQL Trace (ver Figura 4.32) misma que viene incluida dentro del software de SQL Server.

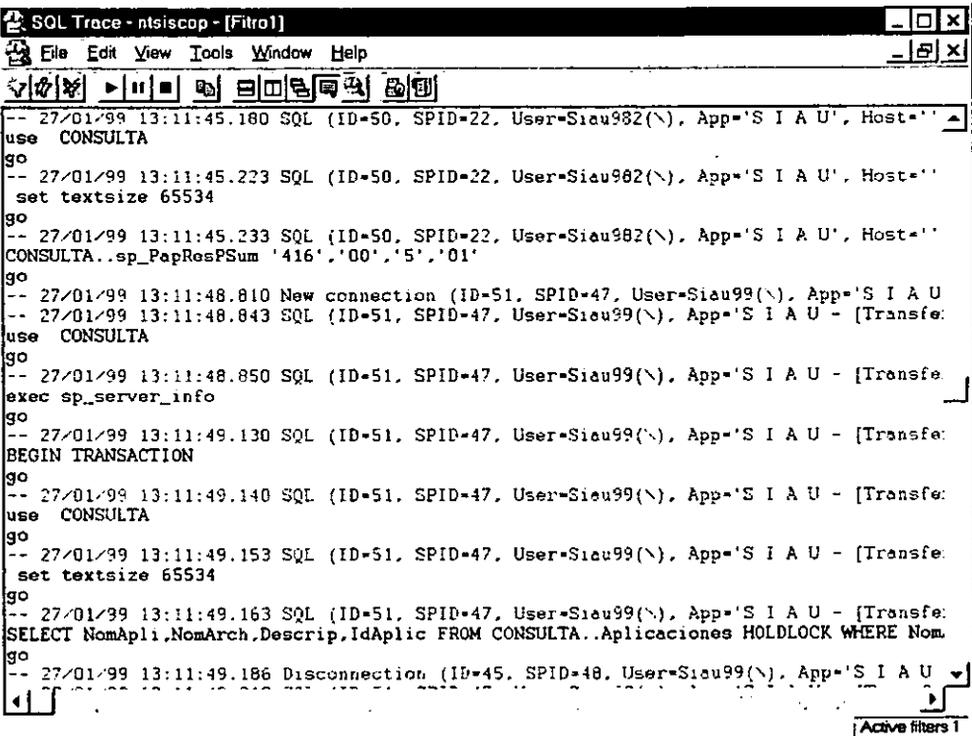


Figura 4.32 Monitoreo del Sistema a través de SQL Trace

Con esta herramienta se pueden visualizar a detalle todas las consultas a tablas, todos los llamados a procedimientos almacenados y, en general, todo el comportamiento que tiene cada usuario, desde el momento en que entra hasta que sale del sistema. Con esta información se puede rastrear, por ejemplo, el usuario que imprimió un reporte de Presupuesto, a qué hora, con qué clave y desde qué computadora.

-Claves-					
Archivo Programas					
<b>Control de Claves de Usuario</b>					
Nombre:	RUBEN	OLMEDO	PONCE	Ejecutar	
Clave:	AALCAZAR	Contraseña:	888888	Tipo:	G Nuevo
Puesto:	Jefe Unidad Administrativa				Actualizar
Estado Actual:	<input type="radio"/> Activo	<input checked="" type="radio"/> Inactivo	Tel:	6221361	Sist. Op. Windows 95
Dep. y SubDep: (Alcance)	727	01	DIRECCION GENERAL DE SERVICIOS INSTITUCIONALES		
Entrada:	25/jan/99 14:14		Salida:	25/jan/99 14:15	
No. de Accesos:	100		Tiempo de Uso:	975	
Cve. Dep: (Pertenece a:)	727	Cve. SDep:	00	Mensaje:	F
No. de Usuarios:	533	Total de Accesos:	130903	Dir. IP:	132.248.171.83
		Accesos de Hoy:	189	Versión:	siau98
					Salir

Figura 4.33 Monitoreo del Sistema a través de CLAVES.EXE

Por último, el sistema cuenta con una herramienta desarrollada por el administrador del sistema, en lenguaje Visual Basic denominada CLAVES.EXE (Figura 4.33) a través de la cual se pueden realizar altas, bajas y cambios de todos los usuarios del sistema.

Como se explicó en el punto 4.2.5, dentro de la base de datos del sistema existe una tabla que maneja toda la información referente a un usuario, tal como: nombre, clave, contraseña, tipo, puesto, teléfono, versión de Windows utilizada, un campo para saber si está dentro del sistema, clave de la dependencia y subdependencia a la que pertenecen así como clave de la dependencia y subdependencia a la que tienen alcance, fecha y hora de su última entrada y salida del sistema, el número de accesos que ha realizado, el tiempo de uso en minutos, la dirección IP de su computadora personal, un campo para saber si ha leído el último mensaje que recibió y la versión del sistema que está utilizando.

Un par de datos importantes que muestra el sistema, son los referentes al número total de usuarios y a la suma total de accesos al sistema que han tenido todos los usuarios desde su puesta en operación el pasado mes de enero de 1997.

Así mismo, a través del módulo de CLAVES, se envían los mensajes del sistema a los usuarios, y se les da mantenimiento a los catálogos de dependencias, aplicaciones, directorio telefónico y oficios.

## **4.5 Mantenimiento**

La fase de mantenimiento se centra en el cambio del Sistema que va asociado a la corrección de errores sobre las aplicaciones requeridas a medida que evoluciona el entorno del software, y a los cambios derivados de las mejoras producidas por los requisitos variables del cliente. La fase de mantenimiento vuelve a aplicar los pasos de las fases de análisis diseño y desarrollo, pero en el contexto del software ya existente.

Durante la fase de mantenimiento se encuentran cuatro tipos de cambios, de los cuales se hablará continuación.

### **Mantenimiento Correctivo**

Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el usuario descubra defectos en el software. El mantenimiento correctivo tiene por función modificar el software a fin de corregir los defectos detectados.

En el caso del SIAU, desde su puesta en marcha, los errores reportados por el usuario fueron corregidos en forma inmediata. Es así que los usuarios reportaron la imposibilidad de imprimir en impresoras conectadas a una red local; también se indicó que el sistema sólo podía respaldar información en disco duro (C:\>) y no directamente en un disquete (A:\>). Ambos problemas fueron corregidos para posteriormente entregar una versión actualizada a todos los usuarios.

### **Mantenimiento Preventivo**

El software de computadora se deteriora debido al cambio, y por esto, el mantenimiento preventivo, también llamado "reingeniería del software", se

debe orientar a la solución de las necesidades de los usuarios finales. En esencia, el mantenimiento preventivo hace cambios en programas de computadora a fin de que puedan corregir, adaptar y mejorar el sistema en beneficio del usuario.

Debido a las características propias del SIAU (únicamente consulta de información), este tipo de mantenimiento no ha tenido aplicación, ya que desde un inicio se tuvo la definición específica de la función de cada módulo dentro del sistema.

### **Mantenimiento Adaptativo**

Con el paso del tiempo, es posible que cambie el entorno para el que originalmente se desarrolló el software (por ejemplo, las actualizaciones en el CPU, la versión del sistema operativo, las políticas de la empresa, las características externas de los productos, etc.). El mantenimiento adaptativo produce modificación en el software a fin de ajustarlo a los cambios de su entorno externo.

Sin duda alguna, este tipo de mantenimiento es el que más se ha aplicado dentro del SIAU, ya que en su versión original (conocida como SIC), sólo se contaba con 12 módulos, mientras que a los seis meses de su puesta en marcha creció a 16 módulos. Para enero de 1998 ya se habían desarrollado 25 módulos; en el mes de agosto del mismo año nace la primera versión del SIAU con modificaciones sobre los 25 módulos de su sistema predecesor, más el desarrollo de 5 módulos adicionales, de los cuales, 4 se enfocan a la Secretaría Administrativa y el último a Patronato Universitario.

Debido a la aceptación que ha tenido el sistema dentro de la comunidad universitaria, es muy posible que el SIAU siga creciendo para ofrecer más opciones

de consulta de información generada en otras dependencias y no solamente en Patronato y en la Secretaría Administrativa de la UNAM.

### **Mantenimiento Perfectivo**

Conforme se utilice el software, el cliente/usuario puede descubrir funciones adicionales que podrían ser de gran beneficio. El mantenimiento perfectivo lleva al software más allá de sus requisitos funcionales originales.

Al paso del tiempo, se han detectado algunas adecuaciones de tipo “perfectivo”, que se pueden realizar sobre el sistema, entre las que se encuentra la posibilidad de enviar correo electrónico desde el mismo SIAU hacia el administrador del sistema con el objeto de reportar fallas, problemas o sugerencias para el mejoramiento de la aplicación. Asimismo, también se ha planeado la posibilidad de entrar a páginas Web de la Universidad con el simple hecho de oprimir un botón desde el SIAU.

Una ventaja que representa la arquitectura cliente/servidor, es el hecho de que cuando cambian las reglas de negocio, sólo se tiene que modificar la programación realizada sobre los procedimientos almacenados ubicados en el servidor de base de datos, de tal manera que los cambios aplican inmediatamente a todos los usuarios del sistema.

Sin embargo, es importante resaltar que, cuando se realiza una modificación sobre el programa fuente (SIAU.EXE), se tiene que distribuir la versión modificada a todos los usuarios, lo que podría representar largos periodos para que todos ellos cuenten con la misma versión, sobre todo cuando hablamos de un universo de más de 500 PC's que utilizan el sistema.

Para resolver este problema, se crearon dos estrategias: la primera de ellas es que, a

partir de la versión de agosto de 1998, el programa cuenta con una opción llamada *Transferencia de Aplicaciones*, donde el usuario puede seleccionar de la lista la última versión del SIAU y siguiendo un sencillo procedimiento puede bajarla al disco duro de su PC. La otra opción consiste en una página Web creada especialmente para el SIAU, en donde el usuario puede consultar el manual de instalación cuantas veces sea necesario.

## CONCLUSIONES

El desarrollo acelerado en los últimos años en materia de comunicaciones, hardware y software, ha permitido generar sistemas de información bajo una perspectiva totalmente diferente a como estos se realizaban en la década de los 70's.

Así mismo, el hecho de que la Universidad Nacional Autónoma de México cuente con un instrumento de comunicación tan eficiente como Red-UNAM, ha propiciado que tanto escuelas, facultades e institutos, como dependencias administrativas puedan compartir e intercambiar más ágilmente información de gran importancia, en beneficio de alumnos, investigadores, profesores y autoridades académicas y administrativas, para un mejor funcionamiento de la máxima casa de estudios del país.

El Sistema de Información de la Administración Universitaria, se convirtió en la primera aplicación dentro de la Universidad que no solamente utiliza lo más avanzado en la tecnología de hardware y software, sino que también hizo uso efectivo de Red-UNAM, es decir, utilizó la infraestructura de la red para compartir información a altas velocidades, sin limitarse al envío de correo electrónico o a la navegación por páginas Web dentro de Internet.

La implantación de este sistema, puso fin a más de 20 años de impresión de una cantidad superior a 20 mil hojas de reportes que eran entregados al fin del ejercicio mensual. Adicionalmente, brindó a las dependencias la posibilidad de obtener información de los movimientos realizados con actualización al día anterior, lo que favoreció la disminución drástica de

las operaciones de conciliación realizadas con la Contaduría General con el correspondiente ahorro de recursos humanos y materiales.

Debido a la gran aceptación que tuvo la aplicación dentro de la comunidad universitaria durante su primer año de operación, el Sistema Integral de Consulta (SIC) antes exclusivo del Patronato Universitario, amplió su alcance adquiriendo el nombre de Sistema de Información de la Administración Universitaria (SIAU), ofreciendo no sólo información de Patronato, sino también la correspondiente a algunas dependencias pertenecientes a la Secretaría Administrativa de la UNAM, dejando abierta la posibilidad de que a futuro se sigan integrando cada vez más dependencias que manejen información de interés para la comunidad universitaria.

Actualmente, la base de datos del sistema se encuentra instalada en un servidor Digital, con sistema operativo Windows NT y SQL Server como manejador de bases de datos. Dado que el sistema fue desarrollado bajo arquitectura cliente/servidor, brinda la posibilidad de una conexión directa con otras tecnologías, es decir, a través del uso de la conectividad abierta para bases de datos (ODBC), el sistema tiene la capacidad de consultar bases que operan con otras plataformas, (por ejemplo: aquellas instaladas sobre equipos SUN, con sistema operativo Unix y manejador de base de datos Sybase), mientras que la aplicación cliente sigue siendo prácticamente la misma, lo que permite expandir las posibilidades de consulta de información.

El uso y explotación que el sistema planteó de las librerías dinámicas DLL's (creadas por terceros), favoreció considerablemente la reducción

del código del programa; permitió aumentar su potencialidad y apoyó la creación de una aplicación muy robusta en menos tiempo y con pocos recursos. Asimismo, gracias al diseño modular con el que cuenta, el sistema está preparado para integrar en breve nuevas opciones de consulta y operación.

A la fecha de finalización del presente trabajo, el SIAU se encuentra instalado en más de 500 PC's pertenecientes a las diferentes dependencias universitarias, distribuidas a lo largo y ancho de todo el país, constituyéndose como una herramienta de gran valor para el trabajo diario de las áreas administrativas así como para la toma de decisiones de los sectores directivos y ejecutivos de la Institución.

A partir de su puesta en operación (enero de 1997) el número de usuarios ha ido en aumento; actualmente, se tienen 533 dados de alta, lo que ha representado más de 130 mil accesos, alcanzando un promedio diario de 500. Considerando estos datos, se puede inferir que el sistema está siendo utilizado por una gran cantidad de áreas de la comunidad universitaria como herramienta básica para la realización de su quehacer cotidiano. En fechas próximas se tiene planeado integrar a todos los investigadores de los Proyectos PAPIIT, lo que representará la incorporación de más de 700 usuarios adicionales.

A pesar de que el funcionamiento del sistema mantiene un buen nivel, es recomendable promover su migración hacia la versión 6.0 del lenguaje Visual Basic, a fin de aprovechar los nuevos alcances logrados por las tecnologías recientes, con lo cual la aplicación tendrá un mejor rendimiento, consumirá menos recursos y podrá sacar ventaja de las

características de Internet y de los controles que la nueva versión de Visual Basic ofrece.

La experiencia obtenida a partir de la implantación del SIAU, permite suponer que en la medida en que se generen y utilicen sistemas similares dentro de la Institución, la Universidad contará con mejores herramientas de trabajo que apoyen sus funciones sustantivas: docencia, investigación y extensión cultural, cumpliendo así con la misión de preparar profesionistas que puedan servir a la sociedad y enfrentar de la mejor manera los retos del siguiente milenio.

## BIBLIOGRAFÍA

Universidad Nacional Autónoma de México  
Patronato Universitario, Guía de Introducción  
UNAM, México, Primera Edición, 1998.

Dirección de Telecomunicaciones de la UNAM  
<http://www.dtd.unam.mx/REDUNAM>

Tom Sheldon  
Aplique Windows NT  
Osborne/MacGraw-Hill, España, Primera Edición, 1994.

James R. Groff y Paul N. Weinerg  
Aplique SQL  
Osborne/MacGraw-Hill, España, Primera Edición, 1992.

Harjinder S. Gill y Prakash C. Rao  
Data Warehousing  
Prentice Hall Hispanoamericana, México, Primera Edición, 1996.

Roger S. Pressman  
Ingeniería del Software  
MacGraw-Hill/Interamericana, España, Tercera Edición, 1993.

**Kendall y Kendall**

**Análisis y Diseño de Sistemas**

**Prentice Hall Hispanoamericana, México, Primera Edición, 1991.**

**Microsoft SQL Server Transact-SQL Reference 6.5**

**Microsoft Corporation, USA, 1997.**

**Microsoft SQL Server System Administrator's Companion 6.5**

**Microsoft Corporation, USA, 1997.**

**Programming in Microsoft Visual Basic 3.0 Workbook**

**Microsoft Corporation, USA, 1994.**

**William R. Vaughn**

**Hitchhiker's Guide to Visual Basic & SQL Server**

**Microsoft Press, USA, Fifth Edition, 1997**

**Q+E Multilink/VB 2.0 User's Guide**

**Q+E Software Inc., USA, 1994**

## INDICE DE ILUSTRACIONES

Figura 1.1	Organigrama General de la Universidad Nacional Autónoma de México.....	5
Figura 1.2	Estructura interna del Patronato Universitario.....	7
Figura 1.3	Proceso de consolidación de información en la Dirección General de Control e Informática.....	12
Figura 2.1	Red Integral de Telecomunicaciones de la Universidad Nacional Autónoma de México.....	25
Figura 2.2	Plataforma de equipo seleccionada para el proyecto SIAU.....	59
Figura 3.1	Procesos de un sistema Data Warehousing.....	70
Figura 3.2	Proceso de Datawarehouse para el SIAU.....	76
Figura 3.3	Componentes de la arquitectura ODBC.....	83
Figura 3.4	Herramientas adicionales al lenguaje Visual Basic.....	86
Figura 4.1	Propuesta de Solución.....	99
Figura 4.2	Diferentes Niveles de Acceso a la información.....	102
Figura 4.3	Modelo Seleccionado para el Desarrollo del Sistema.....	107
Figura 4.4	Forma Básica de un Diagrama de Flujo de Datos.....	110
Figura 4.5	Notación Básica de un Diagrama de Flujo de Datos.....	111
Figura 4.6	Diagrama de Contexto, Nivel 0 del S.I.A.U.....	112
Figura 4.7	Módulo de Carga de Información.....	113
Figura 4.8	Módulo de Acceso al Sistema.....	114
Figura 4.9	Módulo Principal del Sistema.....	115
Figura 4.10	Módulo de Patronato Universitario.....	117
Figura 4.11	Módulo de Ejercicio Presupuestal.....	122
Figura 4.12	Diagrama Entidad-Relación (Primera Parte).....	125
Figura 4.13	Diagrama Entidad-Relación (Segunda Parte).....	126
Figura 4.14	Diagrama Entidad-Relación (Tercera Parte).....	127
Figura 4.14	Enfoque de Diseño Descendente (top-down).....	152
Figura 4.15	Creación de los Devices de Almacenamiento.....	165
Figura 4.16	Creación de la Base de Datos.....	166
Figura 4.17	Creación de las Tablas.....	167
Figura 4.18	Creación de los Indices.....	169

Figura 4.19	Programa Orientado por Menús.....	194
Figura 4.20	Uso de Menús Anidados.....	195
Figura 4.21	Ejemplo de un Diagrama Estructural.....	197
Figura 4.22	Diagrama Estructural del Sistema de Información de la Administración Universitaria .....	198
Figura 4.23	Diagrama Estructural del Módulo Ejercido Presupuestal.....	199
Figura 4.24	Diseño de la Forma de Ejercido Presupuestal .....	200
Figura 4.25	Propiedades del botón Aceptar .....	201
Figura 4.26	Forma de Ejercido Presupuestal configurada .....	203
Figura 4.27	Proceso de Instalación del SIAU .....	214
Figura 4.28	Ayuda en Línea del SIAU.....	216
Figura 4.29	Ayuda de un Tema en Particular .....	217
Figura 4.30	Página Web del SIAU.....	218
Figura 4.31	Monitoreo del Sistema a través de Windows NT Monitor.....	219
Figura 4.32	Monitoreo del Sistema a través de SQL Trace .....	220
Figura 4.33	Monitoreo del Sistema a través de CLAVES.EXE .....	221