



UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA INTEGRAL PARA EL CONTROL Y
REGISTRO DE FACTURAS PARA EL
DEPARTAMENTO DE ADQUISICIONES DE
LA DIRECCIÓN GENERAL DE
BIBLIOTECAS DE LA UNAM

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A:
SILVIA SOCORRO BALLESTEROS
ESTRADA



DIRECTOR DE TESIS: M. I. MARCIAL CONTRERAS
BARRERA

MÉXICO, D.F.

OCTUBRE 2003



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Índice

Índice de Figuras	VI
Índice de Tablas	VII
Índice de Diagramas	VIII
Introducción	IX
1. Definición del Sistema	1
1.1. Definición del problema	1
1.2. Objetivos	1
1.3. Diagnóstico de la problemática actual	2
1.4. Análisis de factibilidad	2
1.4.1. Factibilidad técnica	2
1.4.2. Factibilidad económica	3
1.4.3. Factibilidad legal	3
1.4.4. Factibilidad operacional	3
1.4.5. Factibilidad de programa	3
1.5. Beneficios del sistema	4
1.5.1. Diseño del sistema	4
1.5.2. Interfaz Gráfica	4
1.5.3. Informes	4
1.5.4. Procesos Manuales	4
1.5.5. Base de Datos relacional	4
1.5.6. Diseño Distribuido	5
1.6. Metas para el sistema	5
1.6.1. Metas para el Proceso	5
1.6.1.1. Cualitativas.	5
1.6.1.2. Cuantitativas.	5
1.6.2. Metas para el Producto Final	5
1.6.2.1. Cualitativas	5
1.6.2.2. Cuantitativas	5
1.7. Requerimientos del Sistema	5
1.7.1. Confiabilidad	5
1.7.2. Disponibilidad	5
1.7.3. Flexibilidad	6
1.7.4. Crecimiento	6
1.7.5. Mantenimiento	6
2. Antecedentes	7
2.1. Nociones de Ingeniería de Software	7
2.1.1. Objetivos de ingeniería del software	7
2.1.2. características de los productos de software	7

2.1.3. Paradigmas de ingeniería del Software.	8
2.1.3.1. Modelos Genéricos	8
2.1.3.2. Modelos Híbridos	8
2.1.3.3. Fases del modelo en cascada	8
2.1.3.3.1. Documentos del modelo en cascada	9
2.1.3.4. Modelo Evolutivo	9
2.1.3.4.1. Problemas	9
2.1.3.4.2. Aplicabilidad	9
2.1.3.5. Modelo Prototipado	10
2.1.3.6. Modelo en Espiral Boehm	10
2.1.3.6.1. Fases del modelo en espiral	11
2.1.3.6.2. Desventajas del modelo en espiral	11
2.1.4. Tabla comparativa de los Paradigmas de Ingeniería del Software.	11
2.1.4.1. Técnicas de Cuarta Generación	11
2.1.5. Metodología CASE	12
2.1.5.1. Herramientas CASE	12
2.2. Nociones de Bases de datos	13
2.2.1. Base de Datos	13
2.2.2. Entidades	13
2.2.3. Atributo	13
2.2.4. Valor de los Datos	14
2.2.5. Claves o llaves	14
2.2.6. Tipos de Relaciones	14
2.2.6.1. Relación Uno a Uno	14
2.2.6.2. Relación Uno a Muchos	14
2.2.6.3. Relación Muchos a Muchos	14
2.2.7. Normalización	15
2.2.7.1. Formas Normales	15
2.2.8. Objetivos de la Organización de la Base de Datos	16
2.2.8.1. Desempeño	16
2.2.8.2. Redundancia mínima	16
2.2.8.3. Capacidad de búsqueda	16
2.2.8.4. Integridad	16
2.2.8.5. Migración de datos	16
2.2.9. Pasos en el Diseño de una Base de Datos	17
2.2.10. Modelo de Datos	18
2.2.10.1. Modelo Conceptual	18
2.2.10.2. Modelo Lógico	18
2.2.10.3. Modelo Físico	18
2.2.11. Nociones de Manejadores de Bases de Datos	19
2.2.11.1. Oracle	19
2.2.11.2. PostgreSQL	19
2.2.11.3. ASAnyWhere	20
2.2.11.4. Microsoft SQL Server	20
2.3. Nociones de Herramientas de Desarrollo	21
2.3.1. Parámetros a considerar en las herramientas de desarrollo	21
2.3.2. Herramientas de Desarrollo	21
2.3.2.1. Visual Basic	21

2.3.2.2. Delphi	21
2.3.2.3. Power Builder	22
2.4. Nociones de Sistemas Operativos	22
2.4.1. Herramientas	22
2.4.2. características	22
2.4.3. Sistemas Operativos	23
2.4.3.1. Solaris	23
2.4.3.2. Linux	23
2.4.3.3. Windows 2000	23
3. Análisis del sistema	24
3.1. Alcance del sistema	25
3.2. Recolección de requerimientos	26
3.2.1. Áreas involucradas	26
3.2.1.1. Facturación	26
3.2.1.2. Transferencias	27
3.2.1.3. Sice	28
3.2.1.4. Auditoria	28
3.2.1.5. Proveedores	29
3.3. Resultados de las necesidades	30
3.4. Recursos de Software y Hardware disponibles	31
4. Análisis estructurado	33
4.1. Diagrama de flujo de datos	33
4.1.1. Componentes de un DFD	33
4.2. Diagramas entidad – relación	35
4.2.1. Componentes del diagrama Entidad – Relación	35
4.2.1.1. Tipos de Objetos	35
4.2.1.2. Relaciones	35
4.3. Diccionario de datos	36
4.4. Arquitectura cliente – servidor	36
4.4.1. Dos Capas	36
4.4.2. Tres capas	38
4.5. Funciones del servidor y funciones del cliente	39
4.5.1. Servidor	39
4.5.2. Cliente	39
4.6. Estimación de Costos	39
4.6.1. Técnicas de Estimación de Costos	40
4.6.1.1. Juicio Experto	40
4.6.1.2. Estructuras de División de Trabajo	40
4.6.1.3. Estimación del Nivel de Contratación	40
4.6.1.4. Estimación de Costos de Mantenimiento	40
4.7. Herramientas de Control de Avance	40
4.7.1. Método de la Ruta Crítica (MRC)	41
4.7.2. Programme evaluation and review technique (PERT)	41
4.7.3. Diagrama de GANTT	42

5.	Desarrollo	44
5.1.	Diseño	44
5.1.1.	Diagrama Jerárquico Funcional	44
5.1.1.1.	Convenciones para dibujar un diagrama Jerárquico Funcional	44
5.1.2.	Diagrama de Flujo de Datos	47
5.1.2.1.	Diagrama de Flujo de Datos de Nivel 0 o Conceptual.	47
5.1.2.2.	Diagrama de Flujo de Datos de Nivel 1 o Detalle.	47
5.1.3.	Diagrama Entidad – Relación	48
5.1.4.	Diseño de la Base de Datos	49
5.1.4.1.	Tablas de la Base de Datos	50
5.2.	Manejadores de Bases de Datos	55
5.2.1.	Diseño de la Base de Datos	55
5.2.1.1.	Selección y justificación de las herramientas de modelado	55
5.2.1.2.	Selección y justificación del manejador de Bases de Datos	55
5.3.	Diseño de la Aplicación	55
5.3.1.1.	Selección y justificación de las herramientas para generar la interfaz del usuario	55
5.3.1.2.	Interfaz del usuario	56
5.3.1.3.	Módulos	57
5.3.1.3.1.	Facturas	57
5.3.1.3.2.	Transferencias	58
5.3.1.3.2.1.	Archivos	58
5.3.1.3.2.2.	Facturas Por Archivos	59
5.3.1.3.3.	Sice	60
5.3.1.3.3.1.	Catálogo de Códigos Programáticos	60
5.3.1.3.4.	Proveedores	60
5.3.1.3.4.1.	Precios	61
5.3.1.3.5.	Auditorias	61
5.3.1.3.5.1.	Captura de Títulos	62
5.3.1.3.5.2.	Comprobación de Títulos	62
5.3.1.4.	Catálogos	63
5.3.1.5.	Informes	63
5.3.1.6.	Ayuda	65
5.3.2.	Pantallas	65
5.3.3.	Informes	67
5.3.3.1.	Selección y justificación de la herramienta para generación de informes	67
5.3.4.	Ayuda	68
6.	Liberación	69
6.1.	Revisiones Técnicas Formales	69
6.2.	Pruebas del Sistema	71
6.2.1.	Módulo	71
6.2.2.	Integral	71
6.2.3.	General	71
6.3.	Manual del usuario	71
6.4.	Manual de Técnico	72
6.5.	Capacitación	72

6.6. Mantenimiento	72
6.7. Seguridad	74
6.7.1. SQL Server	74
6.7.2. Windows 2k	74
Conclusiones	76
Apéndice A	78
Apéndice B	84
Apéndice C	97
Bibliografía	104

Introducción

La Ingeniería de Software es un enfoque sistemático de desarrollo, operación y mantenimiento de software, la Ingeniería de Software es la rama de la ingeniería que aplica los principios computacionales y las matemáticas para lograr soluciones económicas a los problemas de desarrollo de software.

El diseño de software debe brindar servicios informáticos de alta calidad para la optimización de trabajo en cualquier tipo de empresa, ya sea privada o gubernamental.

Hay dos clases de productos de software, los genéricos que son producidos por una organización para ser vendidos al mercado y los hechos a la medida que son desarrollados bajo pedido a un desarrollador específico. La mayor parte del gasto del software es en productos genéricos, pero hay más esfuerzo en el desarrollo de los sistemas hechos a medida.

"La Universidad Nacional Autónoma de México es una corporación pública, organismo descentralizado del Estado, dotado de plena capacidad jurídica y que tiene por fines 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".¹

La UNAM es un organismo que para optimizar sus trabajos, ha contratado servicios de empresas de desarrollo de software genérico. Para hacer más rentable, en cuanto a economía se trata, la mayoría de las dependencias universitarias han desarrollado sus propios sistemas, como lo es el caso de la Dirección General de Bibliotecas.

El sistema bibliotecario de la UNAM cuenta con diversas bibliotecas dentro del Campus CU y fuera de él. Dentro de las bibliotecas de mayor importancia con las que cuenta la UNAM están la Biblioteca Nacional de México y la Hemeroteca Nacional de México, localizadas en la zona cultural de CU, la Dirección General de Bibliotecas, localizada en el edificio de la Biblioteca Central, las Bibliotecas de la Dirección General de Servicios de Cómputo Académico, además de las que se encuentran en diversas Facultades e Institutos.

La Dirección General de Bibliotecas (DGB), desde 1966, es la dependencia universitaria encargada de coordinar el Sistema Bibliotecario de la UNAM, y tiene a su cargo la Biblioteca Central (BC).

La Dirección General de Bibliotecas pone a disposición de la comunidad en general, el acceso a diversas bases de datos que en conjunto reúnen más de cincuenta millones de registros, cubriendo con ello la necesidad de los usuarios, además de abarcar los temas

¹ Ley Orgánica de la UNAM, artículo primero.

relacionados con los contenidos de los planes y programas de estudio y proyectos de investigación de la UNAM.

Los catálogos en línea como LIBRUNAM, MAPAMEX, SERIUNAM y TESIUNAM, entre otros, proporcionan información referente a los acervos de libros, mapas, revistas y tesis existentes en las bibliotecas de la UNAM.

La DGB ofrece a la comunidad universitaria, acceso a más de 6000 títulos de revistas electrónicas en texto completo, y a más de 140 bases de datos internacionales.

La DGB da soporte a las dependencias para la actualización y mantenimiento de sus bases de datos, y por lo tanto su acervo.

Los servidores de Bases de Datos están administrados por el personal que labora para la Biblioteca Central, la cual está conformada por diversas subdirecciones, que en conjunto hacen de esta un organismo funcional. Las subdirecciones son: Dirección General, Secretaría Académica, Subdirección de Planeación y Desarrollo, Subdirección Técnica, Subdirección de Biblioteca Central, Subdirección de Servicios Especializados, Subdirección de Informática y Unidad Administrativa.

El personal administrativo que labora en la dependencia tiene contacto directo con los proveedores del material bibliográfico con el que cuenta la biblioteca y con los encargados de las dependencias de la DGB, los cuales solicitan a la dependencia diversas estadísticas, el departamento de adquisiciones es el responsable de entregar tales.

El departamento de adquisiciones para genera las estadísticas solicitadas se había estado apoyando en herramientas de bajo poder para llevar el control y registro de facturas. Diversos eran los medio de los cuales se apoyaba teniendo como resultado el manejo de varias bases de datos y en consecuencia problemas de integración, además de generación de informes en forma manual.

Para llevar a cabo este trabajo de forma más eficiente, el departamento de adquisiciones solicitó al departamento de análisis, desarrollo y mantenimiento de sistemas la unificación de sus bases de datos.

El interés de un nuevo sistema nace debido a la necesidad de un trabajo integral por parte de las áreas del departamento, ya que cada área contaba con una base de uso único. De esta manera existían tantas bases como áreas en el departamento.

Partiendo de los procesos y la información con la que contaba el departamento de adquisiciones se ha diseñado y generado una base de datos y la aplicación correspondiente para trabajar bajo esta, dando como resultado un sistema integral para el control de facturas del departamento.

Para visualizar el contexto general del presente trabajo, en el cual se describe el desarrollo del sistema, se hace una descripción breve sólo de las funciones de las subdirecciones involucradas en el desarrollo de este.

La Subdirección Técnica está conformada por los departamentos de Catalogo Colectivo, Adquisiciones y el Departamento de Procesos Técnicos. Esta subdirección se encarga de la administración del material bibliográfico perteneciente a la DGB.

Una de las labores del Departamento de Adquisiciones es llevar el control y registro de facturas por motivo de adquisiciones de material bibliográfico provenientes de las dependencias de la Dirección General de Bibliotecas. Con las labores de este departamento, la DGB puede hacer auditorias para llevar control de material bibliográfico adquirido y hacer evaluaciones a proveedores para obtener los mejor precios.

El Departamento de Adquisiciones cuenta con cinco áreas: Facturación, Transferencias, Sice, Auditoria y Proveedores. Todas las áreas trabajan con las facturas que se reciben.

Uno de los soportes principales de la DGB, es la subdirección de Informática, la cual cuenta con tres departamentos: Departamento de Producción, Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas y Departamento de Información y Telecomunicaciones.

Para cumplir con las funciones del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas, se requiere de la colaboración de personal que tenga experiencia en el área de computación como lo es un Ing. en Computación, Ing. en Sistemas, Informática o afín. Debe contar con experiencia en:

- Conocimientos de ALEPH 500. (Software genérico)
- Conocimientos indispensables: Administración de Sistema Operativo UNIX (Sun Solaris, HP/UX), y Windows 2000 Server.
- Desarrollo de programas en lenguaje C, C++, PL/SQL.
- Análisis y desarrollo de sistemas.
- Diseño de Bases de Datos.
- Conocimientos de DBMS.
- Programación orientada a eventos y objetos con lenguajes como Delphi y Power Builder.
- Programación en WEB con Java y HTML.
- Inglés Técnico.

Y contar con el siguiente perfil:

- Responsabilidad
- Buenas relaciones interpersonales.
- Iniciativa propia
- Trabajo en equipo
- Organización y administración del trabajo por objetivos.

Uno de los sistemas desarrollado por el Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas es el “*Sistema Integral para el Control y Registro de Facturas para el Departamento de Adquisiciones de la Dirección General de Bibliotecas de la UNAM.*”.

El sistema tiene como objetivo agilizar el proceso de Registro y Control de Facturas del Departamento de Adquisiciones, con lo cual se ha visto relegado en la disminución de la carga de trabajo para el personal, pero sobre todo de contar con un proceso automatizado.

Para el sistema se hizo uso de metodologías de ingeniería del software, para el análisis y diseño y administración.

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el **ciclo de vida del software** que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios.

Para sustentar el sistema realizado por el departamento, se presenta la documentación como trabajo de tesis. En cada capítulo se describe a detalle el diseño de la base y los procedimientos utilizados.

Capítulo 1. Definición del sistema.

La definición del sistema es una parte esencial para su desarrollo, debe mostrar una idea general, por lo cual en este capítulo se describen las necesidades del Departamento de Adquisiciones, la definición del problema y los objetivos del sistema.

Capítulo 2. Antecedentes

En este capítulo se presenta los antecedentes teóricos de los cuales se parte para el desarrollo de sistemas como lo es la ingeniería de software, conceptos de bases de datos y administración de las mismas.

Capítulo 3. Análisis del Sistema

En este capítulo se describen a detalles las necesidades y requerimientos de los usuarios de cada área del Departamento de Adquisiciones. Se especifica el alcance del sistema y se hace un análisis de los recursos de hardware y software disponibles.

Capítulo 4. Análisis Estructurado

En este capítulo es donde se comienza a dar forma al sistema, partiendo de diagramas que ilustran el proceso que siguen las áreas del departamento.

Capítulo 5. Desarrollo

Este capítulo describe el diseño de la Base de Datos, el diseño de la aplicación y la integración de éstos elementos.

Capítulo 6. Liberación.

En este capítulo se dan a conocer los resultados de las pruebas del sistema y la justificación de las herramientas utilizadas para su realización. Así como elementos adicionales que involucran la administración de sistemas.

1. Definición del Sistema

En la definición del sistema se deben considerar los aspectos que más adelante se verán involucrados con el análisis y desarrollo del sistema. Por lo tanto se tiene que definir claramente cuál es la problemática, definir las metas y objetivos, y establecer los requerimientos que debe tener el sistema para satisfacer a los usuarios de este.

1.1. Definición del problema

El departamento de adquisiciones cuenta con cuatro áreas: Facturación, Transferencias, Sice, Auditoria y Proveedores, las cuales, de forma independiente llevan el control de las características de las facturas provenientes de las diversas dependencias de la DGB por motivo de adquisición de material bibliográfico.

Cada área contaba con una base de datos para fines específicos, provocando que no existiera integridad de la información respecto a las demás bases.

Uno de los principales problemas que había en el área era la acumulación de trabajo, debido a la cual, el registro de facturas se iba desfasando por años, es decir, para en el año 2002 se estaban registrando facturas del año 2001.

La acumulación de trabajo se debía a que en el proceso que se llevaba a cabo existía mucha pérdida de tiempo al capturar la misma información de una factura para las diferentes áreas, es decir, los datos generales de una factura se registraban tantas veces como áreas involucradas en el proceso haya en el departamento.

Por el mismo descontrol que existía, los datos o la información en algunas ocasiones no era concordante.

Debido a las limitaciones que se tenían con los semisistemas implementados en el Departamento de Adquisiciones, fue necesario el diseño de un sistema eficiente, integral, multiusuario y amigable para el control y registro de facturas entrantes al departamento, además de poder generar diversos reportes de manera automática.

1.2. Objetivos

- Realizar un sistema que automatice el proceso de control y registro de facturas, que llevaban a cabo las áreas del Departamento de Adquisiciones de la Dirección General de Bibliotecas.
- Integrar el trabajo de las áreas del departamento, ya que manejaban, en gran parte, la misma información. Debido a que el trabajo se hacía de manera aislada, nada era compartido o integrado entre las áreas.

1. Definición del Problema

- Al plantearse como sistema multiusuarios, el nuevo sistema cuenta con restricciones de acceso, captura y edición de información, para que el control de las facturas sea mayor y en dónde cada área sólo trabaje con la información requerida.
- Generación de informes de manera automática, para que al departamento le resulte más fácil obtener el conglomerado de la información.

1.3. Diagnóstico de la problemática actual

En esta parte se describe la situación de la que se parte para considerar los objetivos y las metas a las cuales se aspira con el nuevo sistema.

Las fuentes de información para hacer el diagnóstico del sistema, son las entrevistas a las personas que manejaban los diversos semisistemas con los que contaban, cuestionarios realizados a las mismas, la observación de los procesos que realizaban y recopilación de material como los eran los informes que generaban..

El proceso de registro de facturas, constaba en que un área capturara los datos que requería de una factura que ingresaba a la DGB por concepto de adquisición de material bibliográfico, y posteriormente esa factura la tomaba otra área para capturar información general, es decir, la misma que ya había capturado el área anterior e información adicional. Con el proceso que se había llevado, cada área registraba, en su base local, información redundante, que era un problema considerable para la pérdida de tiempo en el proceso, otro de los principales problemas, era la inconsistencia de los datos para una misma factura.

Debido a la manera en que se trabajaba, la información en su mayoría era redundante e incluso la información que entre área y área debería ser la misma, en ocasiones era incongruente.

La carga de trabajo que se tenía en el departamento, era debido a las múltiples capturas que se hacían de los mismos datos para más de una área. Por lo anterior, la unificación de sus bases reduce considerablemente este problema.

Aunque cada proceso era centralizado, las áreas trabajan con la misma información. La información que se quería compartir era entregada de un área a otra en forma de reporte, lo cual implica tiempo en la solicitud, generación manual de estos reportes y la doble captura de la información, además de un gasto en papelería.

1.4. Análisis de factibilidad

El análisis de factibilidad determina cuál(es) de la(s) estrategia(s) de solución propuestas para resolver la problemática son posibles de llevarse a cabo con los recursos disponibles.

1.4.1. Factibilidad técnica

Dentro de la factibilidad técnica se determina si se puede desarrollar e implementar la estrategia de solución propuesta, empleando la tecnología existente. Más adelante se

1. Definición del Problema

verán los resultados y conclusiones del análisis de las herramientas de desarrollo con las que se cuenta, por el momento se puede decir que la DGB cuenta con el software necesario para realizar cualquier tipo de sistema.

1.4.2. Factibilidad económica

Dentro de la factibilidad económica, se cuestiona acerca de los fondos necesarios para desarrollar e implementar un sistema de información, dados los requerimientos de otros proyectos de capital dentro de la organización.

Por ser un proyecto desarrollado dentro y para la Dirección General de Bibliotecas, resulta factible económicamente, ya que no se invirtió algún gasto extraordinario para su realización. Ya que la DGB cuenta con el personal para su realización, cuenta con software para desarrollar y cuenta con hardware para los usuarios del sistema, servidor de aplicaciones y/o bases de datos.

1.4.3. Factibilidad legal

Este factor refiere a que no haya algún conflicto entre el sistema que se está considerando y la capacidad de la organización para descargar sus obligaciones legales.

Actualmente, la legalidad también tiene que ver con el uso de licencias de software que se emplea para desarrollar los sistemas.

La Dirección General de Bibliotecas al contar con recursos de software, cuenta con la legalidad de las licencias.

El software libre también es válido para el desarrollo del sistema, con lo cual no existe problema alguno de legalidad.

1.4.4. Factibilidad operacional

Consiste en determinar si el análisis y diseño están basados en el ambiente organizacional, los procedimientos existentes y el personal con que opera la organización.

Debido a que el sistema integra las bases de datos existentes del Departamento de Adquisiciones, el proceso de registro y recopilación de datos de las facturas es más sencillo de lo que hasta ahora había sido. Con la unificación las áreas pueden compartir información sin necesidad de volver a capturar, y por lo tanto la operación del sistema no resulta compleja.

1.4.5. Factibilidad de programa

Significa que el diseño del sistema debe ser capaz de ser operativo dentro de algún plazo de tiempo.

1. Definición del Problema

Debido a que no hay grandes modificaciones en cuanto al procedimiento de registro de datos de las facturas que se llevaba a cabo en el departamento, el sistema tiene un lapso de tiempo considerable para que sea utilizado, es decir, la vida del sistema no se considera sea corta.

1.5. Beneficios del sistema

Para que se lleve a cabo la realización de cualquier sistema, se requieren considerar algunos aspectos para hacer un análisis de los beneficios que traerá el sistema, cuyos resultados sea satisfactorios al usuario.

1.5.1. Diseño del sistema

El diseño normalizado de la base de datos y de la aplicación hacen que el sistema sea más accesible al usuario.

Unificando las bases y proceso del departamento, el registro de información ya no permitirá la carga de trabajo.

1.5.2. Interfaz Gráfica

Permite un manejo intuitivo de los elementos de las pantallas. El hecho de que las diversas áreas puedan compartir información de la base de datos, implica una disminución en el consumo de papel, ya que entre áreas compartían información mediante reportes impresos.

1.5.3. Informes

Con la ayuda de la interfaz gráfica es posible visualizar previamente las consultas antes de ser impresas, si es necesario, y si hay algún detalle que modificar o el reporte no es lo que se quiere, es posible generar nuevamente el informe sin necesidad de imprimir y hacer un gasto en papelería.

1.5.4. Procesos Manuales

Los cálculos manuales hacen más lento cualquier proceso, por lo cual en lo posible hay que evitarlos. Sin procesos manuales se agiliza el trabajo evitando también los errores humanos.

1.5.5. Base de Datos relacional

El hecho de contar una base de datos facilita la integridad de la información, facilita el crecimiento sin grandes modificaciones a la vez que facilita el mantenimiento del sistema.

1. Definición del Problema

1.5.6. Diseño Distribuido

Permite la consulta de la información en el lugar que se requiere. Permite la autonomía de operación en las áreas del departamento.

1.6. Metas para el sistema

Las metas son logros por alcanzar y sirven para establecer el marco de referencia para el proyecto de desarrollo.

1.6.1. Metas para el Proceso

1.6.1.1. Cualitativas.

Elaboración del sistema con recursos disponibles o que no impliquen un costo elevado a la DGB.

1.6.1.2. Cuantitativas.

Liberación del sistema en un lapso de 6 meses

1.6.2. Metas para el Producto Final

1.6.2.1. Cualitativas

Agilizar el proceso de registro y control de facturas

Reducir la carga de trabajo del personal

Reducir los errores humanos en el cálculo de operaciones

1.6.2.2. Cuantitativas

Generación de 49 reportes de apoyo al departamento

1.7. Requerimientos del Sistema

Los requerimientos especifican las capacidades que debe tener un sistema para la solución de un problema. Estos se establecen para la funcionalidad, el rendimiento, el equipo, la programación en el equipo y las interfaces con el usuario.

1.7.1. Confiabilidad

El sistema cuenta con un proceso de identificación del usuario, ya que de esta manera se dará acceso a los módulos correspondientes sin intervenir con los datos de otra área.

1.7.2. Disponibilidad

Debido a que la captura de facturas es a lo largo del día, el sistema esta disponible en cualquier momento. Por tal razón la base de datos está en un servidor que cuenta con el soporte suficiente y con equipo adicional para que una caída de energía no detenga el funcionamiento, es decir, se cuenta con un regulador de voltaje. Además cualquier área debe de tener acceso al sistema y disponibilidad de la base de datos en cualquier momento.

1. Definición del Problema

1.7.3. Flexibilidad

El sistema cuenta con las opciones que al usuario le sean necesarias y la posibilidad de moverse entre ellas, es decir, debe de estar definido por módulos de acceso.

1.7.4. Crecimiento

El sistema ha sido pensado y diseñado con posibilidades de crecimiento, ya que en diversas ocasiones van aumentando las necesidades de los usuarios y si no se considera el crecimiento, el sistema quedará obsoleto teniendo abierta la posibilidad de reemplazo.

En esta parte se considerarán opciones de crecimiento por área, es decir, cada área va plantear si hay alguna posibilidad de que su módulo sea modificado posteriormente.

1.7.5. Mantenimiento

El mantenimiento es brindado por parte del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas de la Dirección General de Bibliotecas.

El mantenimiento abarca:

- Importación de datos, de las bases que se tenían en el departamento a la base normalizada.
- Carga masiva de datos a la base, como lo es la actualización de catálogos.
- Actualización de la aplicación.
- Actualización de la Base de Datos.
- Depuración de información innecesaria en la base, etc.
- Generación de reportes.

2. Antecedentes

2.1. Nociones de Ingeniería de Software

“Ingeniería del Software es una aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software”¹

La ingeniería del software es una tecnología multicapa la cual permite instalar todo el sistema en una sola PC o distribuirlo en múltiples PC sobre una red para adaptarlo a instalaciones de gran tamaño.

Cualquier enfoque de ingeniería debe apoyarse sobre un compromiso de organización de calidad, para ello se utilizan los siguientes elementos:

- Procesos. Es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo relacional y oportuno de la ingeniería del software.
- Métodos. Indican como sustituir técnicamente el software. Los métodos abarcan una gran parte de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento.
- Herramientas. Proporcionan un enfoque automático o semiautomático para el proceso y para los métodos.

2.1.1. Objetivos de la Ingeniería del Software

- Definir la Ingeniería de Software y explicar su importancia.
- Discutir los conceptos de producto de software y proceso de software.
- Explicar la importancia de la visibilidad de los procesos.
- Introducir la noción de responsabilidad profesional

2.1.2. Características de los productos de software

- Manténibles. Debe ser posible que el software evolucione y que siga cumpliendo con sus especificaciones.
- Confiables. El software no debe causar daños físicos o económicos en el caso de fallos.
- Eficiencia. El software no debe desperdiciar los recursos del sistema.
- Utilización Adecuada. El software debe contar con una interfaz de usuario adecuada y su respectiva documentación.

¹ Ingeniería de Software, Roger S. Pressman

2.1.3. Paradigmas de Ingeniería del Software.

La Ingeniería de Software esta compuesta de pasos que abarcan los métodos, las herramientas y los procedimientos, estos métodos se denominan paradigmas.

2.1.3.1. Modelos genéricos.

- *Modelo de Cascada.* Separar en distintas fases de especificación y desarrollo.
- *Desarrollo Evolutivo.* La especificación y el desarrollo están intercalados.
- *Prototipado.* Un modelo sirve de prototipo para la construcción del sistema final.
- *Transformación Formal.* Un modelo matemático del sistema se transforma formalmente en la implementación.
- *Desarrollo basado en Reutilización.* El sistema es ensamblado a partir de componentes existentes.

2.1.3.2. Modelos Híbridos.

- *Modelo en Espiral.* Separa en distintas fases especificación y desarrollo.

2.1.3.3. Fases del modelo en Cascada.

- Análisis de requerimientos y definición.
- Diseño del sistema y del software.
- Implementación y prueba de unidades
- Integración y prueba del sistema.
- Operación y mantenimiento.

La dificultad en esta modelo reside, en hacer cambios entre etapas.

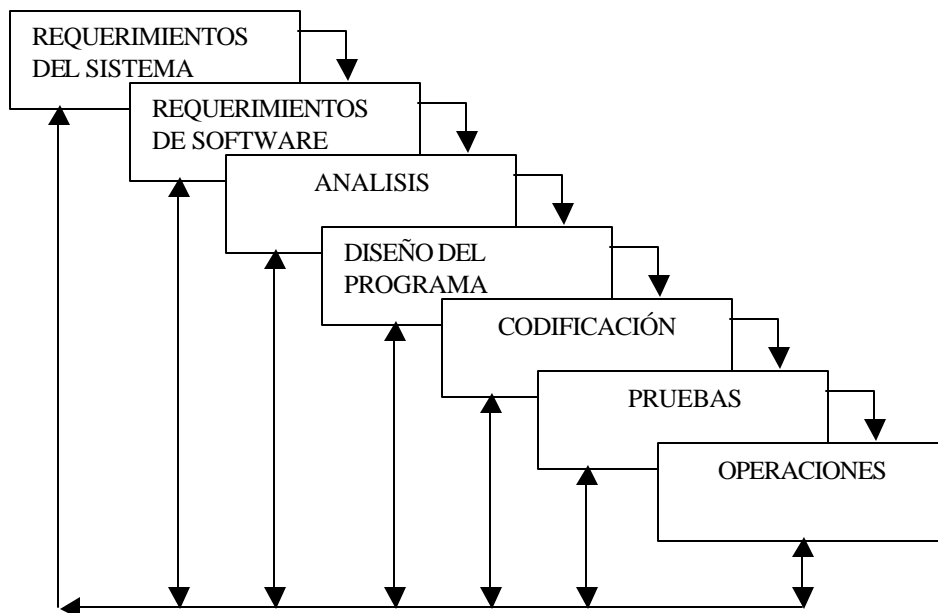


Figura 2.1 Modelo en Cascada Modificado del Desarrollo de Sistemas

2. Antecedentes

2.1.3.3.1.1. Documentos del Modelo en Cascada

Actividad	Documento Producido
Análisis	Requerimientos.
Definición	Requerimientos.
Especificación	Especificación funcional y plan de pruebas de aceptación.
Diseño Arquitectural	Especificación funcional de arquitectura y plan de pruebas del sistema.
Diseño de Interfases	Especificación de las interfaces y plan de pruebas de integración.
Diseño detallado	Especificación del diseño y plan de pruebas de unidades.
Codificación	Código de programa.
Prueba de Unidades	Reporte de prueba de unidades.
Prueba de Módulos	Reporte de módulos.
Prueba de Integración	Reporte de integración y manual de usuario final.
Prueba del Sistema	Reporte de prueba del sistema.
Prueba de Aceptación	Sistema final más la documentación.

Tabla 2.1 Documentos del Modelo en Cascada

2.1.3.4. Evolutivo

2.1.3.4.1. Problemas

- Poca visibilidad en el proceso
- Los sistemas están pobremente especificados
- Se requieren habilidades especiales.

2.1.3.4.2. Aplicabilidad

- Aplicabilidad
- Para sistemas interactivos pequeños o medianos.
- Para partes de sistemas grandes (p.ej. la interfaz de usuario).
- Para sistemas de corta vida.

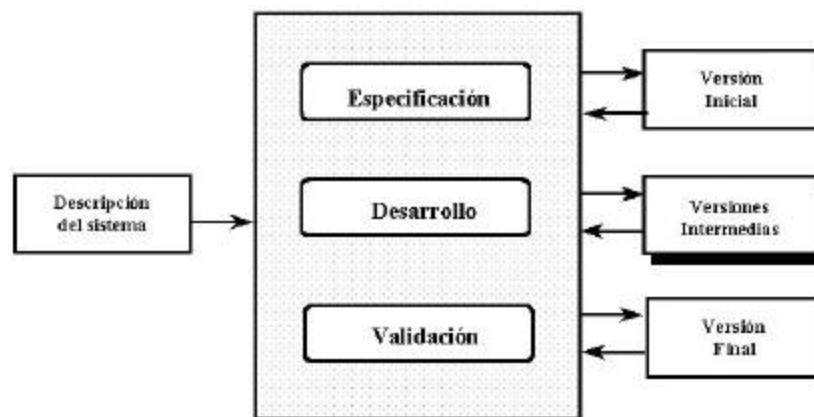


Figura 2.2 Modelo Evolutivo²

² Ian Sommerville, Software Engineering, Addison_Wesley, 6th edition, 2000

2. Antecedentes

2.1.3.5. Prototipado

Un modelo sirve de prototipo para la construcción del sistema final. Este modelo toma las siguientes formas:

- Prototipo en papel
- Prototipo funcional
- Programa que ejecute toda o parte de la función deseada.
- Prototipado exploratorio. El objetivo es trabajar con clientes hasta evolucionar a un sistema final, a partir de una especificación inicial. Se debe comenzar con unas especificaciones bien entendidas.
- Prototipado de “throw-away”. El objetivo es entender los requerimientos del sistema. Se puede comenzar con especificaciones poco entendidas.

2.1.3.6. Modelo en Espiral

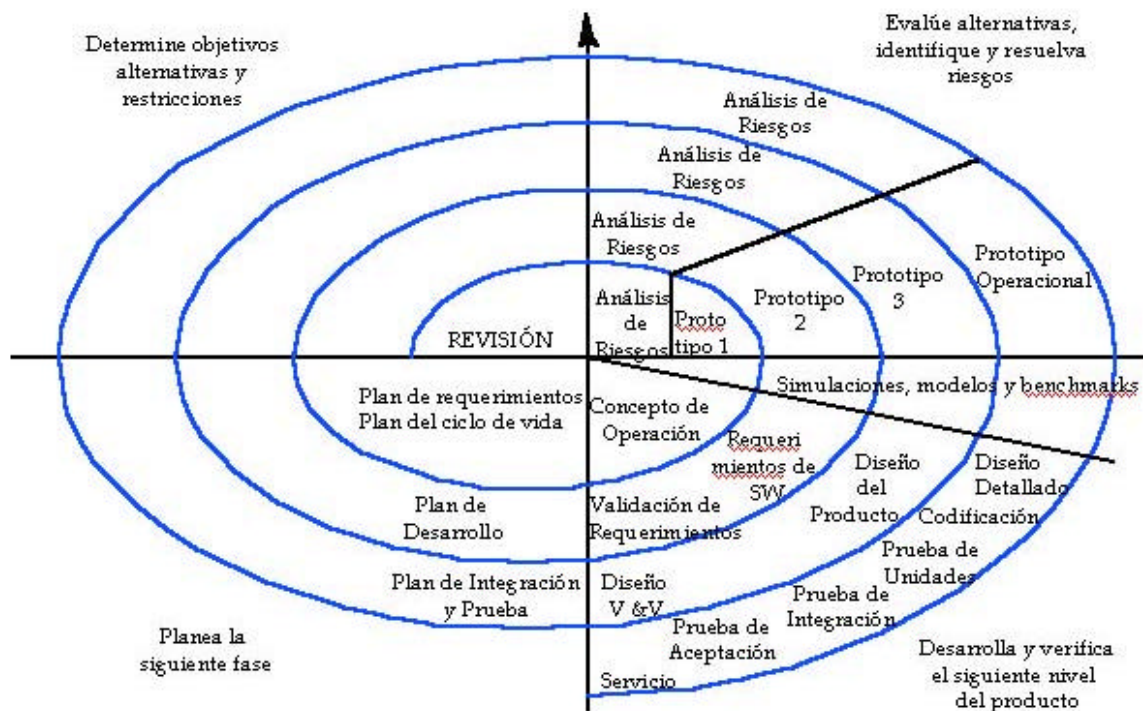


Figura 2.3 Modelo en Espiral ³

³ Ian Sommerville, Software Engineering, Addison_Wesley, 6th edition, 2000.

2.1.3.6.1. Fases del Modelo en Espiral

- Planteamiento de Objetivo
- Identificación y Reducción de Riesgos
- Desarrollo y Validación. Se elige un modelo apropiado para la siguiente fase del desarrollo.
- Planeación. Se revisan el proyecto y se trazan los planes para la siguiente ronda del espiral.

2.1.3.6.2. Desventajas del modelo en espiral

- El desarrollo contractual especifica el modelo del proceso y los resultados a entregar por adelantado.
- Requiere de experiencia en la identificación de riesgos.
- Requiere refinamiento para uso generalizado.

2.1.4. Tabla comparativa de los Paradigmas de ingeniería del Software.

Modelo	Observaciones
Cascada	Cada actividad produce un documento o resultado
Evolutivo	Muy caro al producir documentos en cada iteración
Formales	Cada fase produce un documento
Reutilización	Importante contar con documentos de componentes reutilizables
Espiral	Cada segmento y cada anillo del espiral deben producir documentación.

Tabla 2.2 Paradigmas de ingeniería del Software

2.1.4.1. Técnicas de Cuarta Generación (T4G)

El término T4G abarca un amplio espectro de herramientas de software que facilitan al desarrollador especificar algunas características de alto nivel en su trabajo. Mientras mayor sea el nivel en el que se especifique el desarrollo de software, más rápido y mejor estructurado quedará el nuevo sistema.

En la actualidad un entorno para desarrollo de software que soporta el paradigma T4G puede incluir todas o algunas de las herramientas siguientes:

- Lenguajes no procedimentales de consulta de bases de datos
- Generación de informes
- Manejo de datos
- Generación de códigos
- Generación automatizada de HTML
- Lenguajes similares utilizados para la creación de sitios WEB, usando herramientas de software avanzado

Para fines de desarrollo del presente trabajo, se utilizará como herramienta de ingeniería de software el paradigma *Ciclo de vida Clásico Modificado*.

2.1.5. Metodología CASE

La metodología case se basa en la utilización de herramientas de software que facilitan el desarrollo de sistemas. Estas herramientas incluyen lenguajes de cuarta generación para la consulta y manipulación de los datos, generación automática de código fuente, basándose en las especificaciones del analista, ayudan en la interacción y definición de pantallas y ofrecen facilidades para la documentación del sistema.

Esta metodología inicia con la recolección de requisitos que son traducidos directamente en un prototipo operacional. Para sistemas pequeños, es fácil traducir los requerimientos del cliente de manera que una herramienta CASE los interprete. Sin embargo, para sistemas más grandes se requiere un mayor tiempo para el análisis y el diseño del sistema, de tal forma que la herramienta CASE pueda interpretarla.

La ventaja en la utilización de la metodología CASE es que una vez realizado el análisis y el diseño permite efectuar con rapidez la documentación del sistema, la generación del código fuente y la generación de pantallas.

2.1.5.1. Herramientas CASE

Es el conjunto de una serie de productos orientados a la automatización de la producción de software, específicamente al desarrollo asistido de sistemas. Por las fases del ciclo de vida que abordan, las herramientas CASE, se clasifican como:

- Upper Case.
Son las que automatizan esencialmente las etapas de planificación, análisis y diseño, es decir, las primeras etapas del desarrollo de sistemas.
- Lower Case.
Son las que automatizan la fase de producción o generación de código y su mantenimiento.

Una herramienta CASE moderna debe contar con las siguientes características:

- Dibujo de diagramas
- Soporte de navegación
- Soporte multiusuario
- Generación de código
- Ingeniería de reversa
- Integración con otras herramientas
- Intercambio de modelos

2.2. Nociones de Bases de datos

2.2.1. Base de Datos

La base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales innecesarias; su finalidad es la de servir a una aplicación o mas, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan.⁴

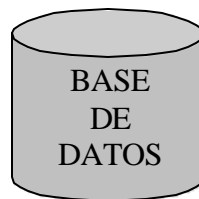


Figura 2.4 Base de Datos

2.2.2. Entidades

Se puede definir como entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual se desea guardar información, por ejemplo: "USUARIO", "AREA", "FACTURA". Figura 2.5.

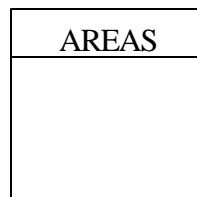


Figura 2.5 Entidad

Las entidades se pueden clasificar en:

- Regulares. Aquellas que existen por sí mismas y que la existencia de un ejemplar en la entidad no depende de la existencia de otros ejemplares en otra entidad.
- Débiles: Son aquellas entidades en las que se hace necesaria la existencia de ejemplares de otras entidades distintas para que puedan existir ejemplares en esta entidad.

2.2.3. Atributo

Al atributo frecuentemente se le llama elemento de datos, campo de datos, campo ítem de datos o ítem.

⁴ Organización de las bases de datos, James Martín.

Existen cuatro tipos de atributos:

- **Obligatorios:** aquellos que deben tomar un valor y no se permite ningún ejemplar no tenga un valor determinado en el atributo. Es decir, deben existir y ser únicos.
- **Opcional:** aquellos atributos que pueden tener valores o no tenerlo.
- **Monoevaluado:** aquel atributo que sólo puede tener un único valor.
- **Multievaluado:** aquellos atributos que pueden tener varios valores

2.2.4. Valor de los Datos

El valor de los datos es la información o los datos mismos contenidos en cada campo. Los valores que toman los campos pueden ser cuantitativos, cualitativos o descriptivos, dependiendo de la manera en que los campos de datos describan a la entidad.

2.2.5. Claves o llaves o Identificadores

El modelo entidad - relación exige que cada entidad tenga un identificador, se trata de un atributo o conjunto de atributos que identifican de forma única a cada uno de los ejemplares de la entidad. De tal forma que ningún par de ejemplares de la entidad puedan tener el mismo valor en ese identificador.

2.2.6. Tipos de Relaciones

2.2.6.1. Relación Uno a Uno

En este tipo de relación, cada registro de la tabla original se enlaza con un solo registro de la tabla asociada. Los campos que sirven de enlace deben ser la clave primaria en las tablas.

Las tablas implicadas en el tipo de relación uno a uno pueden incorporarse una dentro de la otra.

2.2.6.2. Relación Uno a Muchos

Esta es la relación más común. En ella, cada registro de la tabla fuente puede enlazarse a varios registros de la tabla asociada. Para ello, el campo que sirve de enlace debe estar presente en ambas tablas a la vez.

2.2.6.3. Relación Muchos a Muchos

Este tipo de relación se presenta cuando es factible asociar un registro de una tabla con varios registros de otra tabla y viceversa.

2.2.7. Normalización

Es el proceso de agrupar a los campos de datos en tablas que representan a las entidades y sus relaciones. La teoría de la normalización está basada en la observación de que un cierto conjunto de relaciones, tiene mejores propiedades en un medio de inserción, actualización y supresión, que las que tendrían otros conjuntos de relaciones conteniendo los mismos datos.

El objetivo principal de la normalización es utilizar las técnicas de descomposición sin pérdidas como ayuda en el diseño de bases de datos relacionales.

El objetivo principal del proceso de normalización es reducir la redundancia y, por lo tanto, evitar problemas con operaciones de actualización.

La teoría de la normalización está basada en el concepto de Formas Normales.

2.2.7.1. Formas Normales

Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones.

Reglas de normalización (Figura 2.6).

- 1NF Todos los atributos deben tener un solo valor. Si un atributo tiene múltiples valores, se crea una entidad adicional y se relaciona con la entidad original mediante una relación uno a muchos.
- 2NF Un atributo debe ser dependiente del identificador único completo.
- 3NF Ningún atributo que no-identificador único puede ser dependiente de otro atributo no-identificador único (su objetivo es eliminar redundancia).

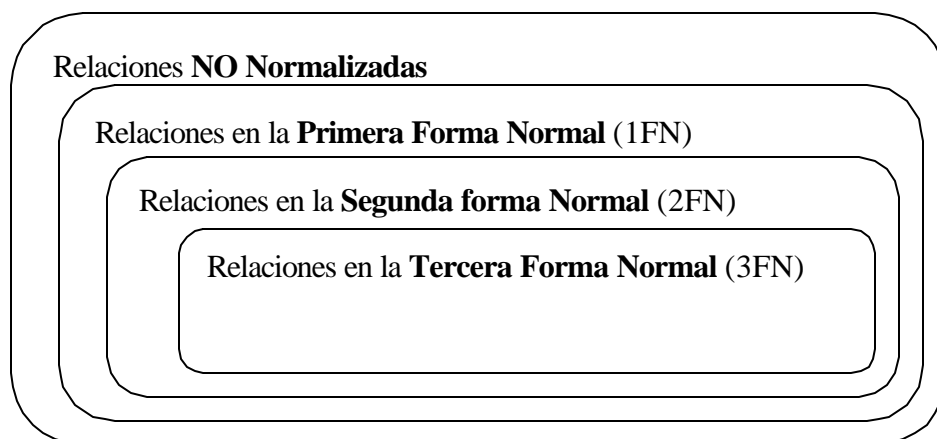


Figura 2.6 Reglas de Normalización

2.2.8. Objetivos de la organización de la base de datos

2.2.8.1. Desempeño

El sistema de bases de datos debe tener la capacidad para manejar un adecuado caudal de transacciones.

2.2.8.2. Redundancia mínima

La redundancia ocupa más espacio en el almacén de datos y requiere múltiples operaciones de actualización. Debido a que diferentes copias de la misma información suelen hallarse en diferentes etapas de actualización, la redundancia da a menudo origen a respuestas incoherentes.

Para evitar la redundancia es necesario identificar los datos que son utilizados en común por varias aplicaciones y almacenarlos una sola vez. Si hay una sola copia de ítems de datos que son necesarios para diversos fines, es preciso almacenar y mantener las relaciones pertinentes. Por tal motivo, en ocasiones, no siempre se elimina del todo la redundancia.

2.2.8.3. Capacidad de búsqueda

En la mayoría de los sistemas es necesaria la consulta de datos o producción de informes, por lo cual es necesaria una exploración rápida y flexible de los datos.

2.2.8.4. Integridad

Cuando una base de datos incluye información utilizada por muchos usuarios, lo cual es frecuente, es importante que no puedan destruirse los datos almacenados ni las relaciones que existen entre los distintos ítems. El almacenamiento de los datos y los procedimientos de actualización e inserción deben asegurar que el sistema pueda recuperarse de estas contingencias sin daño para los datos.

2.2.8.5. Migración de datos

A medida que cambia la necesidad de un conjunto de datos, es conveniente mudarlos a posiciones más o menos accesibles, de acuerdo a su actividad. El proceso de ajustar el almacenamiento de los datos de acuerdo con su popularidad se llama migración de datos. En algunos sistemas se hace automáticamente. En otros, la operación está a cargo de los programadores de sistema o del administrador. La migración de datos es importante, también, para la creación de respaldos de la base de datos.

2.2.9. Pasos en el Diseño de una Base de Datos (Diagrama 2.1).

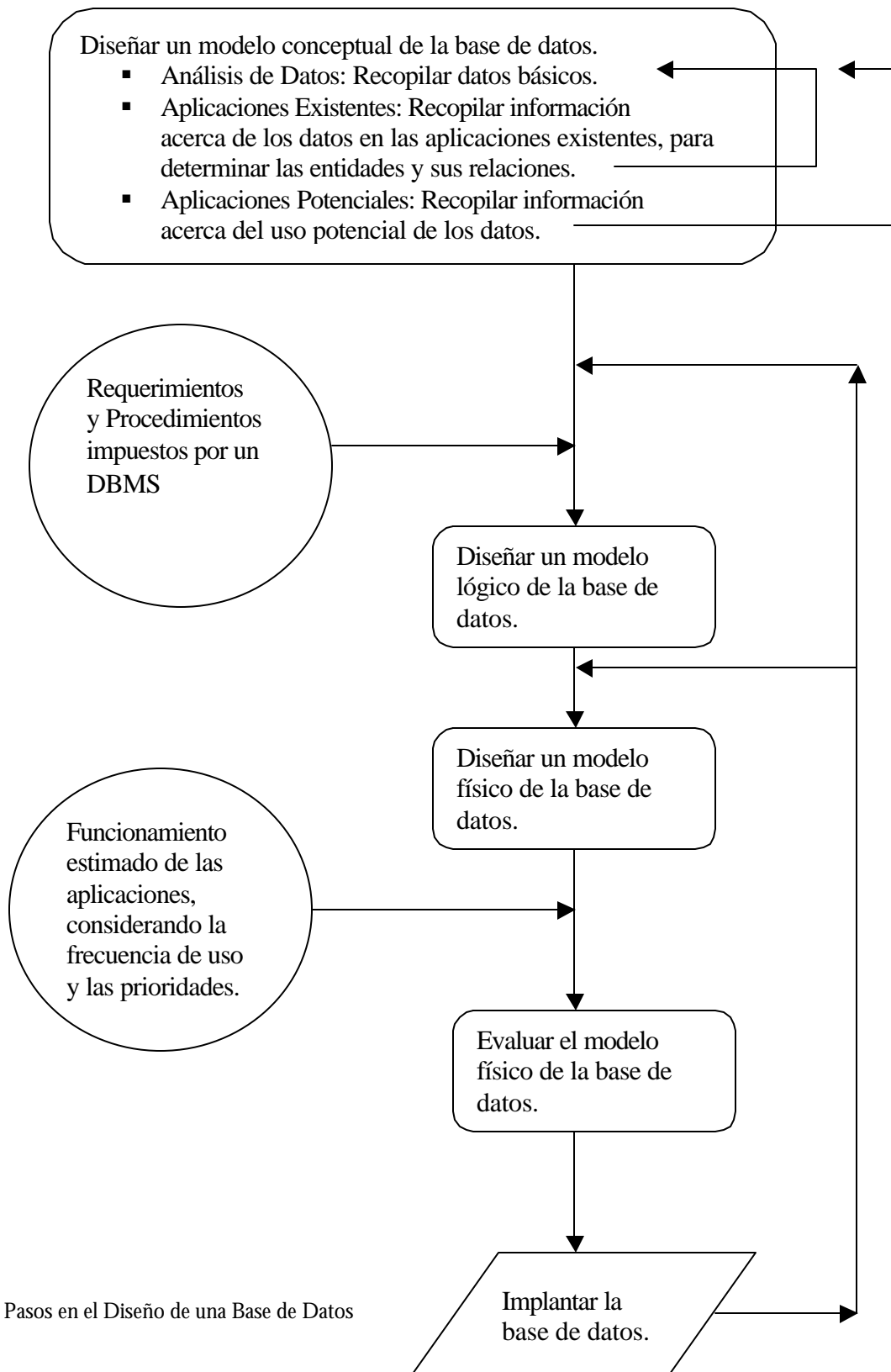


Diagrama 2.1 Pasos en el Diseño de una Base de Datos

2.2.10. Modelo de Datos

Los datos han de ser interpretados para que se conviertan en información útil, ésta interpretación supone un fenómeno de agrupación y clasificación.

En la era actual y con el auge de los medios informáticos aparece el almacenamiento en soporte electromagnético, ofreciendo mayores posibilidades de almacenaje, ocupando menos espacio y ahorrando un tiempo considerable en la búsqueda y tratamiento de los datos. Es en este momento donde surge el concepto de bases de datos y con ellas las diferentes metodologías de diseño y tratamiento.

El objetivo básico de toda base de datos es el almacenamiento de símbolos, números y letras cadentes de un significado en sí, que con un tratamiento adecuado se convierten en información útil.

Según van evolucionando los tiempos, las necesidades de almacenamiento de datos van creciendo y con ellas las necesidades de transformar los mismos datos en información de muy diversa naturaleza. Esta información es utilizada diariamente como herramientas de trabajo y como soporte para la toma de decisiones por un gran colectivo de profesionales que toman dicha información como base de su negocio. Por este motivo el trabajo del diseñador de bases de datos es cada vez más delicado, un error en el diseño o en la interpretación de datos puede dar lugar a información incorrecta y conducir al usuario a la toma de decisiones equivocadas.

Se hace necesario la creación de un sistema que ayude al diseñador a crear estructuras correctas y fiables, minimizando los tiempos de diseño y explotando todos los datos, nace así la metodología de diseño de bases de datos.

La metodología de diseño de datos divide cada modelo en tres esquemas:

2.2.10.1. Modelo Conceptual

Se trata de una representación gráfica legible por el usuario y que nos aporta el flujo de información dentro de una organización. No existen reglas para su construcción y se debe realizar siempre el esquema más sencillo posible para la comprensión por parte del usuario de la base de datos.

2.2.10.2. Modelo Lógico

Se trata de una representación gráfica, mediante símbolos y signos normalizados, de la base de datos. Su objetivo es representar la estructura de los datos y las dependencias de los mismos, garantizando la consistencia y evitando la duplicidad.

2.2.10.3. Modelo Físico

Se trata del almacén de los datos, es la base de datos en sí misma, el soporte donde se almacenan los datos y de donde se extraen para convertir los datos en

información. En función del gestor de bases de datos empleado las reglas de almacenamiento varían.

2.2.11. Nociones de Manejadores de Bases de Datos

Un Sistema Manejador de Bases de Datos DBMS (Data Base Management System) agrupa varios programas que permiten crear, utilizar y dar mantenimiento a una base de datos. Dichos sistema cuenta con tres tipos de lenguajes:

- Lenguaje de Definición de Datos para describir la información elemental y la manera como está estructurada en la base de datos.
- Lenguaje de Manipulación de Datos para facilitar el manejo de la información.
- Lenguaje para Consultar una Base de Datos y extraer información específica.

Los DBMS presentan las siguientes particularidades:

- Eliminan la redundancia de datos y aumentan la uniformidad de los mismos, los cuales por lo general se almacenan una sola vez en la base de datos.
- Aumentan la flexibilidad para soportar los cambios y satisfacer las nuevas necesidades de los usuarios.
- Permiten el compartimiento de recursos entre varios usuarios. Ello significa que los datos pueden utilizarse en varias aplicaciones al mismo tiempo.
- Permite una mejor estructuración de los archivos, lo cual hace posible un acceso más rápido a los archivos que se integran, relacionan y actualizan.

Manejadores de BD:

2.2.11.1. Oracle



- Es un manejador de bases de datos relacional, utilizado para misiones críticas de negocios.
- Soporta plataforma UNIX, Windows NT, Windows 95, OS/2, Netware
- Utiliza sofisticados mecanismos de seguridad para controlar el acceso a los datos a través del manejo de privilegios.
- Realización de copias de seguridad y recuperación.
- Conectividad de carácter abierto, permite acceder hacia y desde paquetes de software de otros fabricantes mediante ODBC y SQLNet (protocolo de comunicación).
- Su principal herramienta de desarrollo es Developer/2000, el cual incluye un conjunto de herramientas para la creación de aplicaciones.

2.2.11.2. PostgreSQL



- Software de código abierto
- El servidor de la base de datos se puede acceder por medio de varios lenguajes: C, Perl, Visual Basic, Delphi, etc.
- Corre bajo plataforma Linux.
- En la versión 7.0 se han implementado las llaves foráneas, para asegurar la integridad relacional de las bases de datos.
- En la versión 7.0 ya se permite el bloqueo de tablas, cuando se encuentran abiertas, mejorando la confiabilidad de la concurrencia.
- Se permite el uso de la cláusula Having sin necesidad de tener una función de agregación.
- Soporta Clases, herencia, tipos y funciones
- Soporta restricciones (Constraints), Disparadores (triggers), reglas (rules) e integridad transaccional.

2.2.11.3. ASAnyWhere



- ASAnyWhere de Sybase es un manejador de bases de datos relacional, utilizado para aplicaciones empresariales.
- Corre bajo plataformas UNIX, Windows NT, Windows 95, etc
- Soporta conectividad con otros manejadores mediante ODBC.
- Provee seguridad mediante contraseñas de entrada y el uso de privilegios proporcional mecanismos de respaldo y recuperación.
- Cuenta con productos de desarrollo para la creación de aplicaciones cliente/servidor (Power Builder) y aplicaciones en el Web mediante Jaguar de Power Soft.

2.2.11.4. Microsoft SQL Server



- Es un manejador de bases de datos relacional, multiusuario, diseñado para soportar una gran cantidad de transacciones sobre computadoras personales.
- Corre sólo bajo la plataforma Windows NT.

- En cuanto a arquitectura es muy parecido al manejador de Sybase, sólo que ocupa menos recursos y esta enfocado para sistemas de mediano tamaño.
- Ofrece un alto rendimiento sobre el sistema operativo Windows NT.
- Soporta la arquitectura cliente/servidor, es decir, divide una aplicación en dos partes, el front-end (cliente) y el back-end (servidor).
- Soporta aplicaciones distribuidas
- Soporta alta seguridad, que se ve reforzada con la seguridad que ofrece el sistema operativo Windows NT.

2.3. Nociones de Herramientas de Desarrollo

Las herramientas de desarrollo, son programas que sirven principalmente par el desarrollo de interfaces gráficas.

2.3.1. Parámetros a considerar en las herramientas de desarrollo

- Entorno físico sobre el que funciona.
- Sistema operativo sobre el que opera.
- Gestores de bases de datos o fuentes a los que es aplicable.
- Grado de especialización.
- Profundidad del análisis que realiza.
- Información adicional que aporta.
- Necesidades de formación que exige a sus usuarios.
- Costo económico.

2.3.2. Herramientas de Desarrollo

2.3.2.1. Visual Basic

- Es un producto de Microsoft.
- Corre sobre plataforma Windows y es de las utilizadas en el mercado.
- Permite desarrollar sistemas amigables y fáciles de utilizar.
- Soporta la creación de aplicaciones cliente/servidor mediante el uso de ODBC (Open Data Base Connectivity).
- Utiliza programación orientada a eventos.
- Permite crear aplicaciones ejecutables que pueden ser distribuidos a todos los usuarios.
- No soporta la recuperación de grandes cantidades de información.

2.3.2.2. Delphi

- Es un producto de Borland.
- Corre bajo plataforma Windows.
- Permite el desarrollo de aplicaciones en InternetWeb mediante asistentes (wizards).
- Dispone de una biblioteca de 25 componentes nativos para Internet para crear aplicaciones como lectores de noticias y servidores de correo.

- Soporte completo y alto desempeño a manejadores de bases de datos Oracle.
- Soporta múltiples manejadores de bases de datos de alto desempeño en 32 bits, como SQL Link, Sybase, Informix, DB2, SQL e InterBase
- Soporta estándares creados por Microsoft Active X
- Utiliza la lógica de Pascal.

2.3.2.3. Power Builder

- Es un producto de Sybase.
- Soporta el desarrollo de aplicaciones cliente/servidor
- Permite crear aplicaciones ejecutables y generar discos de instalación.
- Utiliza la programación orientada a eventos y hace uso de la programación orientada a objetos como, encapsulamiento, polimorfismo y herencia.
- Soporta la recuperación de grandes cantidades de información.

2.4. Nociones de Sistemas Operativos.

Un Sistema Operativo (SO) es el conjunto de programas básicos y utilidades que hacen que una máquina funcione y resulte útil a los usuarios.

El sistema operativo comienza a trabajar cuando en enciende la computadora, y gestiona los recursos de hardware de la máquina en los niveles más básicos.

2.4.1. Herramientas.

- Compilador de C
- Enlazador
- Ensamblador
- Intérprete de comandos
- Amplia biblioteca de C
- Kernel

Aparte de estas herramientas bastante básicas, también se consideran elementos fundamentales de un sistema operativo completo un amplio número de programas orientados a diferentes tareas, incluyendo editores de texto, Lenguajes de programación, etc.

2.4.2. Características

- Administración de tareas
- Administración de usuarios
- Manejo de recursos
- Organización Interna

2.4.3. Sistemas Operativos

2.4.3.1. Solaris

- Portabilidad.
- Escalabilidad.
- Compatibilidad.
- Seguridad.
- Cuenta con ventanas para manejo de servicios rápidos.
- Integra servicios desktop.
- Bibliotecas gráficas.
- Administración de calendario.
- Herramientas de imagen.
- Se puede obtener información sobre dispositivos.
- Brinda paquetes de software (colecciones de archivos y directorios) y clusters (colección de paquetes).
- El kernel de solaris tiene multithread en vez de llave maestra la cual brinda mayor impacto en como está diseñado el controlador.
- Este sistema cuenta con comandos los cuales ayuda a manejar el sistema.

2.4.3.2. Linux

- Código abierto
- Multiusuario y multitarea.
- Tiene las mismas características que UNIX. Pero también aporta elementos propios.
- Soporta el modo protegido de los procesadores de intel 80x86. Pero también las prestaciones más avanzadas de todos los procesadores que soporta su núcleo: PowerPC, SPARC, MIPS, sistemas IBM 390, PDA's, etc.
- Soporta la paginación de memoria a disco duro.
- Entorno gráfico.

2.4.3.3. Windows 2000

- Es el último sistema operativo que reemplazo al NT.
- Seguridad.
- Control de una red y de la PC.
- Multitareas.
- Soporte de memoria.
- Multiprocesamiento simétrico.
- Plug and play.
- Sistema de archivos.
- Servicios de terminal.
- Servicios de instalación remota.

3. Análisis del sistema

En ésta etapa se determinan las necesidades y requerimientos del usuario, la tarea del análisis de requerimientos y especificaciones es un proceso de descubrir, refinar y especificar.

El análisis del sistema llega a la raíz del problema o la necesidad, y define los requerimientos de los usuarios.

Las actividades y eventos que comprende el análisis del sistema se dirigen a responder a las preguntas:

- ¿Qué va a incluir el sistema?
- ¿Qué información se necesita?
- ¿Quién la necesita?
- ¿Cuándo?
- ¿Dónde?
- ¿En qué forma?
- ¿En dónde se origina?
- ¿Cómo puede obtenerse?

Uno de los errores más perjudiciales que se presentan en el análisis del sistema es pensar en términos computacionales y hacer énfasis primeramente en el componente estructural de la tecnología.¹

El primer enfoque del análisis del sistema debe estar en:

- Las operaciones, en este caso, del Departamento de Adquisiciones.
- Los requerimientos de los usuarios.
- Los componentes de la entrada.
- La salida.
- La base de datos.
- Los controles.

Respondiendo de manera general a las preguntas anteriores, el sistema contiene toda la información referente a las facturas provenientes de las dependencias por concepto de adquisición de material bibliográfico.

Los datos a considerar en el sistema son los propios de la factura, aunando a estos, los datos de clasificación manual que hace el Departamento de Adquisiciones, como el tipo de proceso que se efectúa para cada ejemplar de la factura. Por ejemplo, un título de un ejemplar incluido en una factura, puede ser un cargo o estar destinado a un proceso técnico. Se coloca “C” ó “P”, este paso es inevitable ya que es parte de la recepción.

¹ Ing. Orlando Zaldívar Zamorategui, Apuntes de Ingeniería de Programación.

3. Análisis del Sistema

En la copia física de la factura, se hacen anotaciones que posteriormente se vaciaran en la base de datos, como lo es la suma del costo por título, en este caso se hace así porque aún no se acuerda con los proveedores incluir tal dato.

Los datos generales de las facturas, los ingresan las áreas de recepción, como lo son el área de facturación y transferencias, ya que posteriormente las otras áreas harán uso de ellos, aunando información que a cada área interese en particular.

Con el sistema ya no es repetitiva la información, además de ingresar la información, se puede compartir con las demás áreas.

Toda la información se tiene en una base, con lo cual se puede hacer cualquier tipo de consulta. Después de que las áreas de recepción hayan capturado los datos de las facturas, las demás áreas tendrán acceso a la información que soliciten.

3.1. Alcance del sistema

Uno de los principales alcances del sistema es automatizar los procesos que se llevan a cabo en el departamento de adquisiciones, para la captura y registro de las facturas provenientes de las dependencias por concepto de adquisiciones de material bibliográfico.

Con el nuevo sistema se unificó la información de las diferentes bases de datos que se manejaban en el departamento.

Unificando las bases se evitan los problemas que se presentaban, como la inconsistencia y redundancia de datos.

Los datos se van a manejar a través de la red interna de la DGB por lo cual se deben de implantar sistemas de seguridad, tanto en el servidor de bases de datos, como en el manejador de bases de datos, además de la seguridad propia del diseño de la aplicación.

En cuanto a la aplicación, la seguridad se implementó otorgando privilegios a cada área. De esta manera sólo se tiene acceso a la información correspondiente.

Cada área provee los datos, que en general, había estado capturando en sus sistemas. De esta manera se hace la integración de datos para diseñar la nueva base, además con el diseño estructurado de la base, se pueden generar cualquier tipo de informes.

A pesar de que la información se concentra en una base de datos, cada área sólo tiene acceso a la información que necesita y no más. De tal manera que el sistema incluye módulos de acceso por áreas.

Cada área cuenta con módulos de captura de datos, entre los cuáles se encuentran los módulos generales de datos y de catálogos. Adicionales a los módulos de captura están integrados el submódulo de informes y el submódulo de ayuda.

3. Análisis del Sistema

Aunque las áreas puedan compartir los datos, para cada área los submódulos son distintos, ya que tanto los datos que ingresan, los datos de sólo lectura y los informes presentan diferentes características.

El sistema reduce las tareas que llevaba a cabo manualmente cada proceso por área.

3.2. Recolección de requerimientos

Para hacer una análisis de requerimientos fue necesario entrevistar al personal de las áreas del departamento involucradas en el control y registro de facturas, obteniendo con ello un panorama de sus actividades para ser consideradas en la realización del nuevo sistema.

3.2.1. Áreas involucradas

El diagrama 3.1 muestra la organización de las áreas en el departamento.

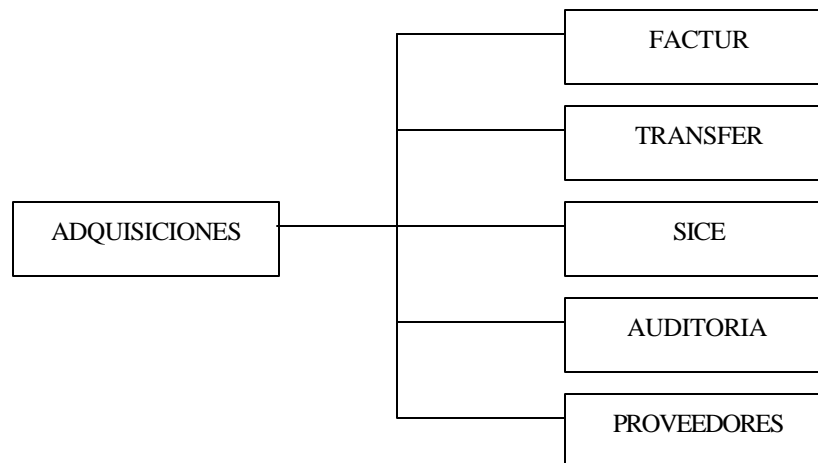


Diagrama 3.1 Área del Departamento de Adquisiciones

3.2.1.1. Facturación

Se encarga de hacer la captura de la información referente a las facturas por portada.

El proceso que se seguía en esta área era, recibir la copia de la factura por concepto de adquisición de material bibliográfico, proveniente de las dependencias de la DGB, posteriormente se le hacían una serie de anotaciones sobre la copia, para distinguir la información, y finalmente se hacía la captura de los datos de la factura en la base de datos local.

Estos datos eran almacenados en una base de datos de Microsoft Access, los cuales se encontraban en su mayoría en una sola tabla que, por supuesto, no estaba normalizada. La base sólo contaba con catálogos de BIBLIOTECAS y PROVEEDORES.

El sistema con el que se había estado trabajando contaba con siete módulos, de los cuales sólo tres eran funcionales. Dos para captura y uno para generación de informes.

Los datos que se ingresaban a la base de datos eran:

- Dentro del módulo de captura de facturas. Clave de la biblioteca (se toma de catálogo); clave del proveedor (se toma de catálogo); número de factura; fecha de factura; monto de la factura, el cual se coloca sólo en uno de tres campos asignados a pesos, dólares y francos, respectivamente; procedencia del material; fecha de recepción; nota de crédito, monto de la nota de crédito; año de presupuesto y observaciones.
- Dentro de módulo de Títulos y Volúmenes, básicamente se repetía la información de la factura, anexando el número de títulos y volúmenes por factura y el número de cargo y procesos por factura.

3.2.1.2. Transferencias

Se encarga de hacer la captura de la información referente a las facturas por transferencia. Es decir, las facturas cuya información bibliográfica de las adquisiciones llega por medio electrónico.

El proceso que se seguía en esta área era, recibir por ftp ⁰ los datos bibliográficos del material adquirido, y cuando llegaba la factura se hacía la captura de los datos en la base de datos local.

Estos datos eran almacenados en una base de datos de Microsoft Access, los cuales se encontraban en su mayoría en una sola tabla que no está normalizada. La base sólo contaba con catálogos de BIBLIOTECAS y PROVEEDORES.

El sistema con el que se había estado trabajando hasta ahora contaba con dos módulos, uno para captura y uno para generación de informes.

Los datos que se ingresaban a la base de datos eran:

- Dentro del módulo de captura de transferencias. Clave de la biblioteca (no se toma de catálogo, se ingresa); clave del proveedor (no se toma de catálogo, se ingresa); número de factura; fecha de recepción; año de presupuesto; número de cargos por facturas; número de títulos transferidos; número de títulos por factura; número de volúmenes; nombre del archivo; fecha técnico y observaciones.

⁰ File Transfer Protocol. Programa que se utiliza para transferir información, almacenada en ficheros, de una máquina remota a otra local, o viceversa

3.2.1.3. Sice

Se encarga de llevar la relación de códigos programáticos y presupuestos asignados a las dependencias y lleva el control administrativo de las facturas.

Sice trabajaba con hoja de cálculo de Microsoft Excel.

Los datos que se registraban en hojas del libro eran:

- Código programático; nombre de la dependencia; clave de la dependencia; año de presupuesto; asignación de presupuesto por trimestre; fecha de comprobación (cuando la factura pasa por sice); fecha de recepción de la factura; proveedor; número de facturas; monto de la factura; tipo de cambio en caso de llenar la columna que corresponde a moneda extranjera; tipo de moneda; número de títulos; número de volúmenes; observaciones y algunos campos calculados.

3.2.1.4. Auditoria

Se encarga de hacer visitas a las dependencias para comprobar el estado del material adquirido y reportarlo.

El proceso que se seguía en esta área era tomar la copia física de una factura al azar, seleccionar algunos títulos de la factura, en una hoja de cálculo de Microsoft Excel registrar tales títulos, hacer la auditoria a la dependencia correspondiente, hacer anotaciones sobre una hoja impresa de los registros de la hoja de Excel. Las anotaciones eran respecto al estado en que se encontraba el material.

Una vez que se tenía la información correspondiente a los títulos, se hacía la captura de los datos en la base de datos local.

Estos datos eran almacenados en una base de datos de Microsoft Access, los cuales se encontraban en su mayoría en una sola tabla que no está normalizada. La base sólo contaba con catálogos de BIBLIOTECAS y PROVEEDORES.

El sistema con el que se había estado trabajando hasta ahora contaba con dos módulos, uno para captura de títulos y otro para comprobación de títulos.

Los datos que se ingresaban a la base de datos eran:

- Dentro del módulo de captura de títulos. Clave de la biblioteca (se toma de catálogo); clasificación; número de adquisición; título; matriz; número de ejemplares; importe y observaciones.
- Dentro de módulo de comprobación de títulos. Clave de la biblioteca; clasificación; número de adquisición; título; matriz; número de edición; número de ejemplares; importe; frecuencia de uso; año de edición; fecha de comprobación; estado del libro, para el cual sólo se llena un campo, de cuatro campos a elegir y observaciones

3.2.1.5. Proveedores

Se encarga de hacer consultas WEB de los precios de un mismo material para diferentes proveedores, llevar una relación de los descuentos a la UNAM y hacer evaluaciones a los proveedores.

El proceso que se seguía en esta área era, tomar la copia física de una factura al azar, seleccionar todos los títulos de la factura; en una hoja de cálculo de Microsoft Excel registran el número de factura; el proveedor; la dependencia; el número de volúmenes por título; el costo por título sin descuento; costo del título con descuento; el precio de lista del país de origen, el cual se sondea en Internet; y en base a estos datos se hacen una serie de cálculos para obtener diferencias de precios y consolidados por proveedor.

Estos datos eran almacenados en un libro de Microsoft Excel.

Cabe destacar, que ésta área era una de las más desorganizadas, ya que no llevaban un estándar en las hojas de Excel para capturar su información. Además de que algunas personas de tal área, al querer facilitar su trabajo, trabajaban con una base de Microsoft Access, de la cual el resto de los integrantes del área no estaban enterados que existía.

Cada área seguía un proceso con fines específicos, pero el departamento llevaba un proceso general que involucra a todas las áreas. El proceso del departamento se ilustra en diagrama 3.2.

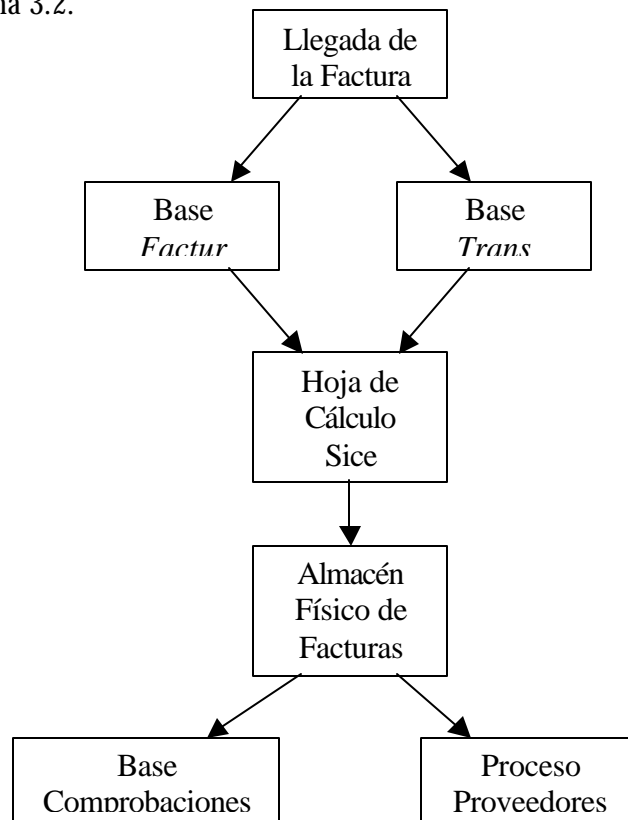


Diagrama 3.2 Proceso de Registro de Facturas

3.3. Resultados de las necesidades

La base de datos utilizada para cada semisistema, era manejada de manera local. Las bases entre sí no tenían ningún tipo de relación, a pesar de todas que las áreas trabajaban con los datos generales de las facturas, cada área manejaba la información como le era conveniente.

Los datos que manejaban las áreas en el departamento, no son muy variados, en su mayoría son los mismos y sólo los datos mínimos son los que cambian. De esta manera datos de un área pueden ser complemento de otra.

Dado que todas las áreas manejan en su mayoría la misma información, es posible que un área ingrese los datos en común de todas y posteriormente las demás áreas puedan visualizar esa información, anexando a la base de datos la información correspondiente de su área.

De esta manera se lleva control de la información en común, ya que era posible que para una misma factura, por error humano, se capturaran datos diferentes. Esto se vio reflejado cuando se hizo el comparativo de la información por área.

En general los datos a ingresar al sistema son:

- Datos de la Factura
 - Número de factura,
 - Fechas de factura,
 - Monto,
 - Moneda y Tipo de cambio,
 - Cantidad de títulos,
 - Cantidad de volúmenes
 - Descuentos,
 - Procedencia (dependencia), etc
- Datos del Proveedor
 - Razón social,
 - Procedencia (Nacional o Extranjero)
 - Tipo (Portada o Transferencia),
 - Estatus (Editorial o Distribuidor), etc
- Datos de la Dependencia
 - Clave de la dependencia,
 - Nombre,
 - Código programático asignado, etc
- Código Programático
 - Monto
 - Trimestre, etc

- Datos de los Usuarios
 - Calve de usuario,
 - Contraseña,
 - Área, etc
- Datos de Libros
 - Título,
 - ISBN,
 - Matriz,
 - Clasificación,
 - Número de adquisición, etc
- Relación de archivos que los proveedores por transferencia mandan por ftp.
 - Nombre,
 - Fecha de transferencia, etc

Los datos capturados por un área, podrán ser consultados por otras, inclusive, aunque el área no pertenezca al departamento.

Se está considerando la opción de incluir al Departamento de Procesos Técnicos, para que tenga acceso a consultar la información, ya que a tal departamento sólo le interesa la información recuperada de las facturas y no algún módulo de captura de datos, de esta manera también se evita el consumo de papelería y se puede obtener la información en cualquier momento.

Dado que todas las áreas trabajan en el módulo de captura de datos, son casi los mismos datos de una factura, por área se les da acceso sólo a la visualización de algunos de la factura y a la modificación de otros. Adicional a esto cada área cuenta con submodulos para capturar sólo la información propia de su área.

Dentro del módulo de catálogos, no todas las áreas tiene acceso a las opciones. Sólo algunas opciones están habilitadas para algunas áreas, de lo contrario se caería en problemas de control de datos e información. Por ejemplo, sólo el usuario administrador del sistema podrá ingresar usuarios, el área de facturación o transferencias no podrá ingresar datos de editoriales, dado que no es algo que tenga que ver con su área.

3.4. Recursos de Software y Hardware disponibles

Los sistemas hasta ahora utilizados por el departamento de adquisiciones, corren bajo plataforma Windows y desarrollados con herramientas de Microsoft, las cuales están demasiado limitadas en la mayoría de los casos, las aplicaciones fueron realizadas en:

- Microsoft Access
- Microsoft Excel

3. Análisis del Sistema

La Dirección General de Bibliotecas cuenta con herramientas variadas para el desarrollo de aplicaciones. Existen desde las más sencillas, como son las de Microsoft:

- Microsoft Access
- Microsoft Excel

y otras mucho más poderosas, como las herramientas de Sybase:

- Power Designer
- Power Builder
- Infomaker

Los recursos de hardware con los que se cuenta para proveer servicio de bases de datos:

- Procesador Pentium III
- Disco Duro de 10 GB
- 128 MB en RAM
- Sistema Operativo Windows 2000 Server

Para la elaboración de bases de datos se cuenta con manejadores de bases de datos corriendo en diferentes plataformas:

- Oracle bajo Solaris
- PostgreSQL bajo Linux
- SQLServer bajo Windows 2000 Server

Los recursos de hardware con los que se cuenta para instalar la aplicación en el Departamento de Adquisiciones, son máquinas las siguientes características:

- Procesador Pentium III
- Disco Duro de 10 GB
- 64 MB en RAM
- Sistema Operativo Windows Millenium

La DGB cuenta con la instalación de red LAN, la cual se encarga de realizar la transmisión de los requerimientos del cliente al servidor y del servidor al cliente. También controla la transmisión de datos entre los diferentes servidores que conforman el ambiente.

4. Análisis estructurado

4.1. Diagrama de Flujo de Datos

El Diagrama de Flujo de Datos (DFD) describe los flujos de información y los procesos o actividades que cambien o transforman los datos en un sistema. El diagrama de flujo de datos puede mostrar los procedimientos que debe seguir el personal de una organización para llevar a cabo ciertas tareas y así convertir los datos en información.

El diagrama de flujo de datos es una herramienta que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por líneas, círculos y rectángulos. Los diagramas de flujo de datos son gráficas dirigidas, en donde los nodos especifican las actividades de proceso y los arcos la transferencia de datos entre nodos de proceso.

4.1.1. Componentes de un DFD

- Proceso

El proceso muestra una parte del sistema que transforma entrada en salidas; es decir, muestra cómo es que una o más entradas se transforman en salidas. El proceso se representa gráficamente como un círculo. El proceso se describe con una sola palabra, frase u oración sencilla. El nombre del proceso describirá lo que hace. El proceso a veces describe quién o qué los está efectuando, más que describir el proceso mismo. La representación gráfica de un proceso, se muestra en el figura 4.1.



*SISTEMA DE CONTROL Y
REGISTRO DE FACTURAS*

Figura 4.1 Proceso

El sistema de adquisiciones lleva a cabo el proceso de control de facturas, por lo cual en el DDF se muestra como proceso.

- Flujo

Un flujo se representa gráficamente por medio de una flecha que entra o sale de un proceso. El flujo se usa para describir el movimiento de los bloques o paquetes de información de una parte del sistema a otra. En la mayoría de los sistemas, los flujos, representarán datos, es decir, bits, caracteres, mensajes, números de punto flotante, etc. La representación gráfica de un flujo, se muestra en el figura 4.2.



Figura 4.2 Flujo

Lo que entra a cada proceso del sistema son datos, en primera instancia, son los datos de la factura. Por lo cual los datos de la factura se muestran como flujo.

- Almacén

El almacén se utiliza para modelar una colección de paquetes de datos en reposo. Se denota por dos líneas paralelas. El nombre que se utiliza para identificar al almacén es el plural del que se utiliza para los paquetes que entran y salen del almacén por medio de flujos. La representación gráfica de un almacén, se muestra en el figura 4.3.



Figura 4.3 Almacén

Una vez ingresados los datos de las facturas se requieren para hacer una serie de consultas y se convierten en información para la factura, de aquí que se muestran como un almacén de datos.

- Terminador

Los terminadores representan entidades externas con las cuales el sistema se comunica. Comúnmente, un terminador es una persona, grupo o algún otro sistema computacional con el cual se comunica. Gráficamente se representa como un rectángulo. La representación gráfica de un terminador, se muestra en el figura 4.4.



Figura 4.4 Terminador

Una vez ingresados los datos de las facturas, la información es utilizada por las diversas áreas del departamento. De aquí que para el sistemas las áreas se muestran como terminadores.

Los Diagramas de Flujo de Datos para el *Sistema de Control y Registro de Facturas* se verán ilustrados en el capítulo siguiente.

4.2. Diagramas entidad – relación

El Diagrama Entidad – Relación (DER) es un modelo que describe la distribución de los datos almacenados en el sistema. Su principal propósito es representar los objetos de datos y sus relaciones.

4.2.1. Componentes del diagrama Entidad – Relación

4.2.1.1. Tipos de Objetos

Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos, pueden además ser identificados de manera única y ser descritos por uno o más atributos.

4.2.1.2. Relaciones

Son la serie de conexiones o asociaciones entre los tipos de objetos. La simbología utilizada para hacer la conexión entre los objetos de un diagrama, se muestran en la tabla 4.1.


Simbología	Existencia	Cardinalidad	Descripción
	Obligatoria	Uno	Debe existir uno y sólo uno
	Obligatoria	Muchos	Debe existir uno o más
	Opcional	Uno	Puede Existir uno o ninguno
	Opcional	Muchos	Puede Existir uno o más o ninguno

Tabla 4.1 Simbología de los terminadores de las relaciones

En el *Sistema Integral para el Departamento de Adquisiciones de la DGB de la UNAM* fue necesario utilizar diversos tipos de relaciones para el diseño de la base de datos. Algunas de las relaciones son las siguientes:

Debe existir un proveedor para cada factura.



FIG. 4.5 FK_FACTURAS_PROVEEDORES

Puede existir un presupuesto para cada factura.



FIG. 4.6 FK_FACTURAS_REFERENCE_PRESUPUE

Debe existir una dependencia para cada presupuesto.



FIG. 4.7 FK_PRESUPUE_REFERENCE_DEPEND

4.3. Diccionario de datos

“El diccionario de datos es un listado organizado de todos los datos pertinentes del sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de bases de datos y cálculos intermedios”¹

Los beneficios del diccionario de datos están relacionados con la recopilación, especificación y manejo efectivo de los recursos totales de datos de una empresa. Un diccionario de datos ayuda a controlar los campos de manera sencilla y efectiva, ayuda a reducir la redundancia e inconsistencia de datos, ayuda a centralizar el control de los campos, ayuda en el diseño y en la expansión del diseño de la base de datos.

Se puede hacer referencia al diccionario de datos del *Sistema de Control y Registro de Facturas* en el apéndice C de este trabajo.

4.4. Arquitectura cliente – servidor

Esta arquitectura consta de un cliente inteligente que puede solicitar servicios de un servidor en red.

En el lado del cliente de esta arquitectura se encuentra una aplicación frontal bastante sencilla ejecutándose en un ordenador personal. A una aplicación cliente / servidor se le puede pedir que realice validaciones o que muestre listas de opciones válidas, pero la mayor parte de las reglas de integridad de los datos y de negocio se imponen en la propia base de datos: relaciones, índices, valores predeterminados, rangos, disparadores, procedimientos almacenados, etc.

En el lado del servidor se encuentra un motor de servidor de bases de datos inteligente. El servidor está diseñado para aceptar consultas SQL desde la aplicación frontal, generalmente en forma de llamadas a procedimientos almacenados que devuelven conjunto de resultados claramente definidos y de ámbito limitado.

Generalmente, la aplicación cliente es responsable, al menos, de la administración de la conexión, la captura de los datos, la presentación de datos y la administración de los errores.

El servidor es el responsable de la administración inteligente de los recursos, la administración de la seguridad, la administración de los datos, de las consultas y sobre todo de la integridad de los datos.

4.4.1. Dos Capas

Esta arquitectura propone centralizar los procesos, como control de acceso, conexión a las bases de datos, ejecución de funciones, etc. La figura 4.8 muestra esta arquitectura.

¹ Análisis Estructurado Moderno, Yourdon Edwar

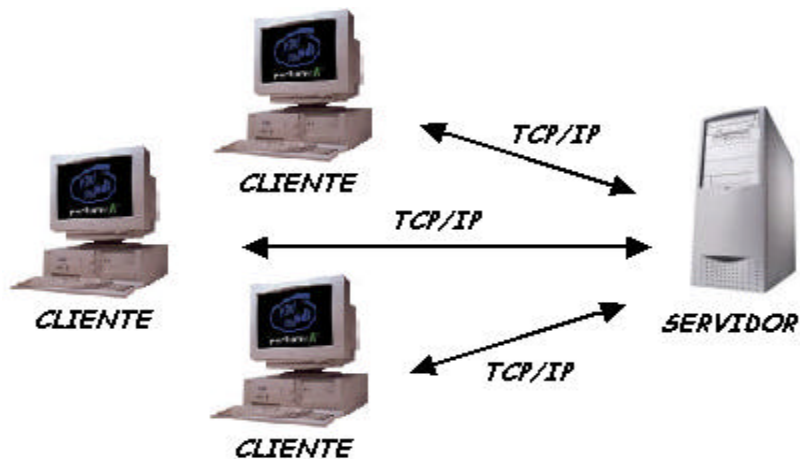


Figura 4.8 Arquitectura Cliente – Servidor de Dos Capas

En ésta arquitectura el servidor recibe las peticiones de los cliente y se encarga de establecer las conexiones necesarias con la base de datos (tantas como clientes existan), siempre y cuando el cliente este autorizado (autenticación).

Ya establecida la conexión, los clientes a través del Front End (interfaces que permiten interactuar al usuario con el sistema), los procesos solicitan y reciben y despliegan los datos extraídos de la base de datos en el servidor.

Por lo tanto, el servidor debe estar construido por los siguientes módulos:

- Listener (Receptor de Peticiones). Se encarga de escuchar a través de un puerto a los clientes que requieren conectarse, éstos envían su solicitud y el servidor establece una conexión dedicada con ellos.
- Autenticación. El cliente envía información de usuario y contraseña de tal manera que se pueda validar el usuario; de ser así se inicia la sesión.
- Administración de Recursos. Ya iniciada la sesión, el servidor es el responsable de proveer los datos que requieren los procesos que el cliente está ejecutando, hasta que finalmente el cliente termina la sesión.

Ventajas:

- Esta arquitectura puede ser implantada en ambientes locales, Intranet e Internet.
- Permite centralizar el almacenamiento de datos.
- Maneja auditorias de usuarios.

Desventajas:

- El proceso de autenticación depende exclusivamente del servidor de bases de datos.
- El procesamiento de datos se realiza directamente en el servidor.

4.4.2. Tres capas

Esta arquitectura conocida como tres capas: cliente, servidor de aplicaciones y servidor de bases de datos, permiten que las capas sean especializadas en funciones; de tal manera que provee a la arquitectura los módulos: Cliente, Servidor de Aplicaciones y Servidor de Bases de Datos. La figura 4.9 muestra esta arquitectura.

- Cliente. Es el más ligero, ya que sólo se encarga de construir y enviar peticiones al servidor a través de métodos con parámetros, es decir no hay procesamiento en el cliente.
Recibe los resultados de los procesos solicitados y los despliega.
- Servidor de Aplicaciones. Se encarga de la autenticación de los usuarios y de la identificación de sus perfiles.
Ejecuta los procesos solicitados a través de los métodos enviados por los clientes.
Resuelve las conexiones con la base de datos, a través de un spool de conexiones, identificando si es necesario abrir o cerrar sesiones, dependiendo de la demanda a la base de datos por los procesos que se están ejecutando en el servidor.
- Servidor de Bases de Datos. Atiende peticiones de inserción, actualización, eliminación y consulta de datos para procesos específicos.

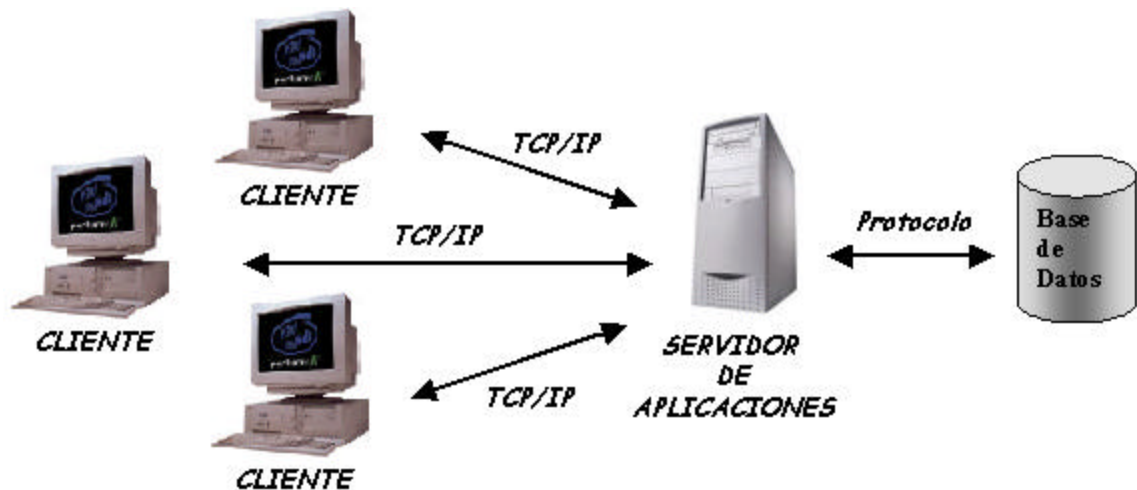


Figura 4.9 Arquitectura Cliente – Servidor de Tres Capas

Ventajas:

- A diferencia de la arquitectura de dos capas, en este caso no resulta tan sencillo violar las reglas del sistema, ya que se tiene dos procesos de autenticación: uno en el servidor de aplicaciones y otro en el servidor de bases de datos.

Dada la separación explícita de funciones, se reduce la posibilidad de que algún cliente pudiera causar daño al sistema.

- La ejecución de todos los procesos es del lado del servidor, dando acceso al sistema a clientes cuyos recursos sean óptimos. Los requerimientos de hardware y software se concentran en el servidor.
- El servidor de aplicaciones y el de bases de datos pueden instalarse en distintas máquinas.

Desventajas:

- El monitoreo y la administración del sistema es complejo, debido al incremento de puntos de falla derivado del número de componentes.
- El desarrollo de la infraestructura del sistema (arquitectura) es más sofisticada.

Para el Sistema de Control y Registro de Facturas, la arquitectura utilizada fue la de Dos Capas.

4.5. Funciones del servidor y funciones del cliente

4.5.1. Servidor

El servidor es el encargado de satisfacer los requerimientos del Cliente. Procesa las consultas del Cliente, envía, recibe y almacena información, provee seguridad y control de acceso.

4.5.2. Cliente

Es el elemento encargado de interactuar directamente con el usuario final. Mediante éste, el usuario realiza el acceso a la información sin importar el lugar en donde se encuentre, el cliente maneja la presentación de los datos, realiza la captura y la validación de los mismos, genera consultas, ejecuta operaciones y recibe información procedente del servidor o de otro cliente.

El cliente debe contar con una gran capacidad de procesamiento, y debe poseer una interfaz amigable para el usuario final.

Una interfaz gráfica de usuarios (GUI) es la ideal para un cliente, ya que le permite realizar operaciones complejas mediante labores sencillas como oprimir botones, los cuales están ubicados en la pantalla gráfica.

4.6. Estimación de Costos

La estimación del costo de un sistema no es algo relativamente fácil, más aún si se desea hacer la estimación de los primeros sistemas para un desarrollador.

La estimación de costos inicia en la etapa de planeación del proyecto. La estimación de costos finaliza en la etapa de revisión preliminar del diseño del proyecto, debido a que se van a ir descubriendo factores que quizá no se habían contemplado inicialmente.

Los aspectos más importantes a considerar en la estimación de costos son:

- Personal
- Tiempo
- Técnicas utilizadas.

Dentro de los aspectos generales a considerar, se cuenta con algunas características particulares para cada uno. Algunas de las características son:

- Capacidad del Programador.
- Complejidad del producto.
- Tamaño del programa. Si un sistema es complejo, generando con ello gran cantidad de programación, evidentemente es más costoso.
- Tiempo disponible.
- Confiabilidad requerida.
- Nivel tecnológico. En el nivel tecnológico entran las herramientas de desarrollo. Como parte del desarrollo tecnológico es necesario contar con personal capacitado para el desarrollo del sistema, ya que de lo contrario, la productividad se afecta si el personal requiere de aprender a utilizar otra herramienta como parte del proceso de desarrollo.

4.6.1. Técnicas de Estimación de Costos

4.6.1.1. Juicio Experto

Es la técnica más utilizada para la estimación de costos. Se base en las experiencias con sistemas anteriores, haciendo comparaciones de estos, se basa también en el juicio comercial del grupo de trabajo.

4.6.1.2. Estructuras de División de Trabajo

En esta técnica se hacen divisiones del trabajo, para verlo como partes de un sistema, haciendo una jerarquía de productos y procesos.

4.6.1.3. Estimación del Nivel de Contratación

En base a la cantidad de personal requerido para desarrollar el sistema.

4.6.1.4. Estimación de Costos de Mantenimiento

Cuesta mas mantener el software que desarrollarlo. Para sistemas con una larga vida, este costo se multiplica.

4.7. Herramientas de Control de Avance

Las herramientas de control de avance nos sirven para administrar un proyecto. Entre las técnicas existentes están el Método de la Ruta Crítica (MRC) y el método de Evaluación de Programas y Técnicas de Revisión (PERT).

Estos métodos consisten en elaborar un diagrama de actividades que muestra la dependencia de cada actividad sobre otra, en la que se tiene que desarrollar en función del tiempo, costo y/o de los recursos utilizados.

4.7.1. Método de la Ruta Crítica (MRC)

El proceso de informatización de la administración de proyectos se inició en los años 50. DuPont Corporation y Remington Rand, en un esfuerzo por mejorar las técnicas de programación de proyectos, desarrollaron un sistema de programación llamado Método de Ruta Crítica: un modelo matemático que calcula la duración total de un proyecto basándose en la duración de cada tarea en particular y en sus dependencias, y que identifica qué tareas son críticas. Este modelo es el método de programación utilizado fundamentalmente en el software para la administración de proyectos actualmente, incluido Microsoft Project.

Este método utiliza términos como:

- Actividad. Es la parte individual de trabajo que hay que efectuar en un sistema. Es un trabajo único con una duración determinada.
- Evento. Es el punto de partida de una actividad y sucede sólo cuando todas las actividades que le preceden han llegado a su término.
- Red. Es el conjunto de actividades y eventos que reflejan el sistema.
- Actividad Virtual. Es una actividad que dura un tiempo igual a cero.
- Tiempo libre u holgura. Es el tiempo que existe entre el final de una actividad y el principio de la siguiente.

4.7.2. Programme evaluation and Review Technique (PERT)

Durante los años 50, la Marina de EE.UU. desarrolló el proyecto de los misiles Polaris, un sistema de armamentos basado en submarinos. Lockheed, el contratista principal del proyecto, creó el sistema de programación PERT (Programme Evaluation and Review Technique) que utiliza probabilidades estadísticas para calcular las duraciones previstas.

La diferencia de PERT con MRC, es que el tiempo de cada actividad es variable y se determina de una forma probabilística. Para estimar el tiempo de cada actividad atribuye tres estimaciones de tiempo diferentes:

- Tiempo pesimista (p)
- Tiempo más probable (m)
- Tiempo optimista (o)
- Tiempo esperado (e)

El tiempo esperado se obtiene con el siguiente cálculo: $e = (o + 4m + p)/6$

Hoy en día, un diagrama PERT (también llamado diagrama de red) hace referencia a la representación gráfica de las relaciones entre tareas.

El método del camino crítico, llamado también con la sigla PERT aplica principios de la teoría de los gráficos. Esta teoría realiza el estudio de procesos, estructuras y sistemas mediante gráficos. Se denomina “gráfico” a un conjunto de puntos llamados “nodos” y a un conjunto de arcos que vinculan esos puntos. Los procesos o proyectos a programar se representan mediante estos gráficos; cada tarea es simbolizada por un nodo o por un arco. Establecidos los tiempos que corresponden a las tareas, el método busca determinar la secuencia de arcos adyacentes (denominada camino) que, yendo desde el principio al final del proceso o proyecto, tenga mayor duración. Este es denominado “camino crítico” e interesa conocerlo, pues condicionará la duración total del proceso o proyecto: por ello, las tareas comprendidas en ese camino (“tareas críticas”) deben ser las más controladas y en caso de desear acelerarlo, habrá que comenzar por ellas.

Este método es utilizado, por lo tanto, para aquellos procesos o proyectos que tienen un comienzo y una terminación (es decir que no son continuos) y que presentan cierta complejidad en la interrelación de las tareas que los forman.

4.7.3. Diagrama de GANTT

En un desarrollo independiente de sistemas de administración de proyectos, Henry L. Gantt desarrolló un modo de representar gráficamente las actividades a lo largo de una escala de tiempo.

El gráfico de Gantt cubre las distintas funciones de la programación: asignación de recursos y determinación de la secuencia y cronología. A pesar de su sencillez constituye uno de los métodos de programación más completos y, sin duda, el más usado. El gráfico se confecciona a dos escalas: en la horizontal se mide el tiempo (en horas, días, semanas, etc); en la vertical, se ordenan los elementos que intervienen en la programación: máquinas, hombres, tareas, órdenes de trabajo, etc.

El gráfico establece, de tal forma, una relación cronológica entre cada elemento productor o tarea. Las subdivisiones horizontales del espacio en el gráfico representan a la vez tres cosas: transcurso de una unidad de tiempo, trabajo programado para ese intervalo y trabajo realizado efectivamente en ese lapso. La inclusión simultánea de estos dos últimos aspectos (generalmente se lo hace mediante segmentos trazados en distintos colores o de diferente textura o forma) implica cotejar lo programado con lo realizado, es decir controlar lo programado.

Para la administración del sistema, se podrá utilizar el Diagrama de Gantt de Microsoft Project para crear una programación, así como hacer el seguimiento e imprimir informes.

Microsoft Project 2000 es una aplicación de Microsoft que ayuda al usuario a crear planes de proyectos, comunicarlos a otros usuarios y adaptarse a los cambios a medida que éstos se van produciendo. Es un sistema de planificación de proyectos fácil de utilizar.

Microsoft Project funciona, en mucho sentidos, en forma similar a otras aplicaciones de Microsoft. Las barras de menús, los comandos, las barras de herramientas, los menús

5. Desarrollo

5.1. Diseño

5.1.1. Diagrama Jerárquico Funcional

El diagrama jerárquico funcional tiene dos objetivos.

- Entendimiento del sistema
- Definir la estructura de las actividades del sistema tan simple como sea posible.

5.1.1.1. Convenciones para dibujar un diagrama Jerárquico Funcional

- Dibujar cada función en una caja
- Usar texto normal para definir la descripción de la función
- Empezar la descripción de la función con un verbo y tratar de usar una frase explicativa.
- Si se desea descomponer la función, dibujar tres asteriscos en la parte superior derecha de la caja
- Mostrar la referencia de la función sobre o al lado izquierdo de la caja.
- Dibujar las funciones comunes con una línea vertical a un lado de la descripción de la función dentro de la caja.
- Mostrar la etiqueta de identificación de la función maestra en la parte derecha superior de la caja en cada una de las funciones copias.

El diagrama 5.1 muestra el Diagrama Jerárquico Funcional Nivel 1 para el Sistema de Control y Registro de Facturas.

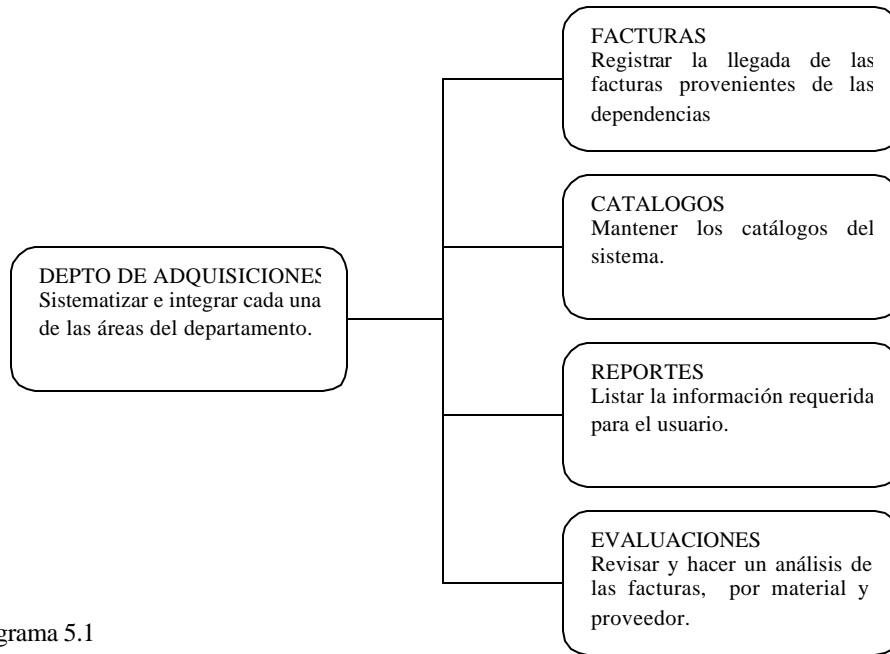


Diagrama 5.1
Diagrama Jerárquico Funcional en Nivel 1

El diagrama 5.2 muestra el Diagrama Jerárquico Funcional Nivel 2 para el Sistema de Control y Registro de Facturas.

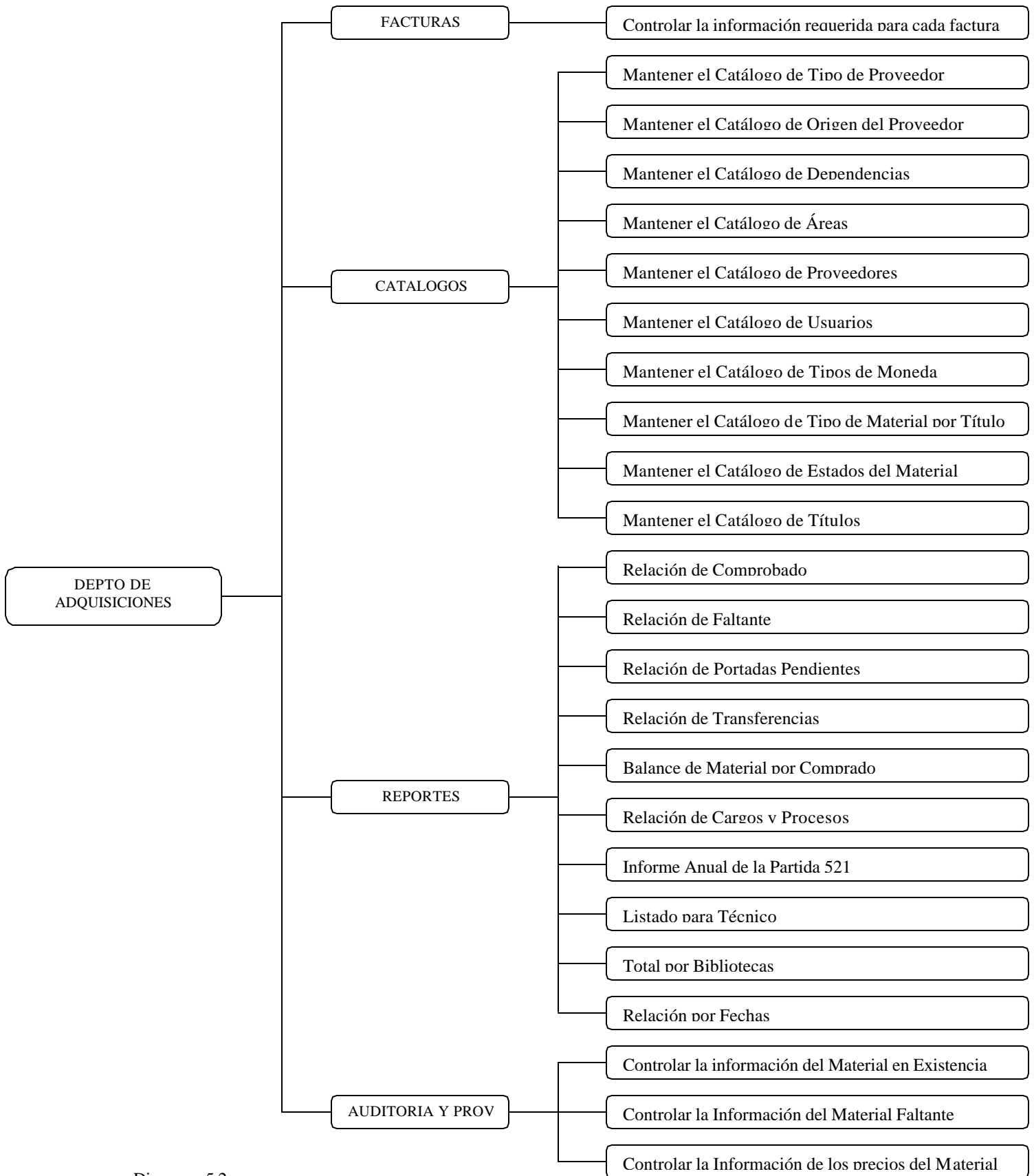


Diagrama 5.2
Diagrama Jerárquico Funcional en Nivel 2

5.1.2. Diagrama de Flujo de Datos

Los Diagramas de flujo de datos, especifican un proceso indicando las entradas y las salidas de datos. Representan procesos manuales o automáticos.

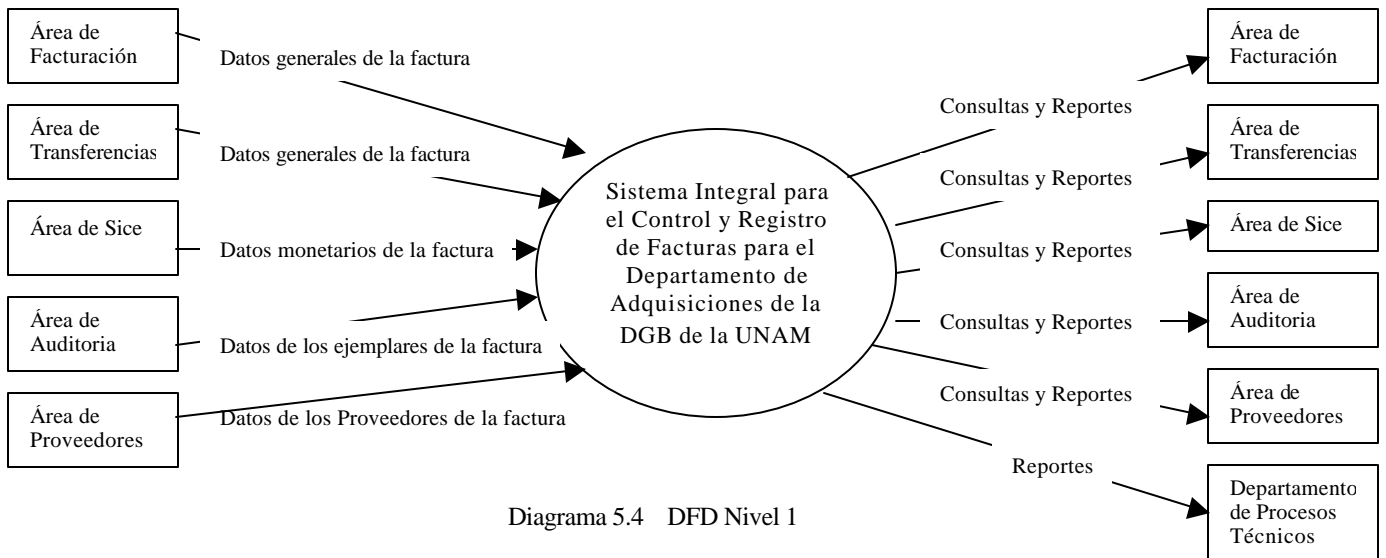
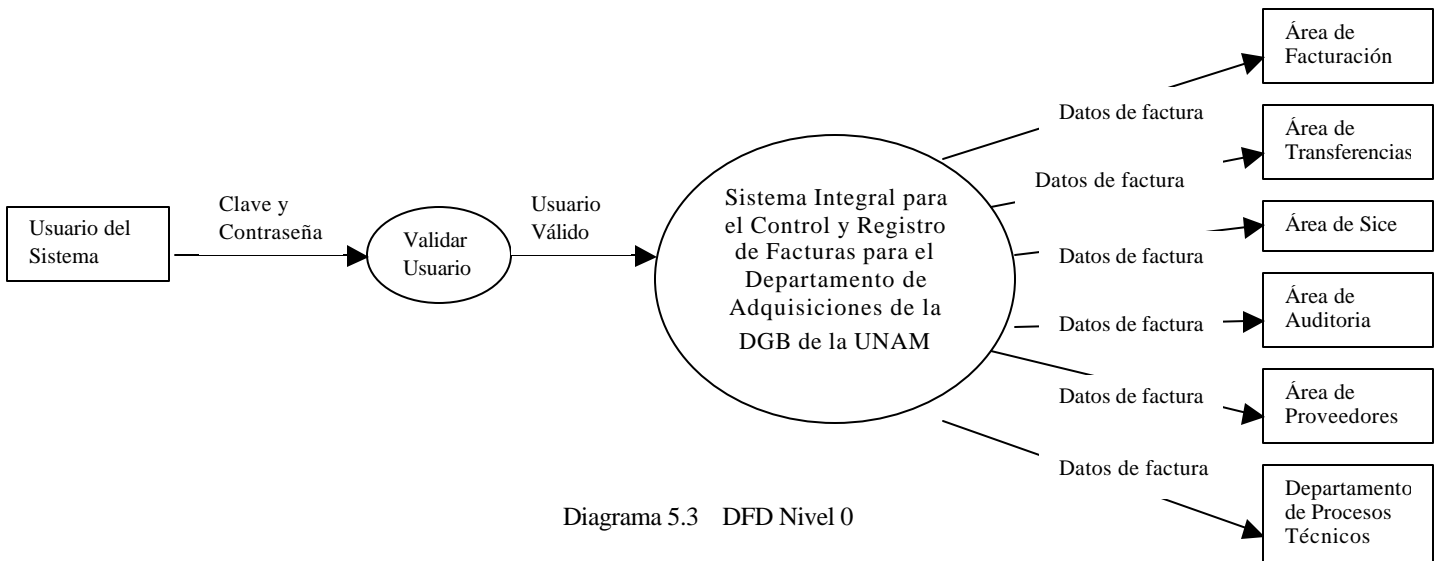
Los Diagramas de Flujo de datos son la representación gráfica de los orígenes y destinos de un sistema de información mediante símbolos plenamente definidos.

El objetivo es mostrar como fluye la información en el sistema, suministrando a las funciones los datos requeridos.

Los Diagramas de Flujo de Datos son de dos tipos o niveles:

5.1.2.1. Diagrama de Flujo de Datos de Nivel 0 o Conceptual. Diagrama 5.3

5.1.2.2. Diagrama de Flujo de Datos de Nivel 1 o Detalle. Diagrama 5.4



5. Desarrollo

5.1.3. Diagrama Entidad – Relación

Para el diseño de la Base de Datos se cuentan con múltiples relaciones entre las entidades establecidas en la etapa de diseño. Las relaciones del diseño y su cardinalidad se muestran en la tabla 5.1.

Relación Tabla1	Tabla 2	Nombre	Cardinalidad	“Tabla1” a	“Tabla2” a	“Tabla1” “Tabla1”
	FACTURASXARCHIVO	rel_arch_factxarch	Uno Muchos	a		Dependiente
	FACTURASXARCHIVO	rel_fact_factxarch	Uno Muchos	a		Dependiente
	PRECIOS	Rel_fact_precios	Uno Muchos	a		Dependiente
	LIBROS	rel_fact_lib	Uno Muchos	a		Dependiente
	LIBROS	rel_tit_libros	Uno Muchos	a		Dependiente
	FACTURAS	rel_usr_fact	Uno Muchos	a		
	FACTURAS	rel_pres_fact	Uno Muchos	a		
	FACTURAS	rel_prov_fact	Uno Muchos	a		Dependiente
	FACTURAS	rel_tipomon_fact	Uno Muchos	a		
	TITULOS	rel_tipomat_tit	Uno Muchos	a		
	TITULOS	rel_ed_tit	Uno Muchos	a		
	USUARIOS	rel_area_usr	Uno Muchos	a		Obligatoria
	PRESUPUESTOSXDEP	rel_dep_pres	Uno Muchos	a		Obligatoria



Tabla 5.1 Relaciones

5.1.4. Diseño de la BD

Durante el diseño de la base de datos se planea cómo las entidades y los atributos del sistema llegarán a ser tablas y columnas. Empezando con un diseño conceptual normalizado, representado por un modelo de entidad, se puede producir un diseño lógico inicial de la base de datos mostrado en la figura 5.1 y finalmente el modelo físico, mostrado en la figura 5.2.

Pasos para el Diseño de la Base de Datos

- Escoger las llaves primarias
- Convertir entidades en tablas
- Convertir atributos en columnas
- Convertir relaciones en llaves foráneas
- Escoger candidatos a índices

Modelo Conceptual

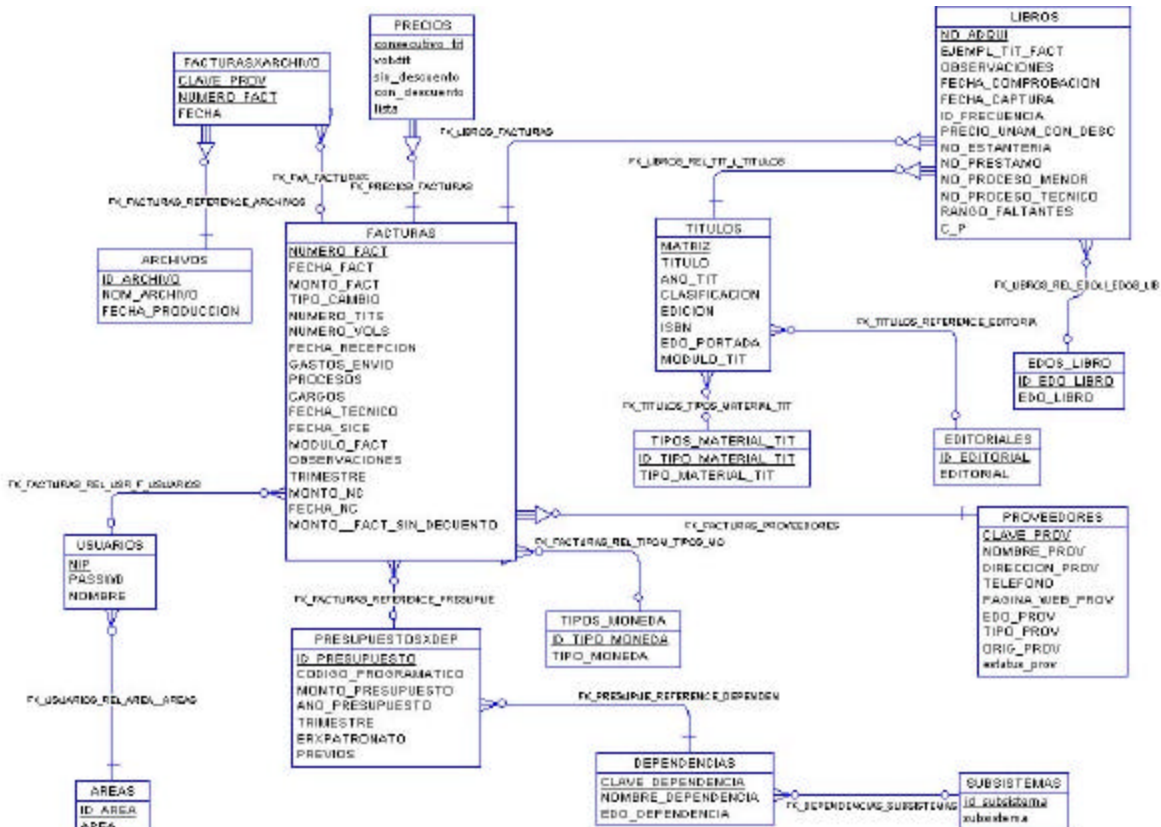


Figura 5.1 Modelo Conceptual

Modelo Físico

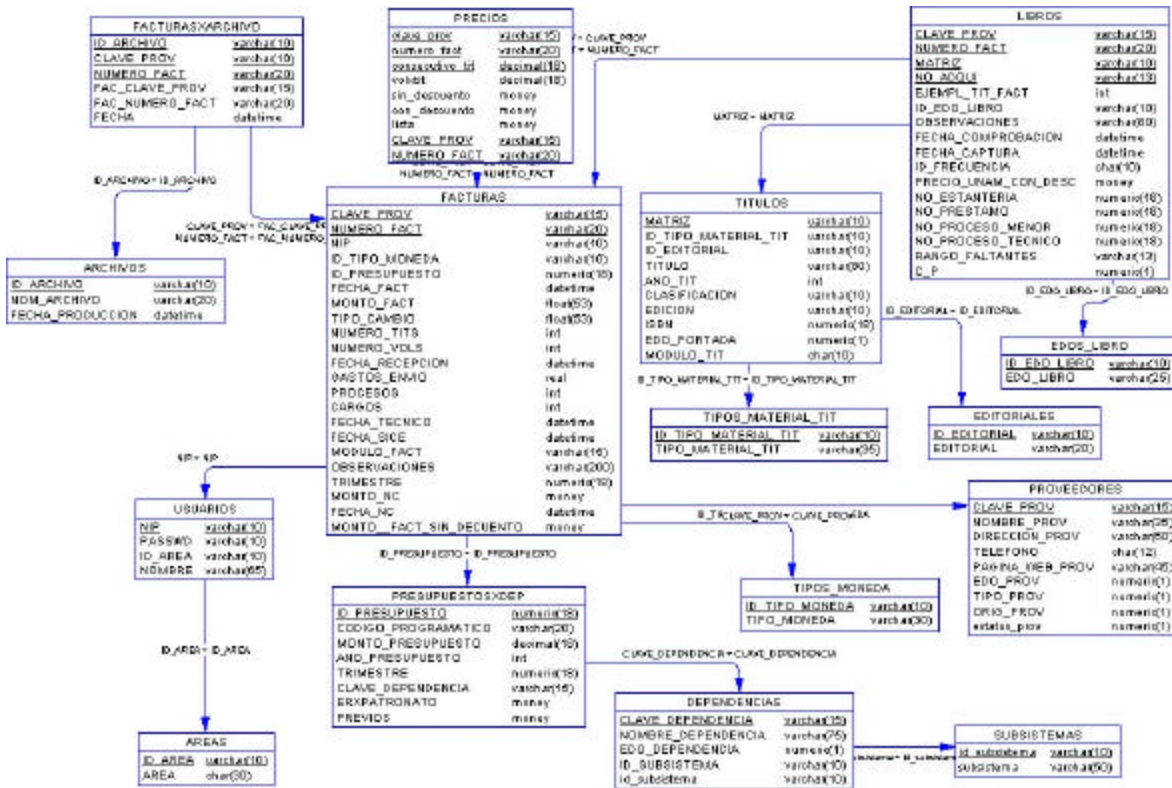


Figura 5.2 Modelo Físico

5.1.4.1. Tablas de la Base de Datos

El área de transferencias lleva un control de los archivos que envía el proveedor por factura, para ello es necesaria la tabla FACTURASXARCHIVO. Esta tabla sólo tiene atributo de fecha_ingreso. Esta tabla y sus relaciones se justifican, ya que con ella se está rompiendo una relación de muchos a muchos. Tabla 5.2

FACTURASXARCHIVO

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Fecha_ingreso	Date			Si

Tabla 5.2 facturasxarchivo

Cada proveedor, en relación con otros, tiene establecidos precios diferentes para material que vende por título, además de que a un título le corresponden tres diferentes precios: sin descuento a la UNAM, con descuento a la UNAM y el precio de lista del proveedor. Los precios es necesario registrarlos para el área de proveedores. Al registrar el precio de un título por factura, no es necesario registrar propiamente el título del ejemplar, basta con asignar un número consecutivo y de esta manera se sabe a que título se hace referencia en la factura. La

5. Desarrollo

relación con la tabla FACTURAS se justifica de esta manera, ya que basta saber el número de la factura y el proveedor. Tabla 5.3

PRECIOS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
consecutivo_tit	Decimal(18)	Si	Si	Si
volxtit	Decimal(18)			Si
Unam_sin_desc	Money			Si
con_desc	Money			Si
Lista_sin_descuento	Money			Si
Unam_porcentaje_desc	Money			Si
Lista_porcentaje_desc	Money			Si
Usr_captura	VarChar(55)			Si

Tabla 5.3 precios

Esta tabla esta relacionada con la tabla TITULOS, de la cual se toma la matriz para hacer referencia al título, pero tiene campos adicionales que sirven para identificar por ejemplar. El campo principal es el número de adquisición (no_adqui), el cual es único y por tal, este campo también es la llave primaria de la tabla. Tabla 5.4

LIBROS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
no_adqui	Varchar(20)	Si	Si	Si
observaciones	Varchar(80)			Si
fecha_comprobación	Date			Si
fecha_captura	Date			Si
id_frecuencia	Char(10)			Si
precio_UNAM_con_desc	Money			Si
No_estantería	Decimal(18)			Si
No_prestamo	Decimal(18)			Si
No_proceso_menor	Decimal(18)			Si
No_proceso_técnico	Decimal(18)			Si

Tabla 5.4 libros

Al área de transferencias le llegan archivos con los datos de los títulos por factura, de los cuales hay que llevar una relación, para lo cual es necesaria la tabla de ARCHIVOS. Tabla 5.5

ARCHIVOS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Id_archivo	VarChar(10)	Si	Si	Si
Nom_archivo	VarChar(20)			Si
Fecha_producción	date			Si

Tabla 5.5 archivos

En la tabla FACTURAS están contenidos los datos generales de la factura, es decir, aquí se pueden encontrar la mayoría de la información que interesa a las área. Tabla 5.6

FACTURAS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Numero_fact	VarChar(20)	Si	Si	Si
Nip	Varchar(15)			Si
Fecha_fact	Integer			Si
Monto_fact	LongFloat			Si
Tipo_cambio	Float			Si
Numero_tits	Integer			Si
Numero_vols	Integer			Si
Fecha_recepción	Integer			Si
Gastos_envio	Float			Si
Procesos	Integer			Si
Cargos	Integer			Si
Fecha_tecnico	Integer			Si
Fecha_sice	Integer			Si
Modulo_fact	Char(10)			Si
Observaciones	VarChar(200)			Si
Trimeste	Decimal(18)			Si
Monto_nc	Moneda			Si
Fecha_nc	Date			Si
Monto_fact_sin_descuento	Flota			Si
Suma_cuxtit	Money			Si
Suma_cuxtit_desc	Real			Si
Clave_dependencia2	Varchar(15)			Si
Fecha_evaluación	DateTime			Si
Periodo_evaluación	Varchar(60)			Si

Tabla 5.6 facturas

La tabla TITULOS contiene la información de los títulos de las muestras que toma de las facturas el área de auditoría. Tabla 5.7

TITULOS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Id_tit	Char(12)	Si	Si	Si
Matriz	VarChar(10)		Si	Si
Titulo	VarChar(80)		Si	Si
Año_tit	Integer			Si
Clasificacion	VarChar(10)			Si
Edición	VarChar(10)			Si
Isbn	Integer			Si
Edo_portada	Number(1)			Si
Modulo_tit	Char(10)			Si
Autor	Char(10)			

Tabla 5.7 titulos

5. Desarrollo

Las facturas pueden llegar con monto en moneda nacional, dólares o francos. La tabla TIPOS_MONEDA lleva la relación de los posible tipos de moneda, para asignarlo a la factura y de esta manera calcular la conversión a moneda nacional. Tabla 5.8

TIPOS_MONEDA

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
id_tipo_moneda	VarChar(10)	Si	Si	Si
tipo_moneda	VarCahr(30)			Si

Tabla 5.8 tipos_moneda

Dado que el sistema va a ser usado por varias áreas, es necesario llevar una relación de usuarios para dar acceso y establecer permisos en el sistema, para ello se utiliza la tabla de USUARIOS. Tabla 5.9

USUARIOS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Nip	Char(10)	Si	Si	Si
Passwd	VarChar(10)	Si	Si	Si
Nombre	VarChar(65)		Si	Si

Tabla 5.9 usuarios

El área de sice lleva el control de los códigos programáticos asignados a las dependencias. Cada código tiene asignado un monto y año de presupuesto, los cuales forman parte de los atributos propios de la tabla, anexos a el atributo clave_dependencia tomado de la relación con la tabla DEPENDENCIAS. Tabla 5.10

PRESUPUESTOSXDEP

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Id_presupuesto	Serial(18)	Si	Si	si
codigo_programatico	VarChar(20)		Si	Si
monto_presupuesto	LongFloat		Si	Si
año_presupuesto	Integer		Si	Si
trimestre	Number(18)		Si	Si
erxpatronato	Money			Si
Previos	Money			Si

Tabla 5.10 presupuestosxdep

Cada factura tiene proviene de un proveedor, el cual, es tomado de esta tabla para asignarlo a la factura. Tabla 5.11

PROVEEDORES

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
clave_prov	Char(10)	Si	Si	Si
nombre_prov	VarChar(35)		Si	Si
direccion_prov	VarChar(60)			Si
Telefono	Char(12)			Si
pagina_web_prov	VarChar(45)			Si

5. Desarrollo

edo_prov	Number(1)			Si
tipo_prov	Numero(1)			Si
orig_prov	Numero(1)			Si
estatus_prov	Numero(1)			Si

Tabla 5.11 proveedores

Para normalizar la tabla de TITULOS se genera la tabla EDITORIALES que hace referencia por título. Tabla 5.12

EDITORIALES

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
id_editorial	VarChar(10)	Si	Si	Si
Editorial	VarChar(20)			Si

Tabla 5.12 editoriales

La tabla AREAS contiene los registros de todas las áreas que tienen acceso al sistema. Cada usuario tiene asignada un área, la cual es tomada de la relación con esta tabla. Tabla 5.13

AREAS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
ID_AREA	Char(10)	Si	Si	Si
AREA	Char(30)			Si

Tabla 5.13 area

La tabla DEPENDENCIAS contiene a las dependencias de la DGB que generan facturas. Tabla 5.14.

DEPENDENCIAS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
clave_dependencia	Char(10)	Si	Si	Si
nombre_dependencia	VarChar(75)			Si
edo_dependencia	Number(1)			Si

Tabla 5.14 dependencias

Cada dependencia debe estar clasificada por subsistema, para lo cual se genero la tabla SUBSISTEMAS, que tiene relación con la tabla DEPENDENCIAS. Tabla 5.15

SUBSISTEMAS

Nombre	Tipo de Dato	Identificador	Obligatorio	Mostrar
Id_subsistema	VarChar(10)	Si	Si	Si
Subsistema	VarChar(50)			Si

Tabla 5.15 subsistemas

5.2. Manejadores de BD

Para el desarrollo del sistema es necesario hacer un análisis de las herramientas de desarrollo con las que se cuenta, tanto con licencia como libres, e incluso considerar la posibilidad de adquirir una.

Una vez hecho el análisis se deben de establecer las herramientas con las que se va a trabajar.

5.2.1. Diseño de la Base de Datos

5.2.1.1. Selección y justificación de las herramientas de modelado

Sybase

Power Designer. Es un conjunto de herramientas de modelado de software que son accesibles, permitiendo crear aplicaciones sólidas.

Es un producto para la automatización del análisis, diseño, generación, mantenimiento y recuperación de aplicaciones de bases de datos. Soporte de DBMS heterogéneos (Sybase, Oracle, IBM, Informix, Microsoft, etc.) en ambientes abiertos y mixtos. Almacenar información orientada a objetos en un formato relacional (Modelado de objetos - relacional).

Diseña y genera componentes de negocio que se puedan instalar en la base de datos (procedimientos almacenados, tipos de datos complejos), en un servidor de aplicaciones o en un cliente pesado.

5.2.1.2. Selección y justificación del manejador de DB

Microsoft.

Debido a los recursos con los que cuenta el Departamento de Adquisiciones y los recursos que la DGB brinda para desarrollar este sistema, el manejador de bases de datos a utilizar es SQL Server.

Se planteó, e incluso se hicieron algunas pruebas con manejadores como PostgreSQL.

SQL Server soporta la recuperación de datos que hace el departamento y en conjunto con la aplicación, resulta de velocidad considerable.

5.3. Diseño de la Aplicación

5.3.1.1. Selección y justificación de las herramientas para generar la interfaz del usuario

La herramienta seleccionada en esta etapa es Power Builder, de Sybase, la cual en comparación con Visual Basic o Delphi hace posible la recuperación de datos en gran cantidad mediante un ODBC (Open DataBase Connectivity) sin que se bloquee la aplicación. Además de tener diversos controles que facilitan la creación

de acceso a datos como el llamado DataWindow. La configuración del ODBC para una aplicación se indica en el apéndice A.

5.3.1.2. Interfaz del usuario

La interfaz del usuario es el medio por el cual el usuario se comunicará con la computadora, está formado por pantallas.

Se deben desarrollar interfaces que mejoren la velocidad de captura de los datos y reduzcan los errores.

La primer pantalla que muestra el sistema es la de validación de usuario. En la pantalla de acceso el usuario debe ingresar su nombre de usuario y contraseña. Como los usuarios previamente fueron dados de alta por el administrador, el sistema reconoce al área a la que pertenece y da acceso al módulo correspondiente. Figura 5.3

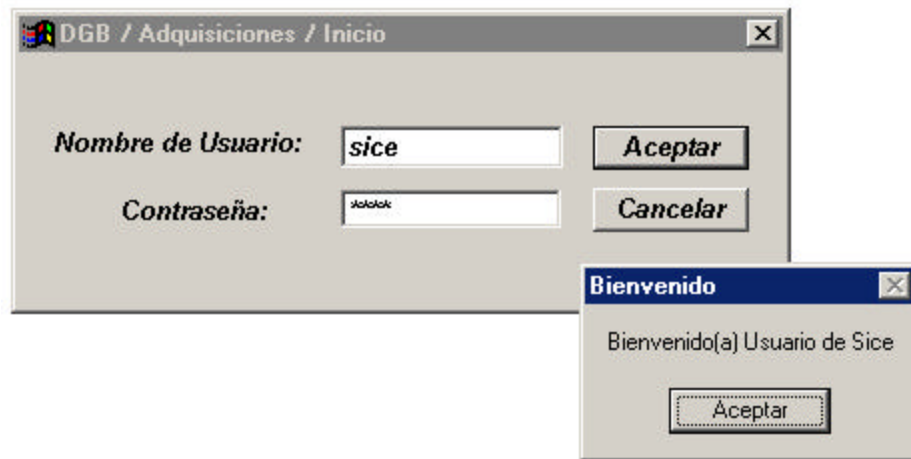


Figura 5.3 ventana de inicio (validación usuario)

Una vez dentro del sistema el usuario podrá ver la ventana principal, la cual contiene diversos menús que le permiten moverse dentro del sistema. Figura 5.4

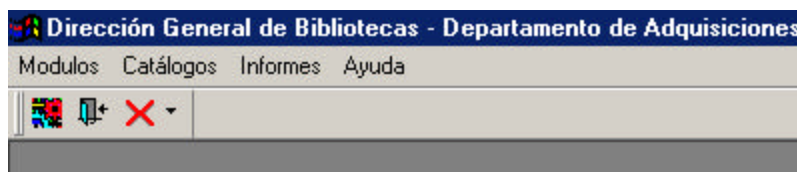


Figura 5.4 barra de menús

5.3.1.3. Módulos

5.3.1.3.1. Facturas

En este módulo aparece un submódulo “facturas” el cual utilizará el área de Facturación y el área de Transferencias, ya que básicamente es la misma información la que se captura, haciendo la diferencia el tipo de adquisición, que es por portada o transferencia. En este módulo el Área de Sice sólo asigna el código Programático a la factura. Figura 5.5

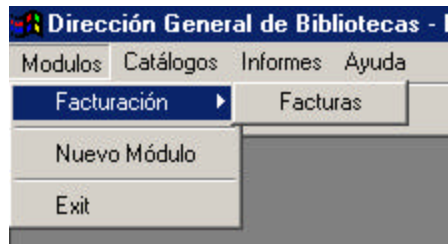


Figura 5.5 módulos de facturación

Dentro del submódulo de facturas aparece una ventana (figura 5.6) en la cual se van a ingresar los datos correspondientes a la factura dependiendo del área a la que pertenece el usuario que ingresó al sistema.

Los datos que aquí se capturan son:

- Proveedor (Catálogo)
- Fecha de factura
- Número de factura
- Número de Títulos
- Número de Volúmenes
- Número de Cargos
- Número de Procesos
- Número de Transferencias (Sólo área de Transferencias)
- Fecha de Recepción
- Fecha en se va al Departamento de Técnico
- Fecha de Comprobación
- Monto de la Factura con Descuento
- Monto de la Factura sin Descuento
- Gastos de Envío (en caso de haberlos)
- Tipo de Moneda con laque se facturó (Catálogo)
- Tipo de cambio (en caso de ser moneda extranjera)
- Monto y Fecha de la Nota de Crédito (en caso de haberla)
- Código Programático (Catálogo)
- Observaciones

Campos calculados

- Costo Unitario Promedio por Títulos
- Costo Unitario Promedio por Volúmenes

Campos que se llenan automáticamente.

- Usuario
- Área a la que pertenece el usuario
- Dependencia correspondiente al código programático (sólo para sice)

Por medio del tipo de proveedor se sabe el tipo de factura.

Figura 5.6 ventana de captura de facturas

5.3.1.3.2. Transferencias

El módulo de Transferencias contiene tres submodulos (figura 5.7), de los cuales, el primero ya ha sido descrito.



Figura 5.7 módulos de transferencias

5.3.1.3.2.1. Archivos

En el submodulo “archivos” (figura 5.8) se ingresan los datos de los archivos que los proveedores han hecho llegar al departamento de adquisiciones.

Clave de archivo:

Nombre de archivo:

Fecha Produccion [mm/dd/aaaa]:

Figura 5.8 ventana de captura de archivos

Los datos a capturar en esta ventana son:

- Clave de archivo
- Nombre de Archivo
- Fecha Producción

5.3.1.3.2.2. Facturas Por Archivos

Para llevar un control de os archivos con los que se cuenta y los respectivos datos de facturas contenidas en estos, se cuenta en el submodulo de facturas x archivo.

Hay que hacer una selección del archivo y posteriormente ir ingresando las facturas. Figura 5.9

FACTURAS POR ARCHIVO

Clave Arch	Clave Proy	Numero Fact
BNA02	BNAME	66774V
BNA02	BNAME	71684V
BNA02	BNAME	66764V
BNA02	BNAME	64388V
BNA02	BNAME	71682V
BNA02	BNAME	71683V
BNA02	BNAME	60003V
BNA02	BNAME	64389V
BNA02	BNAME	66773V

Numero Factura	Clave Proveedor
000M	AJUSTE
000FA	AJUSTE
000AE	AJUSTE
000AE/2	AJUSTE
000AE/3	AJUSTE
000AE/4	AJUSTE
000AE/5	AJUSTE
000CH	AJUSTE
000CH/2	AJUSTE
000CH/3	AJUSTE
000CH/4	AJUSTE
000EU	AJUSTE

Figura 5.9 ventana de captura de facturas por archivo

5.3.1.3.3. Sice

El módulo de Sice cuenta con dos submódulos (figura 5.10), el primero es para llevar la relación de códigos programáticos y el segundo es el descrito dentro del módulo de facturación, es decir, es el submódulo de “facturas”.

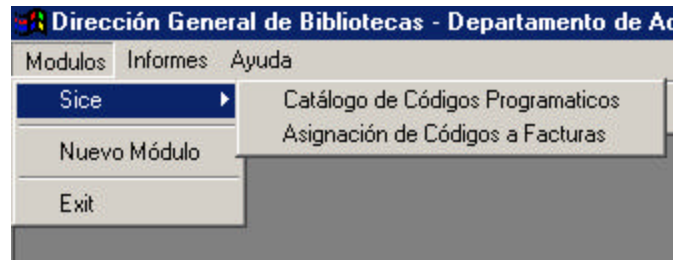


Figura 5.10 módulos de sice

5.3.1.3.3.1. Catálogo de Códigos Programáticos

Dentro de este submódulo aparece una ventana en la cual se lleva la relación de códigos programáticos que maneja el departamento. Figura 5.11

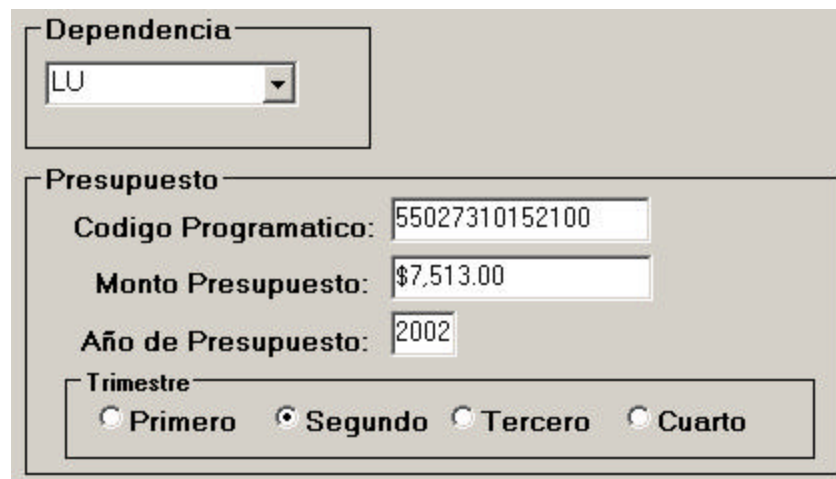


Figura 5.11 ventana de captura de presupuesto

Los datos a ingresar son:

- Clave de dependencia
- Código Programático
- Monto del presupuesto asignado al trimestre
- Año de presupuesto
- Trimestre del año de presupuesto

5.3.1.3.4. Proveedores

El módulo de proveedores cuenta con un submódulo (figura 5.12) para llevar la relación de precios de material.

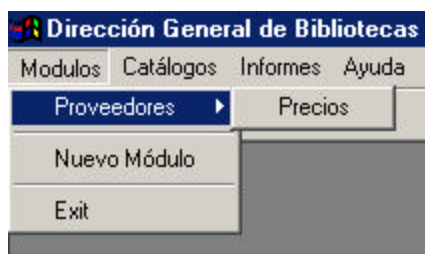


Figura 5.12 módulos de proveedores

5.3.1.3.4.1. Precios

Antes de abrir la ventana de captura para precios por proveedor, hay que seleccionar la factura de cual se va a hacer la evaluación. La pantalla de selección la muestra la figura 5.13.

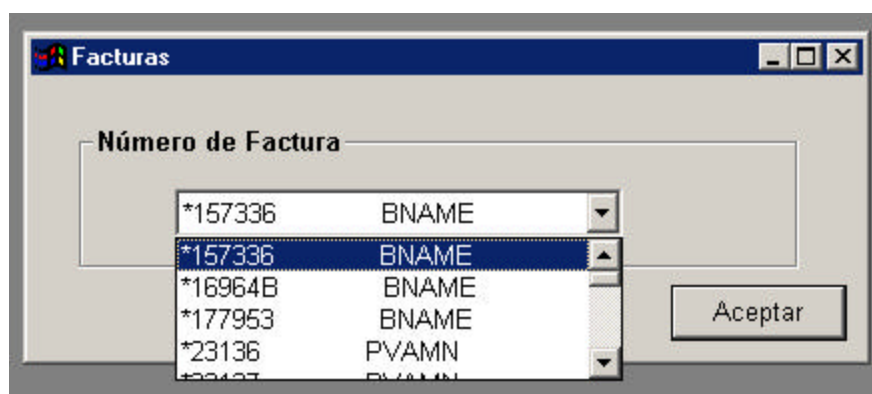


Figura 5.13 selección de facturas

La pantalla anterior da acceso a la pantalla donde se van a registrar los precios. La pantalla de registro es la que aparece en la figura 5.14, donde los campos de Clave y Factura se llenan automáticamente.

EVALUACIÓN DE PRECIOS DEL PROVEEDOR:				
CLAVE:		<input type="text"/>	FACTURA:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Título	Volúmenes	Precio a UNAM Sin Descuento	Precio a UNAM Con Descuento	Precio de Lista del País de Org

Figura 5.14 captura de precios

5.3.1.3.5. Auditorias

El módulo de auditorias cuenta con dos submodulos (figura 5.15), el primero para hacer la captura de los datos del título de la muestra seleccionada (figura 5.16) y el segundo para hacer la comprobación (figura 5.17).

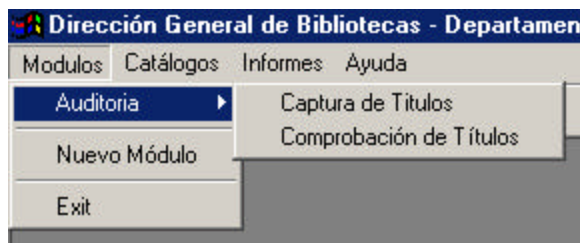


Figura 5.15 módulos de auditorias

5.3.1.3.5.1. Captura de Títulos

A screenshot of a data entry window titled "Captura de Títulos". The window has a light gray background. It contains several input fields and dropdown menus: "Matriz:" with a text box; "Título:" with a wide text box; "Tipo Material:" with a dropdown menu; "Clasificación:" with a text box; "Editorial:" with a dropdown menu; "Edición:" with a text box; "Isbn:" with a text box; and "Año de Publicación:" with a text box.

Figura 5.16 ventana de captura de títulos

5.3.1.3.5.2. Comprobación de Títulos

Antes de abrir la ventana de captura (Fig. 5.17) para datos de ejemplares, hay que seleccionar la factura de cual se va a hacer la auditoria (Fig. 5.13).

A screenshot of a data entry window titled "Comprobación de Títulos". The window has a light gray background and is divided into several sections. The top left section is "Datos de la Factura" and contains fields for "Proveedor:" (value: #ALCMN), "Factura:" (value: 24593), and "Fecha:" (value: 23/08/1999). The top right section is "Año Presupuestal" with a value of 2002. Below that is the "Fechas" section with "Comprobación:" (value: 00/00/0000) and "Auditoria:" (value: 00/00/0000). The middle section contains "Matriz:" (value: 12313), "No de Adquisición:" (value: 13221), "Ejemplares" (value: 003/012), and "Edo Libro:" (value: Faltante). The bottom right section is "Frecuencia de Uso" with three radio buttons: "Mucho", "Regular" (which is selected), and "Nada". At the bottom, there is an "Observaciones:" text box and an "Importe:" field with the value \$257.00.

Figura 5.17 ventana de captura de datos de comprobación de títulos

5.3.1.4. Catálogos

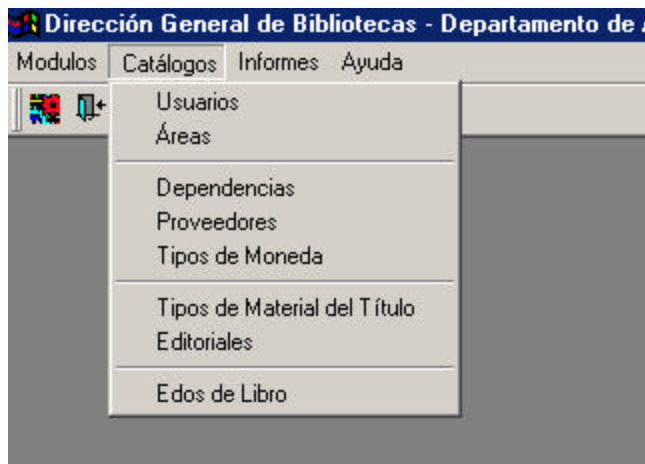


Figura 5.18 menú de catálogos

- Usuarios
- Áreas
- Dependencias
- Proveedores
- Tipos de Moneda
- Tipos del Material del Título
- Editoriales
- Estados del Libro

5.3.1.5. Informes

Para algunos informes, es necesario ingresar algún parámetro para la recuperación de la información.

Estos parámetros son ingresados en una ventana que se abre antes de generar el reporte, figura 5.19. El número de parámetros varía dependiendo de lo refinado que se quiera el reporte.



Figura 5.19 argumentos de informes

En el módulo de Sice, por ejemplo, se reporta la estadística por código programático y para generar tal es necesario ingresar el código.

Informes del módulo de Sice:

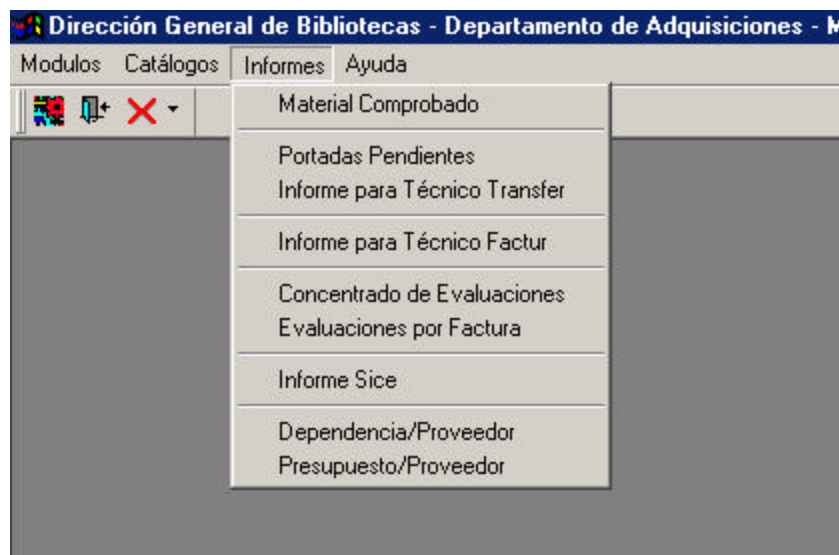


Figura 5.20 menú de informes

- Material Comprobado
- Portadas Pendientes
- Informe para Técnico de Facturaciones
- Informe para Técnico de Transferencias
- Concentrado de Evaluaciones
- Evaluaciones por Factura
- Informe Sice (figura 5.21)
- Dependencia / Proveedor
- Proveedor / Factura



Figura 5.22 ventana base

Dentro del cuerpo de la pantalla, se encuentran los botones con diversas funciones:

- Moverse entre Registros. Mueve los registros según se especifique, mostrando en una pequeña ventana el número de registro.
 - Primero. Mueve al primer registro
 - Anterior. Mueve al registro anterior del actual
 - Siguiente. mueve al registro siguiente del actual
 - Último. Mueve al último registro
- Modificar. Modifica un registro ya existente, es decir, modifica los datos del registro.
- Agregar. Inserta un registro dejando en blanco sus campos para ser llenado con los datos correspondientes.
- Eliminar. Elimina el registro actual
- Buscar. Busca un registro dependiendo del campo especificado para hacer la búsqueda.
- Guardar. Una vez que se agrego el registro y se hizo la captura de los datos, el botón de actualizar es necesario para graba los datos, de lo contrario no se verán reflejados en la base.

- Cancelar. Cancela el registro agregado o los cambio realizados.
- Cerrar. Cierra la ventana

Las pantallas deben atraer al usuario y mantener su atención, esto se logra con el uso de espacios abiertos que rodean los campos de captura de datos. Nunca se debe saturar una pantalla.

Las pantallas muestran las formas de captura y los reportes, es decir, son el medio por el cual se desplegará la información

5.3.3. Informes

Son los reportes de salida que requieran los usuarios. Al diseñar un informe, se deben incorporar elementos tanto funcionales como estéticos, de tal forma que el informe proporcione al usuario la información que requiere en un formato legible.

Atributos:

- Encabezado
- Título del informe
- Número de página
- Fecha del informe
- Encabezados de las columnas
- Agrupamiento de los datos

Objetivos:

- Diseñar un reporte para satisfacer el objetivo planteado
- Diseñar un reporte que se adapte al usuario
- Provee la cantidad adecuada de información
- Asegura que el reporte este disponible donde se necesite
- Proporciona oportunamente el reporte

Los informes son variados, dependen de las necesidades de los usuarios de cada área.

5.3.3.1. Selección y justificación de la herramienta

InfoMaker

Es un producto de Sybase. Mediante una interfaz similar a la de Power Builder, InfoMaker permite diseñar reportes por medio de consultas.

5.3.4. Ayuda

Help Workshop

Los archivos de ayuda se generan con el software Help Workshop 4.0¹ de Microsoft que corre bajo plataforma Windows.

Help Workshop es un programa que se usa para crear archivos de ayuda *.hlp. A los archivos de ayuda por, lo general, los distingue un icono que muestra un libro con un signo de interrogación (figura 5.23).



Figura 5.23 icono de ayuda

Help Workshop toma la información de un archivo *.hpl para combinarlo con un archivo *.rtf, mapas de bits, etc. El ejecutable de Help Workshop se llama *hcw.exe*.

Este programa es útil para mostrar al usuario la ayuda (figura 5.24) de la aplicación de manera agradable y de fácil acceso, como en cualquier otra aplicación basada en gráficos, así como lo son las aplicaciones de Microsoft.

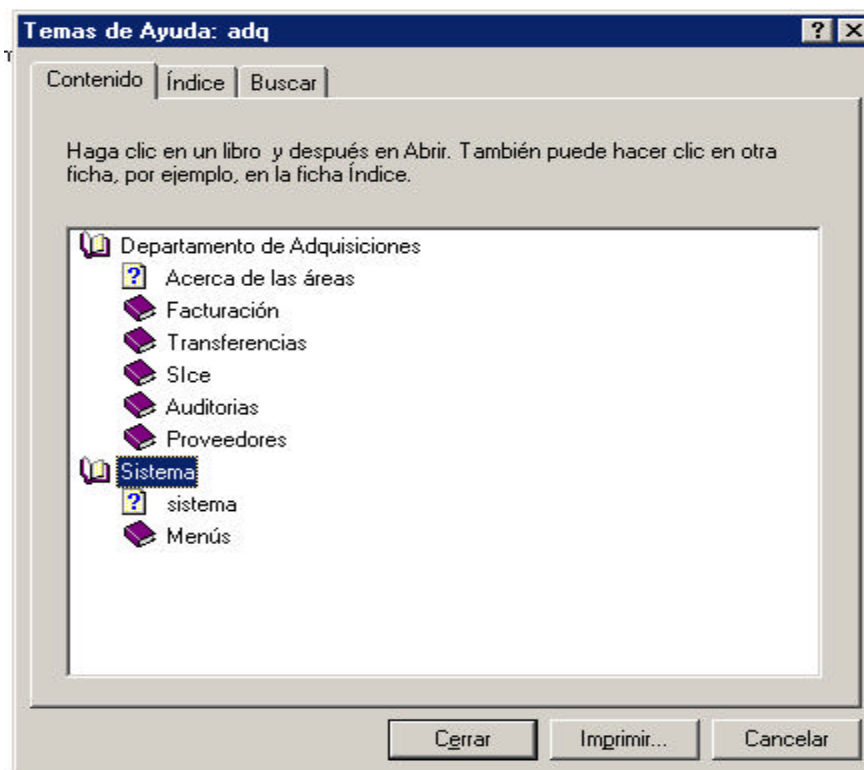


Figura 5.24 ventana de ayuda

¹ http://www.dcs.napier.ac.uk/hci/VB5/HTML/tutorial_16.html

6. Liberación

6.1. Revisiones Técnicas Formales

La calidad de un sistema inicia desde la etapa de análisis y con las especificaciones del usuario. Las revisiones técnicas formales son herramientas que auxilian a la calidad del sistema.

Las Revisiones Técnicas Formales tienen como objetivo encontrar problemas en la calidad que presente el sistema y garantizar ésta.

Si se cuenta con un considerable número de personal para el análisis y desarrollo del sistema, se podría asignar a una persona, que no estuviera muy dentro de las cuestiones técnicas, para que se dedique a hacer las Revisiones Técnicas Formales. De esta manera se evitaría que los desarrolladores pasen por alto algunos detalles.

Objetivos de las RTF:

- Detectan errores en las funciones, en la lógica o en la implementación del sistema.
- Verificar que el sistema cumple con los requerimientos.
- Garantizar que el sistema desarrollado se realizó de acuerdo con los estándares establecidos.
- Conseguir que el desarrollo del sistema sea uniforme
- Lograr que los proyectos sean manejables.

Restricciones:

- Debe convocarse a la junta de revisión entre tres y cinco personas.
- Preparar con tiempo la reunión.
- La duración de la reunión debe ser en un lapso menor de dos horas

Se hacen revisiones por cada módulo o un pequeño grupo de módulos.

Para este sistema, las Revisiones Técnicas Formales se llevaron a cabo por parte del personal del departamento de adquisiciones.

Registro e informe de revisión.

Durante las reuniones de la RTF, el encargado registra todas las modificaciones que surjan durante la revisión. Al final todas las modificaciones se resumen y se realiza una lista con todos los acuerdos a los que se llegó. Además, se prepara un resumen de la reunión de revisión. Dicho informe responde a las preguntas básicas:

- ¿Qué fue revisado?
- ¿Quién lo revisó?

- ¿Qué se descubrió y cuáles son las conclusiones?
- ¿Cuáles son las áreas problemáticas?
- ¿Cuáles son los puntos de acción para realizar las correcciones?

Factores que contribuyen en la calidad del sistema:

- **Corrección.** Es el grado en que un programa realiza satisfactoriamente las especificaciones establecida por los clientes.
- **Fiabilidad.** El grado en que se espera que los resultados del programa realicen sus funciones con la precisión requerida por el cliente.
- **Eficiencia.** La cantidad de recursos de computadora y de código requeridos por el programa para llevar a cabo las funciones.
- **Integridad.** El grado en que se controla el acceso al sistema o a los datos por personal no autorizado.
- **Facilidad de uso.** El esfuerzo requerido para aprender, trabajar, preparar la entrada e interpretar la salida de un programa.
- **Facilidad de Mantenimiento.** El esfuerzo para localizar y arreglar un error en un programa
- **Facilidad de Prueba.** El esfuerzo para probar un programa de forma que se asegure que realiza las funciones requeridas.
- **Portabilidad.** El esfuerzo necesario para transferir el programa desde un hardware y/o entorno de sistemas a otro.
- **Reusabilidad.** El grado en que un programa se pueden reutilizar en otras aplicaciones.
- **Facilidad de operación.** Facilidad para operar el programa
- **Modularidad.** Independencia funcional de los componentes del programa.
- **Seguridad.** La disponibilidad para controlar o proteger los programas y la información.

Para mantener la seguridad de la información se establecen políticas generales y controles que previenen o detectan cualquier intento de acceso no autorizada a un sistema de computo.

Para evitar riesgos en la seguridad de la información, se debe considerar lo siguiente:

- ✓ Proteger los datos de cualquier uso indebido
- ✓ Definir y asignar niveles de control de acceso
- ✓ Monitorear los sistemas para detectar cualquier uso indebido.

Para el área de Proveedores fue un poco complicado visualizar el nuevo proceso, ya que estaban acostumbrados a un proceso más complejo. Al ver que el nuevo proceso es más sencillo sentían que no cubría sus necesidades.

Para el área de facturación, transferencias y sice sólo se hicieron observaciones en cuanto a la restricción de campos y formato de estos.

6.2. Pruebas del Sistema

6.2.1. Módulo

Esta prueba se lleva a cabo para cada módulo del sistema de manera individual. *“La prueba de unidad consiste básicamente en probar la interfaz del módulo para asegurar que la información fluye de forma adecuada hacia y desde la unidad del programa que está siendo probada. Se examinan las estructuras de datos locales para asegurar que los pasos que se mantiene temporalmente, conserven su integridad durante los pasos de ejecución del algoritmo. Se prueban las condiciones límite, para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento. Se ejercita todos los caminos independientes de la estructura de control, con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez. Y finalmente, se prueban todos los caminos de manejo de errores.”*¹

En el punto anterior se han descrito las problemáticas encontradas para cada área. De ahí se partió para hacer las correcciones y hacer pruebas sobre estas.

6.2.2. Integral

Para esta prueba se trabaja con la interacción de varios módulos, es decir, se trabaja moviéndose entre módulos para ver cual es el comportamiento del sistema.

En este caso, el módulo de facturación y transferencias son muy similares, de ahí que esta prueba se pudo realizar con una persona que maneja los dos módulos simultáneamente, obteniendo resultados satisfactorios.

6.2.3. General

La prueba general engloba todo el sistema, ya que incluye software y hardware. Dentro de esta prueba se valida la conexión a la base de datos y la recuperación de información de manera simultánea.

Para esta prueba se solicitó que todo el personal trabajara en conjunto, es decir, todo el personal de las áreas trabajó al mismo tiempo. Con esto se comprobó la capacidad del servidor y de la propia aplicación, soportando múltiples conexiones.

6.3. Manual del usuario

El manual de usuario es una guía cuyo propósito es instruir al usuario sobre el uso del sistema, dirigiéndolo desde la entrada del sistema hasta la operación de sus opciones.

En el manual, se describe como se puede realizar la operación deseada, que opciones y parámetros tienen disponibles, que valores tiene que tomar para elegir la opción seleccionada, así como condiciones de error y puntos importantes dentro de cada subsistema.

¹ Ingeniería de Software, Roger S. Pressman

La guía de usuario necesita ser construida de tal forma que los temas sean presentados con una estructura más fácil de entender al usuario.

Características.

- Interfaz gráfica que habilite al usuario para seleccionar una lista de temas, en la cual se puede ir a detalle.
- Implementar definiciones por medio de contexto sensitivo.
- Utilización de saltos a definiciones.
- Implementación de botones para el control de saltos entre temas.

Dentro del manual de usuario no se debe considerar demasiada información técnica, debido a que al usuario sólo le interesan los detalles y funciones del sistema.

6.4. Manual de Operación

El manual de operación es una guía, no para el usuario, sino para los desarrolladores del sistema.

Este manual incluye detalles técnicos acerca del diseño de sistema, acerca de la manera en que opera el sistema.

Un manual bien elaborado ayuda en la etapa de mantenimiento, ya que de esta manera será fácil ubicar el origen del problema.

Documentos asociados al manual:

- Documentos que son requeridos en el desarrollo del sistema.
- Documentos que describan la arquitectura del sistema
- Para cada componente, una especificación y una descripción
- Listados del código fuente de los programas, estos deben estar comentados apropiadamente, donde se explique la función de cada línea de código
- Documentos de validación en donde se describa cómo la información se valida.

6.5. Capacitación

La capacitación para cada área del Departamento de Adquisiciones, es por parte del Departamento de Análisis, Desarrollo y Mantenimiento de Sistemas. De esta manera, la capacitación se hace de manera personalizada para poder aclarar todas las inquietudes por parte de los usuarios. Además se entregó el respectivo manual de usuario.

6.6. Mantenimiento

En cuanto al mantenimiento de la base de datos, no hay mayor problema, porque se estarán haciendo respaldos, se depurará de la información no necesaria o errónea e incluso se pueden hacer modificaciones a los datos.

Si posterior a la liberación del sistema, el Departamento de Adquisiciones desea hacer algún cambio en la aplicación, es decir dar o negar accesos, crear más módulos o generar más informes, por ejemplo, es posible hacerlo ya que la programación y la generación de informes tienen posibilidades de crecimiento, el único inconveniente es que se tendría que hacer la actualización del archivo *.exe de la nueva versión en todas las máquinas en donde se haya instalado el sistema.

Dentro de la etapa de mantenimiento se encuentran los respaldos de la información, los cuales se realizan cada semana.

Para sistemas grandes los manejadores de bases de datos habituales corren bajo Linux o Solaris, para los cuales el hardware utilizado para respaldo son unidades de cintas magnéticas.

Debido a las características de software con el que está implementada la Base de Datos del Sistema Integral del Departamento de Adquisiciones, el respaldo de información se hace con unidades de CDR/W.

Las unidades de CDR/W pueden ser de dos tipos: Internas o Externas, sólo deben de cumplir con los requerimientos del sistema operativo donde esta montada la base.

Windows 2000 Server no soporta cualquier unidad de CDR/W, debido a que Windows trabaja con múltiples interrupciones, las cuales en ocasiones provocan ruido en la escritura de información, ya que esta lee y escribe datos usando un haz láser. Para Windows 2000, ya sea en versión Server o Profesional, se tiene que adaptar una unidad que sea compatible con el Sistema Operativo.

Las herramientas SQL Server, de las cuales nos podemos apoyar para el mantenimiento a la Base de Datos son:

- “Mantenimiento”. Esta herramienta proporciona un asistente que nos guiara en la creación de un plan de mantenimiento a la base, el cual ejecutará SQL Server. Con este plan de mantenimiento se puede:
 - Ejecutar comprobaciones de integridad de base de datos.
 - Actualizar estadísticas de la base de datos
 - Realizar copias de seguridad de la base de datos.

- “Copia de Seguridad de la Base de Datos”.
 - Copia de Seguridad de Base de Datos. Esta opción nos ayuda en la creación, mediante un asistente de SQL Server, de un archivo de respaldo de la base. Los archivos de respaldo los deja en un directorio específico, por lo tanto, es el directorio que se va a bajar a Disco Compacto.
 - Restaurar la Base de Datos. Esta opción permite hacer la recuperación de los datos de la copia de seguridad.

6.7. Seguridad

6.7.1. Seguridad en SQL Server

- Tipos de seguridad:

Integrada: validación en PDC.

Mixta: validación en PDC o en SQL Server (si no existe esa cuenta de dominio).

Estándar: validar en SQL Server.

- Permisos por usuario:

Se utiliza esta opción al establecer la seguridad por primera vez.

- Permisos por objeto:

Se utiliza esta opción al crear nuevos objetos en la base de datos.

6.7.2. Seguridad de Windows 2000

Dominio. Grupo de ordenadores que poseen un concepto de seguridad común.

Administrado por un controlador principal de dominio (PDC).

Para acceder a cualquier recurso del dominio es necesario autenticarse (userid y password).

Tareas fundamentales:

Personalizar el entorno y rendimiento del sistema para cada usuario.

Garantizar la seguridad del sistema.

Proteger contra el acceso indebido de terceros.

Herramientas:

Administrador de usuarios para dominios: crear nuevos usuarios y definir directivas del sistema.

Explorador de Windows NT: realizar las configuraciones de seguridad en directorios y archivos.

Administrador de usuarios para dominios. Permite crear usuarios.

Windows Update:

Es la extensión en línea de Windows que ayuda a mantener actualizado el equipo.

Windows Update se utiliza para elegir actualizaciones para el sistema operativo, el software y el hardware del equipo. Periódicamente se agrega nuevo contenido al sitio, para que siempre disponga de las actualizaciones y correcciones más recientes con el fin de proteger el equipo y mantenerlo en perfecto funcionamiento.

Las categorías para las actualizaciones del sistema operativo Windows son:

- Actualizaciones críticas – correcciones de seguridad y otras actualizaciones importantes para mantener sus equipos actualizados y la seguridad de su red.
- Descargas recomendadas – los Service Packs más recientes para Windows e Internet Explorer y otras actualizaciones importantes.

- Herramientas de Windows – utilidades y otras herramientas ofrecidas para mejorar el rendimiento, facilitar las actualizaciones y simplificar el trabajo a los administradores del sistema.
- Actualizaciones multimedia y de Internet – las versiones más recientes de Internet Explorer, las actualizaciones para Windows Media™ Player y otras aplicaciones.
- Descargas de Windows adicionales – actualizaciones para su configuración de escritorio y otras características de Windows.
- Características plurilingües – menús y cuadros de diálogo, compatibilidad con idiomas y editores de métodos de entrada para diversas lenguas.

CONCLUSIONES

Para este sistema se llevaron a cabo una serie de encuestas al personal del Departamento de Adquisiciones para obtener la información inicial para el desarrollo. Las entrevistas se llevaron a cabo en diversas ocasiones para entender al máximo las necesidades del personal.

Una vez entrevistado por separado al personal de cada área se hizo una evaluación de las necesidades para englobarlas todas. Esta integración de información se le hizo llegar al departamento para que dieran su punto de vista y si era necesario concretar nuevamente una entrevista.

Una vez recopilada los datos necesarios, se comenzó con el análisis y diseño del sistema.

Durante el desarrollo de la aplicación se observó que normalmente, las especificaciones son incompletas o anómalas. De ahí que se pudiera regresar a las etapas del modelo seleccionado para el desarrollo de este sistema.

En base a la recopilación de datos y necesidades, se concluyó hacer grupos de usuarios por área y por nivel, con nivel se considera al cargo que ocupa el personal por área. Con esto se dio el caso que, siendo de la misma área, sólo cierto personal tenía más privilegios sobre otro.

Con lo anterior se fue cumpliendo el modelo de Vida Clásico Modificado de la ingeniería del Software para el desarrollo de sistemas.

En cuanto a la conversión de las bases de datos a la nueva base, las pruebas de exportación de datos fueron muy laboriosas, debido a que se tenían que hacer consultas de una sola tabla y tratar de normalizar la información para poder ser insertada en la nueva base. Finalmente se normalizó la mayor información ya que en algunas ocasiones los datos no se ingresaban con éxito, debido a que los datos no cumplían con las reglas de validación de la nueva base.

Las primeras pruebas de aceptación se llevaron a cabo por el equipo de desarrollo. Como se podría dar el caso de que se obviarán algunos detalles por parte del equipo de desarrollo, la siguiente etapa de pruebas se llevó a cabo con personal del departamento de Análisis, Desarrollo y Mantenimiento de Sistemas. Así se pudo liberar al personal de Departamento de Adquisiciones para que ellos hicieran su evaluación y pruebas.

Una vez instalado el sistema, se ha dado mantenimiento. Este mantenimiento se ha basado en la generación de nuevos reportes que ha solicitado el departamento, básicamente. Además de hacer respaldos continuos de la base de datos generada.

Los planes futuros para el *Sistema Integral para el Control y Registro de Facturas para el Departamento de Adquisiciones de la Dirección General de Bibliotecas de la UNAM*. Una vez que el departamento de Adquisiciones ha hecho la captura de información y ha generado sus informes y estadísticas, se pretende a cada dependencia se le instale el cliente sólo para que pueda estar monitoreando los movimientos que se hacen bajo esta o bien que pueda imprimir los informes generados.

Como parte del mantenimiento se hacen monitoreos de las conexiones iniciadas en el servidor al DBMS, con esto monitoreos se logró detectar un intruso.

Este proyecto se realizo con el objetivo de automatizar el proceso de registro el control de facturas para el departamento de adquisiciones, pero también con el fin de disminuir los costos que involucra la adquisición y manutención de software genérico adquirido por la UNAM para cubrir las mismas necesidades.

La visión a futuro es implementar los reportes en WEB para que tanto las dependencias como los proveedores puedan visualizar la información sin necesidad de hacer una solicitud al departamento de adquisiciones.

Con la disminución en la carga de trabajo al personal se vio reflejado en que ya no se hace doble captura y además que los informes se generan automáticamente sin necesidad de hacer recopilación de información o cálculos manuales, lo cual ha implicado que al personal que se había encargado de esta labora se le asigne otras resultando más productivo al departamento.

Finalmente se puede decir que se ha obtenido un sistema integral, amigable y productivo.

Apéndice A

❖ Configuración del ODBC

La norma ODBC (Open Database Connectivity) permite conectarse a una fuente de informaciones externa y utilizar sus datos. Es un intermediario entre bases de datos y aplicaciones. Se gestiona con el **Administrador ODBC** que dispone de los drivers necesarios para este proceso. Este Administrador se añade de manera automática en el **Panel de control de Microsoft Windows** durante la instalación de la mayoría de las bases de datos.



figura A.1 panel de control

La norma ODBC permite a las aplicaciones, utilizar el lenguaje de consulta **SQL** (Structured Query Language). Se utiliza generalmente para las conexiones a las bases implantadas en los servidores. Drivers ODBC (archivos que contienen la entera descripción de la estructura de los archivos de otras aplicaciones y que permiten la coherencia de los intercambios entre su base y el archivo consultado) estén disponibles para Oracle, Microsoft SQL Server, Sybase SQL Server y otros administradores de bases de datos en un gran número de plataformas.

La aplicación envía una consulta SQL. El Driver Manager intercepta esta llamada y la dirige de nuevo hacia el driver ODBC adecuado. Este driver analiza la consulta SQL y luego recupera los datos por la fuente de datos. Se hacen volver después dichos datos a la aplicación. La fuente de datos es entonces el vínculo entre la aplicación y el archivo físico.

Para crear un ODBC, se abre la opción de **Fuentes de datos de ODBC** del panel de control.

Una vez abierta, aparecerá una ventana similar a la siguiente, en la cual se oprimirá el botón de **Agregar** para que nos permita agregar el controlador que necesita nuestro ODBC.

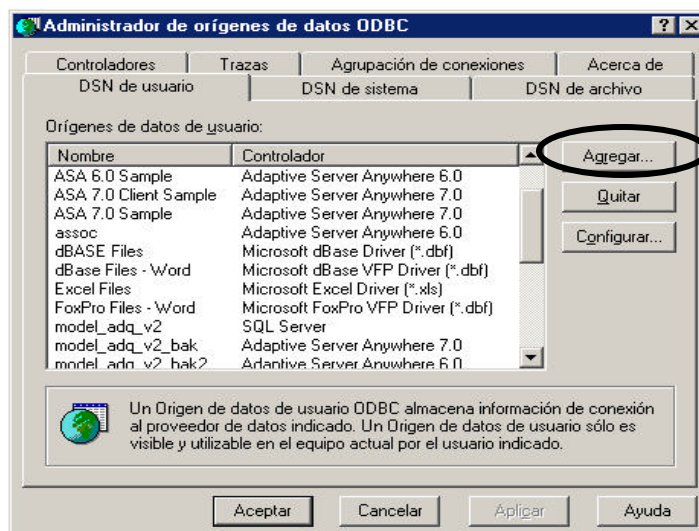


figura A.2 administrados de origen de datos ODBC

Una vez que se dio la opción de Agregar, se selecciona el controlador a utilizar. En este caso el controlador es para bases de datos de SQL Server.

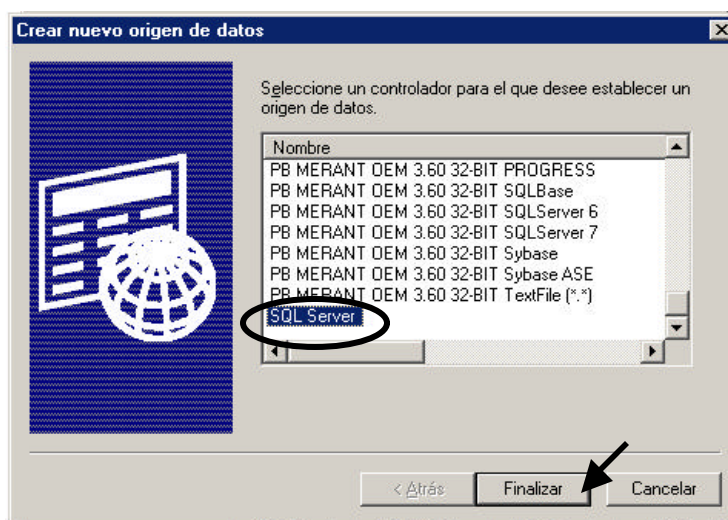


figura A.3 controladores para el ODBC

Después de seleccionar un controlador para el ODBC, se da un clic en Finalizar y posteriormente se abrirá una ventana similar a la siguiente. En la nueva ventana se iniciará el proceso de creación de ODBC.

Los primeros tres campos a llenar son:

- Nombre. Es el nombre con el cual la aplicación va a reconocer al ODBC.
- Descripción. Es una nota que nos va auxiliar en la identificación del ODBC.
- Servidor. En este campo se pone la IP del servidor en donde está montada la base de datos que utiliza la aplicación.

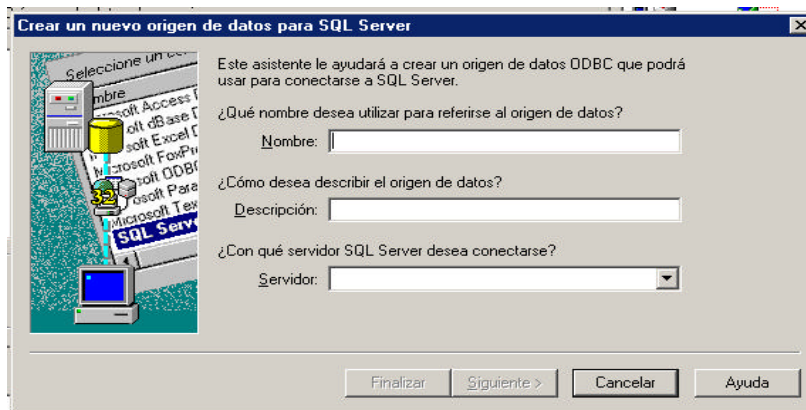


figura A.4 nuevo origen de datos

Una vez que se han llenado estos campos, se selecciona **Siguiente >**, con lo cual aparecerá una ventana similar a la siguiente.



figura A.5 configuración DSN

En primera instancia se selecciona la opción **Configuración del Cliente ...**, que nos permite establecer el medio de conexión. Una vez seleccionada aparecerá una ventana en la cual sólo se va a indicar que el protocolo es el **TCP/IP**, y **Aceptar**.

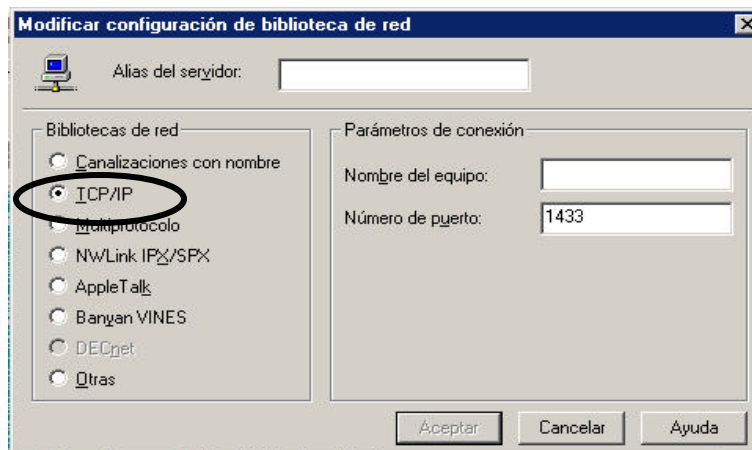


figura A.6 biblioteca de red

Posterior a la Configuración del Cliente, se selecciona la opción con la cual se va a iniciar sesión, en este caso será **Con la autenticación de SQL Server**, se da el usuario y contraseña. Se selecciona **Siguiente** >.

En la siguiente pantalla, se va a seleccionar la base de datos a utilizar. Se selecciona **Siguiente** >.

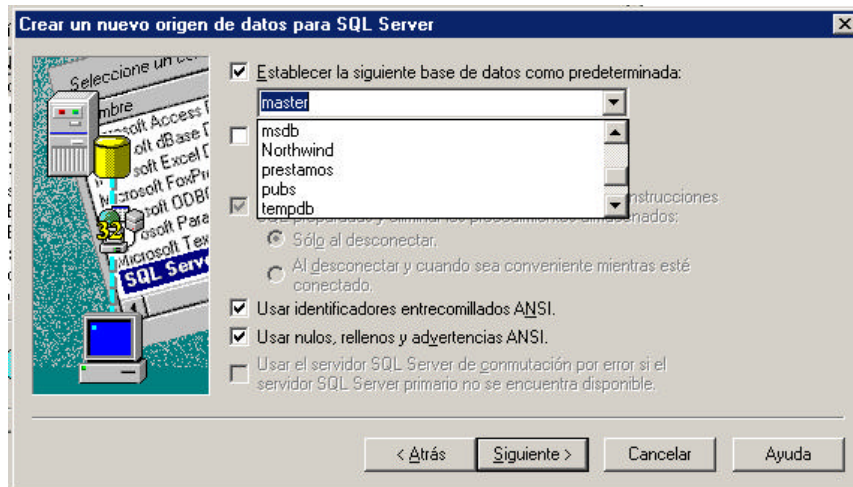


figura A.7 selección de base de datos

A los parámetros de la siguiente pantalla no se les modifica nada. Se selecciona **Finalizar**.



figura A.8 configuración de DSN

Una vez finalizada la configuración, aparecerá una ventana que nos indica los parámetros que se han ingresado.

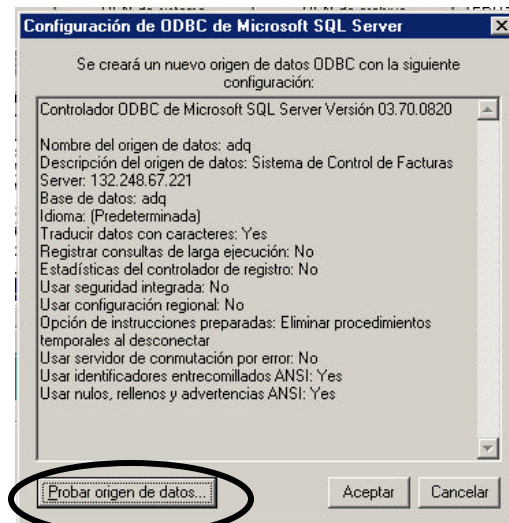


figura A.9 Configuración de ODBC

Para probar que hace la conexión de manera correcta, sólo se da clic al botón **Probar Origen de Datos**.

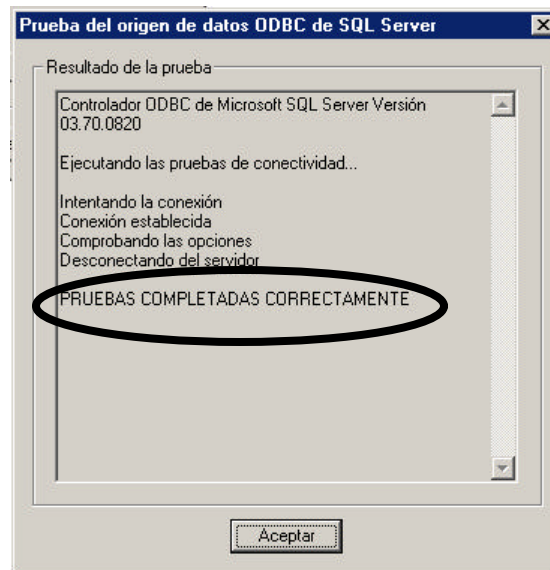


figura A.10 prueba de ODBC

Una vez realizado todos los pasos anteriores el ODBC queda agregado a la lista de **Orígenes de Datos**.

Nota: El ODBC creado es sólo visible y funcional en el equipo actual, lo cual indica que si la aplicación se va a instalar en n equipos, en los mismos hay que configurar el ODBC.

Apéndice B

❖ Código de la Creación de la Base de Datos

El código para la creación de la base se generó con Power Designer especificando el manejador de bases de datos que se utilizaría, en este caso, SQL Server.

```

/* ===== */
/* Database name: SISADQUI */
/* DBMS name: Microsoft SQL Server */
/* ===== */

/* ===== */
/* Database name: SISADQUI */
/* ===== */
create database SISADQUI
go

use SISADQUI
go

/* ===== */
/* Table: ARCHIVOS */
/* ===== */
create table ARCHIVOS
(
    ID_ARCHIVO varchar(10) not null,
    NOM_ARCHIVO varchar(20) null ,
    FECHA_PRODUCCION datetime null ,
    constraint PK_ARCHIVOS primary key nonclustered (ID_ARCHIVO)
)
go

/* ===== */
/* Table: AREAS */
/* ===== */
create table AREAS
(
    ID_AREA varchar(10) not null,
    AREA char(30) null ,
    constraint PK_AREAS primary key nonclustered (ID_AREA)
)
go

/* ===== */
/* Table: EDITORIALES */
/* ===== */
create table EDITORIALES
(
    ID_EDITORIAL varchar(10) not null,
    EDITORIAL varchar(20) null ,
    constraint PK_EDITORIALES primary key nonclustered (ID_EDITORIAL)
)
go

/* ===== */
/* Table: PROVEEDORES */
/* ===== */
create table PROVEEDORES

```



```

(
  CLAVE_PROV          varchar(15)    not null,
  NOMBRE_PROV        varchar(75)    not null,
  DIRECCION_PROV     varchar(60)    null ,
  TELEFONO           char(12)       null ,
  PAGINA_WEB_PROV    varchar(45)    null ,
  EDO_PROV           decimal(1)     null ,
  TIPO_PROV          decimal(1)     null ,
  ORIG_PROV          decimal(1)     null ,
  estatus_prov       decimal(1)     null ,
  constraint PK_PROVEEDORES primary key nonclustered (CLAVE_PROV)
)
go

/* ===== */
/* Table: SUBSISTEMAS */
/* ===== */
create table SUBSISTEMAS
(
  id_subsistema      varchar(10)    not null,
  subsistema         varchar(50)    not null,
  constraint PK_SUBSISTEMAS primary key nonclustered (id_subsistema)
)
go

/* ===== */
/* Table: TIPOS_MATERIAL_TIT */
/* ===== */
create table TIPOS_MATERIAL_TIT
(
  ID_TIPO_MATERIAL_TIT varchar(10)    not null,
  TIPO_MATERIAL_TIT   varchar(35)    null ,
  constraint PK_TIPOS_MATERIAL_TIT primary key nonclustered (ID_TIPO_MATERIAL_TIT)
)
go

/* ===== */
/* Table: TIPOS_MONEDA */
/* ===== */
create table TIPOS_MONEDA
(
  ID_TIPO_MONEDA     varchar(10)    not null,
  TIPO_MONEDA        varchar(30)    null ,
  constraint PK_TIPOS_MONEDA primary key nonclustered (ID_TIPO_MONEDA)
)
go

/* ===== */
/* Table: USUARIOS */
/* ===== */
create table USUARIOS
(
  NIP                varchar(15)    not null,
  PASSWD             varchar(10)    not null,
  ID_AREA            varchar(10)    not null,
  NOMBRE             varchar(65)    null ,
  constraint PK_USUARIOS primary key nonclustered (NIP)
)

```

```

go

/* ===== */
/* Table: FACTURAS */
/* ===== */
create table FACTURAS
(
    CLAVE_PROV          varchar(15)    not null,
    NUMERO_FACT        varchar(20)    not null,
    NIP                varchar(15)    null ,
    ID_TIPO_MONEDA     varchar(10)    null ,
    ID_PRESUPUESTO     decimal(18)    null ,
    FECHA_FACT        datetime        null ,
    MONTO_FACT         float(53)      null ,
    TIPO_CAMBIO        float(53)      null ,
    NUMERO_TITS        int            null ,
    NUMERO_VOLS        int            null ,
    FECHA_RECEPCION    datetime        null ,
    GASTOS_ENVIO       real           null ,
    PROCESOS           int            null ,
    CARGOS             int            null ,
    FECHA_TECNICO      datetime        null ,
    FECHA_SICE         datetime        null ,
    MODULO_FACT        varchar(15)    null ,
    OBSERVACIONES      varchar(200)   null ,
    TRIMESTRE          decimal(18)    null ,
    MONTO_NC           money          null ,
    FECHA_NC           datetime        null ,
    MONTO__FACT_SIN_DECUENTO money    null ,
    SUMA_CUXTIT        money          null ,
    SUMA_CUXTIT_DESC   real           null ,
    CLAVE_DEPENDENCIA2 varchar(15)    null ,
    FECHA_EVALUACION   datetime        null ,
    PERIODO_EVALUACION varchar(60)    null ,
    constraint PK_FACTURAS primary key nonclustered (CLAVE_PROV, NUMERO_FACT)
)
go

/* ===== */
/* Table: DEPENDENCIAS */
/* ===== */
create table DEPENDENCIAS
(
    CLAVE_DEPENDENCIA  varchar(15)    not null,
    NOMBRE_DEPENDENCIA varchar(75)    null ,
    EDO_DEPENDENCIA    decimal(1)     null ,
    ID_SUBSISTEMA      varchar(10)    null ,
    constraint PK_DEPENDENCIAS primary key nonclustered (CLAVE_DEPENDENCIA)
)
go

/* ===== */
/* Table: PRESUPUESTOSXDEP */
/* ===== */
create table PRESUPUESTOSXDEP
(
    ID_PRESUPUESTO     decimal(18)    not null,
    CODIGO_PROGRAMATICO varchar(20)    not null,

```

Apéndice

```
MONTO_PRESUPUESTO    decimal(18)    not null,
ANO_PRESUPUESTO      int           null    ,
TRIMESTRE             decimal(18)    null    ,
CLAVE_DEPENDENCIA    varchar(15)    not null,
ERXPATRONATO         money         null    ,
PREVIOS              money         null    ,
constraint PK_PRESUPUESTOSXDEP primary key nonclustered (ID_PRESUPUESTO)
)
go

/* ===== */
/* Table: TITULOS */
/* ===== */
create table TITULOS
(
  id_tit      char(12)      not null,
  MATRIZ      varchar(10)   null    ,
  ID_TIPO_MATERIAL_TIT  varchar(10)   null    ,
  ID_EDITORIAL  varchar(10)   null    ,
  TITULO      varchar(80)   not null,
  ANO_TIT     int          null    ,
  CLASIFICACION  varchar(50)   null    ,
  EDICION     varchar(10)   null    ,
  ISBN       varchar(50)   null    ,
  EDO_PORTADA  decimal(1)    null    ,
  MODULO_TIT  varchar(10)   null    ,
  id_tema     varchar(10)   null    ,
  autor      char(10)      null    ,
  precio     money        null    ,
  constraint PK_TITULOS primary key nonclustered (id_tit)
)
go

/* ===== */
/* Table: FACTURASXARCHIVO */
/* ===== */
create table FACTURASXARCHIVO
(
  ID_ARCHIVO    varchar(10)    not null,
  CLAVE_PROV    varchar(10)    not null,
  NUMERO_FACT   varchar(20)    not null,
  FECHA_INGRESO  datetime      null    ,
  constraint PK_FACTURASXARCHIVO primary key nonclustered (ID_ARCHIVO, NUMERO_FACT)
)
go

/* ===== */
/* Table: LIBROS */
/* ===== */
create table LIBROS
(
  CLAVE_PROV    varchar(15)    not null,
  NUMERO_FACT   varchar(20)    not null,
  MATRIZ      varchar(10)    not null,
  NO_ADQUI     varchar(13)    not null,
  id_tit      char(12)      null    ,
  OBSERVACIONES  varchar(80)    null    ,
  FECHA_COMPROBACION  datetime      null    ,
)
```

```

FECHA_CAPTURA      datetime      null  ,
ID_FRECUENCIA      char(10)      null  ,
PRECIO_UNAM_CON_DESC  money        null  ,
NO_ESTANTERIA      decimal(18)   null  ,
NO_PRESTAMO        decimal(18)   null  ,
NO_PROCESO_MENOR   decimal(18)   null  ,
NO_PROCESO_TECNICO decimal(18)   null  ,
constraint PK_LIBROS primary key nonclustered (CLAVE_PROV, NUMERO_FACT, MATRIZ,
NO_ADQUI)
)
go

/* ===== */
/* Table: PRECIOS */
/* ===== */
create table PRECIOS
(
  clave_prov      varchar(15)      not null,
  numero_fact     varchar(20)     not null,
  consecutivo_tit  decimal(18)     not null,
  FAC_CLAVE_PROV  varchar(15)      null   ,
  FAC_NUMERO_FACT varchar(20)      null   ,
  volxtit         decimal(18)     null   ,
  unam_sin_desc   money           null   ,
  lista_sin_desc  money           null   ,
  unam_porcentaje_desc money       null   ,
  lista_porcentaje_desc money       null   ,
  usr_captura     varchar(55)     null   ,
  constraint PK_PRECIOS primary key nonclustered (clave_prov, numero_fact, consecutivo_tit)
)
go

alter table USUARIOS
  add constraint FK_USUARIOS_AREAS foreign key (ID_AREA)
  references AREAS (ID_AREA)
go

alter table FACTURAS
  add constraint FK_FACTURAS_PRESUPUESTOSXDEP foreign key (ID_PRESUPUESTO)
  references PRESUPUESTOSXDEP (ID_PRESUPUESTO)
go

alter table FACTURAS
  add constraint FK_FACTURAS_PROVEEDORES foreign key (CLAVE_PROV)
  references PROVEEDORES (CLAVE_PROV)
go

alter table FACTURAS
  add constraint FK_FACTURAS_TIPOS_MONEDA foreign key (ID_TIPO_MONEDA)
  references TIPOS_MONEDA (ID_TIPO_MONEDA)
go

alter table FACTURAS
  add constraint FK_FACTURAS_USUARIOS foreign key (NIP)
  references USUARIOS (NIP)
go

alter table DEPENDENCIAS

```

```
add constraint FK_DEPENDENCIAS_SUBSISTEMAS foreign key (ID_SUBSISTEMA)
references SUBSISTEMAS (id_subsistema)
go

alter table PRESUPUESTOSXDEP
add constraint FK_PRESUPUESTOSXDEP_DEPENDENCI foreign key (CLAVE_DEPENDENCIA)
references DEPENDENCIAS (CLAVE_DEPENDENCIA)
go

alter table TITULOS
add constraint FK_TITULOS_EDITORIALES foreign key (ID_EDITORIAL)
references EDITORIALES (ID_EDITORIAL)
go

alter table TITULOS
add constraint FK_TITULOS_TIPOS_MATERIAL_TIT foreign key (ID_TIPO_MATERIAL_TIT)
references TIPOS_MATERIAL_TIT (ID_TIPO_MATERIAL_TIT)
go

alter table FACTURASXARCHIVO
add constraint FK_FACTURASXARCHIVO_ARCHIVOS foreign key (ID_ARCHIVO)
references ARCHIVOS (ID_ARCHIVO)
go

alter table FACTURASXARCHIVO
add constraint FK_FACTURAS_REF_198_FACTURAS foreign key (CLAVE_PROV, NUMERO_FACT)
references FACTURAS (CLAVE_PROV, NUMERO_FACT)
go

alter table LIBROS
add constraint FK_LIBROS_REF_187_TITULOS foreign key (id_tit)
references TITULOS (id_tit)
go

alter table LIBROS
add constraint FK_LIBROS_REF_204_FACTURAS foreign key (CLAVE_PROV, NUMERO_FACT)
references FACTURAS (CLAVE_PROV, NUMERO_FACT)
go

alter table PRECIOS
add constraint FK_PRECIOS_REF_191_FACTURAS foreign key (FAC_CLAVE_PROV,
FAC_NUMERO_FACT)
references FACTURAS (CLAVE_PROV, NUMERO_FACT)
go
```

❖ Código de las Funciones Principales de las Pantallas

Evento OPEN de la aplicación.

```
sqlca.DBMS = ProfileString("sisadqui.ini", "Profile adq2",&
"dbms", "")
```

```
sqlca.database = ProfileString("sisadqui.ini", "Profile adq2", &
"database", "")
sqlca.userid = ProfileString("sisadqui.ini", "Profile adq2",&
"userid", "")
sqlca.dbpass = ProfileString("sisadqui.ini", "Profile adq2",&
"dbpass", "")
sqlca.logid = ProfileString("sisadqui.ini", "Profile adq2",&
"logid", "")
sqlca.logpass = ProfileString("sisadqui.ini", "Profile adq2",&
"LogPassWord", "")
sqlca.servername = ProfileString("sisadqui.ini", "Profile adq2",&
"servername", "")
sqlca.dbparm = ProfileString("sisadqui.ini", "Profile adq2",&
"dbparm", "")

connect using SQLCA;

//Check for errors when connected to the database
If SQLCA.SQLCODE <> 0 Then
    MessageBox ("Error", sqlca.sqlerrtext)
    Halt Close
    Return
End If

//Abre la ventana de inicio de la aplicación
open(w_inicio)
```

Archivo sisadqui.ini

```
[DBMS_PROFILES]
Profiles=adq2

[Profile adq2]
DBMS=ODBC
Database=
UserId=
DatabasePassword=
LogId=
LogPassword=
ServerName=
DBParm=ConnectString='DSN=adq2;UID=adq;PWD=adq%03/.'
Lock=
Prompt=FALSE
AutoCommit=FALSE
```

Evento Clicked del botón ACEPTAR de la ventana de inicio.

```
//Definición de variables
long ll_nip, ll_passwd, getrow

//Validación de los datos de las cajas de texto
IF Len(Trim (sle_login.Text)) = 0 OR Len(Trim(sle_passwd.Text)) = 0 THEN
    messagebox ("Error", "Debe Ingresar su Nombre de Usuario o su Password", StopSign!)
ELSE
    ls_llogin_usr = Trim(sle_login.Text)
    ls_lpasswd_usr = Trim(sle_passwd.Text)

    ll_nip = gu_usrstore.Find ("nip = '" + ls_llogin_usr + "'", 1, gu_usrstore.RowCount ())
```

```
ll_passwd = gu_usrstore.Find ("x_passwd = " + ls_lpasswd_usr + """, 1,
gu_usrstore.RowCount ())

IF ll_nip > 0 AND ll_passwd > 0 AND ll_nip = ll_passwd THEN
    getrow = gu_usrstore.GetRowFromRowId(ll_nip)
    usr_nombre = gu_usrstore.GetItemString(getrow,"nombre")
    id_area = trim(gu_usrstore.getitemstring( getrow, 4))
    messagebox ("Bienvenido", "Bienvenido(a) " + usr_nombre,None!)
    OPEN (w_modeladq)
    CLOSE(PARENT)
    CHOOSE CASE string(id_area)
        CASE "id_admon"
            {Activación de los módulos correspondientes a admón.}
        CASE "id_fact"
            {Activación de los módulos correspondientes a fact.}
        CASE "id_trans"
            {Activación de los módulos correspondientes a trans}
        CASE "id_sice"
            {Activación de los módulos correspondientes a sice}
        CASE "id_aud"
            {Activación de los módulos correspondientes a aud}
        CASE "id_prov"
            {Activación de los módulos correspondientes a prov}
    END CHOOSE
ELSE
    MessageBox("Error", "El nombre de Usuario o la Contraseña son
Incorrectos",StopSign!)
END IF

END IF
```

Evento OPEN de la ventana BASE

```
//definición e inicialización de las variables
string nombre_cols
long indice = 0, i = 0

//Habilita el objeto de datos
dw_base.SetTransObject(sqlca)
dw_base.retrieve()

dw_base.Enabled = False

d_win_obj = dw_base.DataObject
dw_base.ScrollToRow(1)

//Llena el combo de campos con los cuales se puede hacer búsquedas
Do While dw_base.setcolumn(indice) <> -1
    nombre_cols=string(dw_base.GetColumnname())
    if Match(nombre_cols,"^x_*")=False then
        ddlb_campos.InsertItem(nombre_cols,i)
        i++
    end if
    indice ++
Loop
ddlb_campos.DeleteItem(i)

//Cuenta el número de registros del objeto de datos
RowCount = dw_base.RowCount()
```

```
dw_base.ScrollToRow(1)

IF RowCount > 0 Then
    //Despliega el número de registro actual
    sle_numreg.Text = string(1)
Else
    //Despliega el número de registro actual
    sle_numreg.Text = string(0)
    {Habilita y Deshabilita Botones}
End If

//despliega el total de registros
sle_totalreg.text = 'de ' + string(RowCount)
```

Evento Clicked del botón AGREGAR

```
RowCount = dw_base.RowCount()
dw_base.InsertRow(RowCount+1)
dw_base.ScrollToRow(RowCount+1)
```

Evento Clicked del botón ELIMINAR

```
//definición de variable
Integer msg

msg = MessageBox(" Advertencia ", "Desea Eliminar El Registro ", Exclamation!, OkCancel!)

If msg = 1 Then
    //Borra el registro actual
    dw_base.DeleteRow(dw_base.GetRow())
    If dw_base.UpDate() <> -1 Then
        Commit; //actualiza la base
    Else
        RollBack; //deshace los cambios
    End If
End if

RowCount = dw_base.RowCount()
sle_numreg.text = string(RowCount)
sle_totalreg.text = 'de ' + string(RowCount)
cb_eliminar.SetFocus()

If RowCount = 0 Then
    messagebox(" Advertencia ", "No Hay más Registros En La Base", None!)

cb_primerο.Enabled = False
cb_anterior.Enabled = False
cb_siguiete.Enabled = False
cb_ultimo.Enabled = False

cb_modificar.Enabled = False
cb_eliminar.Enabled = False
End If
```

Evento Clicked del botón ACTUALIZAR

```
//Variable
```



```
string a=""
long upd

a=trim(string(dw_base.Object.Data[dw_base.getrow(),1]))

//No actualiza si algún campo llave está vacío
if isNull(a) then
    messagebox("Error en la actualización","casilla vacía")
    return
end if

upd=dw_base.UpDate()

If upd <> -1 Then
    Commit; //Actualiza los cambios
    IF SQLCA.SQLCode = -1 THEN
        MessageBox("SQL error " + String(SQLCA.SQLDBCode),&
            SQLCA.SQLErrText )
    END IF
Else
    messagebox("", "registro no actualizado")
    RollBack; //Deshace los cambios
    dw_base.retrieve()
End If

dw_base.Enabled = False
```

Evento Clicked del botón BUSCAR

```
//Abre la ventana W_BUSCAR
openWithParm(w_buscar,Parent.dw_base )
```

Evento OPEN de la ventana W_BUSCAR

```
//variables
string nombre_cols
long indice = 1, i = 0, row

window a

dwbase=Message.powerobjectparm

row = dwbase.ScrollToRow(1)

//Llena el combo con los campos posibles para hacer la búsqueda
Do While dwbase.setColumn(indice) <> -1
    nombre_cols=string(dwbase.GetColumnname())
    //No incluye en el combo las columnas cuyo nombre comience con 'x_'
    if Match(nombre_cols,"^x_")=False then
        ddlb_campos.InsertItem(nombre_cols,i)
        i++
    end if
    indice ++
Loop
```

Evento CLICKED del botón BUSCAR de la ventana W_BUSCAR

```
//variables
```

```
long FoundRow
string tipo_dato,camp,col

//Cuenta el número de registros
RowCount = dwbase.RowCount()
//Trim elimina espacios de la cadena que se encuentra en la caja de texto
buscar = Trim(sle_buscar.text)
col=ddlb_campos.text

IF len(trim(ddlb_campos.text)) > 0 THEN

    IF len(trim(sle_buscar.text)) > 0 THEN
        tipo_dato=dwbase.Describe("" + col + ".ColType")
        camp = ddblb_campos.text + "=" + buscar + ""

        IF Match(tipo_dato,"char*") = False THEN
            camp = ddblb_campos.text + "=" + buscar + ""
        END IF

        //Busca el registro
        FoundRow =dwbase.find(camp,1,RowCount)

        IF FoundRow > 0 Then
            dwbase.ScrollToRow(FoundRow)
            messagebox("Aviso","Se Encontro El Registro",None!)
            close(Parent)
        ELSE
            messagebox("Aviso","No Se Encontro El Registro",None!)
            sle_buscar.text=""
            ddblb_campos.selectitem(0)
        END IF

    ELSE
        messagebox("Aviso","Debe Ingresar un Dato",None!)
    END IF

ELSE
    messagebox("Error","Debe Seleccionar un Campo",StopSign!)
END IF
```

Procedimietnos Almacenados

```
create procedure proc_transf_tec
@fecha_tecnico datetime
as
SELECT facturasarchivo.id_archivo,
        facturas.clave_prov,
        facturas.numero_fact,
        facturas.fecha_recepcion,
        presupuestosxdep.ano_presupuesto,
        facturas.procesos,
        facturas.numero_tits,
        facturas.numero_vols,
        facturas.fecha_tecnico,
        facturas.cargos,
        facturas.modulo_fact,
        facturas.clave_dependencia2
```

```
FROM facturas,
presupuestosxdep,
facturasxarchivo
WHERE ( facturasxarchivo.clave_prov = facturas.clave_prov ) and
      ( facturasxarchivo.numero_fact = facturas.numero_fact ) and
      ( facturas.id_presupuesto = presupuestosxdep.id_presupuesto ) and
(facturas.modulo_fact= 'id_trans') and
      ( facturas.fecha_tecnico = :fecha_tecnico)
UNION
SELECT ",
facturas.clave_prov,
facturas.numero_fact,
facturas.fecha_recepcion,
presupuestosxdep.ano_presupuesto,
facturas.procesos,
facturas.numero_tits,
facturas.numero_vols,
facturas.fecha_tecnico,
facturas.cargos,
facturas.modulo_fact,
facturas.clave_dependencia2
FROM facturas,
presupuestosxdep
WHERE facturas.id_presupuesto = presupuestosxdep.id_presupuesto and
facturas.modulo_fact= 'id_trans' and
facturas.fecha_tecnico = :fecha_tecnico and
not EXISTS
(
SELECT *
FROM facturasxarchivo
WHERE facturasxarchivo.clave_prov = facturas.clave_prov and
facturasxarchivo.numero_fact = facturas.numero_fact and
facturas.modulo_fact= 'id_trans' and
facturas.fecha_tecnico = :fecha_tecnico
)
```

Apéndice C

❖ Diccionario de datos

ENTIDAD ARCHIVOS

Nombre	Descripción	Áreas que utilizan el campo
Id_archivo	Proveedores como Puvill y Blackwell envían archivos (por ftp) que contienen los datos generales de los libros, es decir el formato MARC utilizado para la catalogación de libros, a este tipo de envío se le llama entrega por transferencia que es referente al tipo de proveedor. El departamento de adquisiciones se encarga de organizar tales archivos y registrarlos en el sistema para que sea enviados al departamento de producción. Cada nuevo archivo creado por el departamento puede contener la relación de archivos de una o más facturas. En este campo se le asigna un identificador al archivo generado por el departamento.	Transferencias.
Nom_archivo	Se registra sólo el nombre del archivo	Transferencias
Fecha_producción	Se registra la fecha en que el archivo es enviado al departamento de producción	Transferencias

ENTIDAD AREAS

Nombre	Descripción	Áreas que utilizan el campo
id_area	Se registra el identificador de las distintas áreas del departamento de adquisiciones	
area	Se registra el nombre del área	

ENTIDAD DEPENDENCIAS

Nombre	Descripción	Áreas que utilizan el campo
clave_dependencia	Se registra la clave asignada a la dependencia	Facturación y Transferencias.
nombre_dependencia	Se registra el nombre de la dependencia	Facturación y Transferencias.
edo_dependencia	Estado de la dependencia se refiere a si se encuentra dada de baja o no. {0 1 } cero para altas y uno para bajas.	Facturación y Transferencias.

ENTIDAD EDITORIALES

Nombre	Descripción	Áreas que utilizan el campo
id_editorial	Se registra el identificador de la editorial como llave única	Auditoria

Editorial	Se registra el nombre de la editorial	Auditoria
-----------	---------------------------------------	-----------

ENTIDAD FACTURAS

Nombre	Descripción	Áreas que utilizan el campo
Numero_fact	Se registra el número de factura. Para registrar una factura se hace una llave compuesta con el <i>id_proveedor</i> debido a que se puede dar el caso de que dos proveedores asignen un mismo número de factura, si sólo fuese <i>numero_fact</i> la llave primaria para cada factura, el sistema rechazaría la segunda factura que se quisiera registrar, así que le proveedor hace la diferencia.	Facturación y Transferencias.
Fecha_fact	Se registra la fecha en que se hizo el pedido	Facturación y Transferencias.
Monto_fact	Se registra el monto de la factura	Facturación y Transferencias.
Tipo_cambio	Dado que no todas las facturas son de procedencia nacional, se registra el tipo de cambio para la moneda de procedencia. El tipo de cambio nos va a permitir establecer el monto en pesos de la factura.	Facturación y Transferencias.
Numero_tits	Se registra la cantidad total de títulos por factura	Facturación y Transferencias.
Numero_vols	Se registra la cantidad total de volúmenes o libros por factura	Facturación y Transferencias.
Fecha_recepcion	Se registra la fecha en que llegó el pedido de la factura	Facturación y Transferencias.
Gastos_envio	En ocasiones los proveedores cobran cierta comisión por envío, esta cantidad es registrada en este campo	Proveedores
Procesos	Es el cantidad de libros que serán enviados a procesos técnicos. Haciendo la diferencia con el número de volúmenes.	Facturación y Transferencias.
Cargos	Cantidad de libros de la factura que son para cargos remotos.	Facturación y Transferencias.
Fecha_tecnico	Se registra la fecha en la que el material correspondiente a <i>procesos</i> se envía al departamento de <i>técnico</i>	Facturación y Transferencias.
Modulo_fact	Corresponde al área que haya registrado la factura, esta área es asignada según la reconocida cuando el usuario ingresa al sistema. {id_trans id_fact id_sice}	Sice, Facturación y Transferencias.
Observaciones	Observaciones acerca de la factura.	Sice, Proveedores, Auditoria, Facturación y Transferencias

Apéndice

Fecha_sice	Fecha en que factura es analizada y completa por el departamento de sice.	Sice
Monto_nc	Monto de la nota de crédito, si es que la hay.	Sice, Proveedores, Facturación y Transferencias
Fecha_nc	Fecha de la nota de crédito, si es que la hay.	Sice, Proveedores, Facturación y Transferencias
Monto_fact_sin_descuento	Monto de la factura sin descuento	Sice, Proveedores, Facturación y Transferencias
Suma_cuxtti	Suma de los costos unitarios por título.	Proveedores.
Suma_cuxtit_desc	Suma de los costos unitarios por título con descuento.	Proveedores.
Fecha_evaluación		
Periodo_evaluación		

ENTIDAD FACTURASXARCHIVO

Nombre	Descripción	Áreas que utilizan el campo
Fecha_ingreso	Fecha en la que se hace la asignación de la factura al archivo.	Transferencias

ENTIDAD LIBROS

Nombre	Descripción	Áreas que utilizan el campo
Id_tit	Llave primaria de la entidad libros.	Auditoria
No_adqui	Se registra en número de adquisición del libro, este número es único	Auditoria
Observaciones	Observaciones a cerca del libro.	Auditoria
Fecha_captura	Fecha en la que se hace la evaluación de la factura.	Auditoria
Id_frecuencia	Frecuencia con la que es utilizado el material. Toma los valores de M de Mucho, R de Regular y P de Poco.	Auditoria
Precio_UNAM_con_desc	Se registra el precio del libro a la UNAM con descuento	Auditoria
Precio_lista	Se registra el precio que ofrece el proveedor	Auditoria
Fecha_comprobación	Fecha en la que se hace la auditoria a las dependencias	Auditoria
No_estantería	Cantidad de volúmenes en estantería por factura al momento de hace la auditoria a la dependencia.	Auditoria
No_prestamo	Cantidad de préstamos por factura al momento de hace la auditoria a la dependencia.	Auditoria
No_proceso_menor	Cantidad de procesos menores por factura al momento de hace la auditoria a la dependencia.	Auditoria
No_proceso_tecnico	Cantidad de procesos técnicos por factura al	Auditoria

	memento de hace la auditoria a la dependencia.	
--	--	--

ENTIDAD PRECIOS

Nombre	Descripción	Áreas que utilizan el campo
consecutivo_tit	Número consecutivo de la lista de títulos asignado en la fatura.	Proveedores
volxtit	Cantidad de volúmenes para cada título de la factura.	Proveedores
Unam_sin_desc	Monto del título sin descuento.	Proveedores
Lista_sin_descuento	Precio de lista sin descuento.	Proveedores
Unam_porcentaje_desc	Porcentaje de descuento por títulos.	Proveedores
Lista_porcentaje_desc	Porcentaje de descuento al precio de lista.	Proveedores
Usr_captura	Usuario que capturó los detalles de los títulos de la factura.	Proveedores

ENTIDAD PRESUPUESTOSXDEP

Nombre	Descripción	Áreas que utilizan el campo
Id_presupuesto	Es la clave con la que se identifica el presupuesto asignado a las dependencias	Sice
codigo_programatico	Es el código programático	Sice
monto_presupuesto	Se registra el monto del presupuesto	Sice
año_presupuesto	Se registra el año en que se asigno el presupuesto	Sice
Trimestre	Trimestre del año relacionado el presupuesto. {1 2 3 4}	Sice
Erxfanonato		Sice
Previos		Sice

ENTIDAD PROVEEDORES

Nombre	Descripción	Áreas que utilizan el campo
clave_prov	Se registra la clave asignada al cierto proveedor	Facturación, Transferencias.
nombre_prov	Se registra el nombre del proveedor	Facturación y Transferencias.
direccion_prov	Se registra la dirección física del proveedor	Facturación y Transferencias.
telefono	Se registra el teléfono del proveedor	Facturación y Transferencias.
pagina_web_prov	Se registra la página web del proveedor, en caso de tenerla	Facturación y Transferencias.
edo_prov	Estado del proveedor se refiere a si se encuentra dado de baja o no. Valor 0 para activo y 1 para bajas. {0 1}	Facturación y Transferencias.
Orig_prov	Se refiere a la procedencia del proveedor, ya sea nacional o	Proveedores

	extranjero. 0 para Nacional y 1 para Extranjero. {0 1}	
Estatus_prov	Se refiere a el tipo de proveedor, ya sea Distribuidor o editorial. 0 para Editorial y 1 para Distribuidor {0 1}	Proveedores

ENTIDAD SUBSISTEMAS

Nombre	Descripción	Áreas que utilizan el campo.
Id_subsistema	Clave asignada al sistema bibliotecario de la UNAM	Proveedores
Subsistema	Sistema Bibliotecario de la UNAM. Este subsistema es asignado a las dependencias. Ejemplo CCH (Colegio de Ciencias y Humanidades), H (Humanidades), etc. En el sistema sirve para generar estadísticas. {H IC EYF ENP CCH EYAU OD}	Proveedores

ENTIDAD TIPOS_MAT_TIT

Nombre	Descripción	Áreas que utilizan el campo
Id_tipo_material_tit	Debido a que no sólo se hacen adquisiciones de libros, el sistema dará la opción a registrar el tipo de materia. Este campo registra el identificador del material	Auditoria
tipo_material_tit	{LIBRO REVISTA MAPA MONOGRAFIA DISCO COMPACTO}	Auditoria

ENTIDAD TIPOS_MONEDA

Nombre	Descripción	Áreas que utilizan el campo
id_tipo_moneda	Se registra un identificador para un tipo de moneda {MN DL FR}. Es importante que el catálogo contenga "MN" ya que en base a esta se hacen comparaciones para ver si hay que multiplicar o no por un tipo de cambio.	Facturación y Transferencias.
tipo_moneda	Se registra el nombre de la moneda	Facturación y Transferencias.

ENTIDAD TITULOS

Nombre	Descripción	Áreas que utilizan el campo
Id_tit	Es la clave con la que se identifica al título.	
Matriz	Se registra el número de matriz que es asignado al libro.	Auditoria
Título	Se registra el título del libro	Auditoria
Año_tit	Se registra el año de publicación del libro	Auditoria
Clasificacion	Se registra la clasificación asignada al libro	Auditoria
Edición	Se registra el número de edición del libro	Auditoria

Isbn	Se registra el ISBN del libro	Auditoria
Edo_portada		Auditoria
Autor	Autor del libro.	Auditoria

ENTIDAD USUARIOS

Nombre	Descripción	Áreas que utilizan el campo
Nip	Se registra el nombre de usuario del sistema	Sice, Proveedores, Auditoria, Facturación y Transferencias
Passwd	Se registra la clave para el usuario	Sice, Proveedores, Auditoria, Facturación y Transferencias
Nombre	Se registra el nombre del usuario	Sice, Proveedores, Auditoria, Facturación y Transferencias

BIBLIOGRAFÍA

- James Martín, Organización de las Bases de Datos
- Thao Vo, SQL Plus Bajo Oracle
- Ian Sommerville, Software Engineering, 6th Edition, 2001, 693 p.p.
- Pressman, Roger S., Software engineering
- Edward Yourdon, Análisis Estructurado, 1989, 735 p.p.
- Ken England, The SQL Server 7.0 Handbook, 1999, 451 p.p.
- Coffman, SQL Server 7 Manual de Referencia,
- Richard Waymire, Aprendiendo Microsoft SQL Server 2000 en 21 días
- Ramesh chandak, Advanced Power Builder Techniques, 1998, 843 p.p.
- William B. Heys, Power Builder, Prentice Hall, 844 p.p
- Ing. Orlando Zaldívar Zamorategui, Apuntes de Ingeniería de Programación,

LINKS

- http://www.dcs.napier.ac.uk/hci/VB5/HTML/tutorial_16.html
- <http://www.sei.cmu.edu/>
- <http://www.rational.com/>
- <http://www.powerbuilder.org/>
- <http://sybooks.sybase.com/pbg0800e.html>
- [http://support.microsoft.com/default.aspx?scid=kb;\[LN\];Q313980](http://support.microsoft.com/default.aspx?scid=kb;[LN];Q313980)
- http://support.microsoft.com/default.aspx?scid=http://support.microsoft.com/support/servicepacks/SQL/7.0/ESsp4ReadMe.asp#_5.12.2_improved_retrieval_of_versioned_objects
- <http://v4.windowsupdate.microsoft.com/es/default.asp>
- <http://www.sybase.com/detail?id=1013232>

TUTORIALES

- Seguridad en bases de datos, Seguridad en computo 2001, DGSCA UNAM
- Administración de W2k, DGB, UNAM

Índice de Figuras

1. Definición del Sistema	
2. Antecedentes	
2.1. Modelo en Cascada	8
2.2. Modelo Evolutivo	9
2.3. Modelo en Espiral	10
2.4. Base de Datos	13
2.5. Entidad	13
2.6. Formas Normales	15
3. Análisis del Sistema	
4. Análisis Estructurado	
4.1. Proceso	33
4.2. Flujo	34
4.3. Almacén	34
4.4. Terminador	34
4.5. fk_facturas_proveedores	35
4.6. fk_facturas_reference_presupue	35
4.7. fk_presupue_reference_depend	35
4.8. Arquitectura Cliente – Servidor. Dos capas	37
4.9. Arquitectura Cliente – Servidor. Tres Capas	38
5. Desarrollo	
5.1. Modelo Conceptual	49
5.2. Modelo Físico	50
5.3. Ventana de Inicio	56
5.4. Barra de Menús	56
5.5. Módulos de Facturación	57
5.6. Captura de Facturas	58
5.7. Módulo de Transferencias	58
5.8. Captura de Archivos	59
5.9. Captura de Facturas por Archivo	59
5.10. Módulo de Sice	60
5.11. Captura de Códigos Programáticos	60
5.12. Módulo de Proveedores	61
5.13. Selección de Facturas	61
5.14. Precios	61
5.15. Módulo de Auditorias	62
5.16. Captura de Títulos	62
5.17. Comprobación de Títulos	62
5.18. Menú Catálogos	63
5.19. Argumentos para Informes	63
5.20. Menú Informes	64
5.21. Informe Sice	65
5.22. Pantalla Base	66
5.23. Icono Ayuda	68
5.24. Ayuda	68

6. Liberación	
Apéndice A	
A.1. Panel de Control	79
A.2. Administrador de ODBC	80
A.3. Crear nuevo ODBC	80
A.4. Crear nuevo ODBC para SQL Server	81
A.5. Configuración DNS de SQL Server	81
A.6. Modificar Configuración de Biblioteca de Red	81
A.7. Crear nuevo ODBC para SQL Server	82
A.8. Configuración DNS de SQL Server	82
A.9. Probar ODBC	83
A.10. Prueba de ODBC	83
Apéndice B	
Apéndice C	

Índice de Tablas

1. Definición del Sistema	
2. Antecedentes	
2.1. Documentos del Modelo en Cascada	9
2.2. Tabla comparativa de los Paradigmas de Ingeniería del Software	11
3. Análisis del Sistema	
4. Análisis Estructurado	
4.1. Simbología de los terminadores de las relaciones	35
5. Desarrollo	
5.1. Relaciones entre entidades	49
5.2. facturasxarchivo	50
5.3. precios	51
5.4. libros	51
5.5. archivos	51
5.6. facturas	52
5.7. edos_libros	52
5.8. titulos	53
5.9. usuarios	53
5.10. presupuestosxdep	53
5.11. proveedores	53
5.12. editoriales	54
5.13. areas	54
5.14. dependencias	54
5.15. subsistemas	54
6. Liberación	
Apéndice A	
Apéndice B	
Apéndice C	

Índice de Diagramas

1. Definición del Sistema	
1.1. Pasos en el Diseño de una Base de Datos	
2. Antecedentes	
3. Análisis del sistema	
3.1. Área del Departamento de Adquisiciones	26
3.2. Proceso de Registro de Facturas	29
4. Análisis estructurado	
4.1. Diagrama de Gantt	43
5. Desarrollo	
5.1. Diagrama Jerárquico Funcional en Nivel 1	45
5.2. Diagrama Jerárquico Funcional en Nivel 2	46
5.3. DFD Nivel 0	47
5.4. DFD Nivel 1	47
6. Liberación	
Apéndice A	
Apéndice B	
Apéndice C	