



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

“DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA  
PARA LA RECEPCIÓN, ADMINISTRACIÓN Y RECUPERACIÓN  
DE TESIS ELECTRÓNICAS PARA LA INTEGRACIÓN  
DE LA BIBLIOTECA DIGITAL EN LA  
DIRECCIÓN GENERAL DE BIBLIOTECAS”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

ISRAEL DURÁN MARTÍNEZ  
ALEJANDRO ALBERTO RAMÍREZ RAMOS



DIRECTOR DE TESIS  
M.I. MARCIAL CONTRERAS BARRERA

NOVIEMBRE 2004



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**

	Pag.
<b>CAPÍTULO 1. ANTECEDENTES Y DEFINICIÓN DEL PROBLEMA</b>	
1.1 Introducción.	2
1.2 Antecedentes de las tesis digitales.	3
1.2.1 Las tesis digitales en la UNAM.	4
1.2.2 Tesis digitales.	5
1.2.3 Tesiunam.	5
1.2.3.1 Breve Reseña Histórica.	6
1.2.3.2 Antigüedad.	6
1.2.3.3 Grado.	8
1.2.3.4 Procedencia.	8
1.2.3.5 Escuelas y facultades.	9
1.3 Bibliotecas digitales.	10
1.3.1 Características.	11
1.3.2 Arquitectura técnica.	11
1.3.3 Construcción de colecciones digitales.	12
1.3.4 La solidez de la colección.	12
1.3.5 Copyright.	13
1.3.6 Preservación.	14
1.4 Definición del problema.	15
1.5 Objetivos.	16
1.6 Alcances.	16
<b>CAPÍTULO 2. INGENIERÍA DE SOFTWARE</b>	
2.1 Aplicaciones de software.	19
2.2 Ingeniería del software.	20
2.3 Modelos de procesos de software.	22
2.3.1 Modelo lineal secuencial .	23
2.3.2 Modelo de prototipos.	27
2.3.3 Modelo en espiral.	29
2.3.4 Otros modelos.	30
2.4 Conceptos básicos de bases de datos.	34
2.4.1 Objetivos de las bases de datos.	36
2.4.2 Modelo de datos.	38
2.4.3 Sistemas manejadores de bases de datos (DBMS).	42
2.4.4 Formas normales.	44
2.5 Sistemas operativos	45
2.5.1 Conceptos básicos de Sistemas Operativos	47
2.5.2 Componentes del Sistema Operativo	47
2.5.3 Evolución de la estructura de los Sistemas Operativos	49
2.5.4 Historia de los Sistemas Operativos:	52
MS- DOS, Unix, Windows, Macintosh, Linux.	
<b>CAPÍTULO 3. ANÁLISIS DEL SISTEMA</b>	
3.1 Análisis de los Recursos Disponibles.	63
3.1.1 Recursos informáticos disponibles.	63
3.2 Análisis de la Bases de Datos.	63
3.2.1 Modelo Entidad Relación.	63
3.3 Análisis del formato MARC 21.	71

3.3.1 Introducción.	71
3.3.2 MARC 21.	73
3.3.3 Origen de los registros MARC.	74
3.4 Análisis de la Estructura y el Flujo de Datos del Sistema.	77
3.4.1 Diagrama Jerárquico Funcional.	78
3.4.2 Diagrama de Flujo de Datos.	80
<b>CAPÍTULO 4. DISEÑO DEL SISTEMA.</b>	
4.1 Descripción del sistema.	84
4.1.1 División de una tesis.	86
4.2 Diseño de la Base de Datos.	87
4.2.1 Definición de Tablas.	88
4.2.2 Diagrama Entidad Relación.	93
4.3 Características de los sistemas manejadores de base de datos.	94
4.3.1 Oracle.	95
4.3.1.1 Introducción a Oracle.	95
4.3.1.2 Arquitectura de Oracle.	95
4.3.1.3 Funciones de Oracle.	96
4.3.1.4 Estructura física de una base de datos en Oracle.	97
4.4 Diseño de la interfaz de usuario.	99
4.4.1 Diseño de la interfaz gráfica.	99
4.4.1.1 Módulo de registro.	100
4.4.1.2 Módulo de alumnos.	101
4.4.1.2.1 Altas alumnos.	102
4.4.1.2.2 Bajas alumnos.	103
4.4.1.2.3 Actualizaciones alumnos.	104
4.4.1.2.4 Búsquedas alumnos.	105
4.4.1.2.5 Seleccionar alumnos.	106
4.4.1.2.6 Borrar selección de alumnos.	107
4.4.1.3 Módulo de asesores.	108
4.4.1.3.1 Altas asesores.	109
4.4.1.3.2 Bajas asesores.	110
4.4.1.3.3 Actualizaciones asesores.	111
4.4.1.3.4 Búsquedas asesores.	112
4.4.1.3.5 Seleccionar asesores.	113
4.4.1.3.6 Borrar selección de asesores.	114
4.4.1.4 Módulo de tesis.	115
4.4.1.4.1 Altas de tesis.	116
4.4.1.4.2 Bajas de tesis.	117
4.4.1.4.3 Actualizaciones de tesis.	118
4.4.1.4.4 Búsqueda de tesis.	119
4.4.1.4.5 Selección de tesis.	120
4.4.1.4.6 Borrado selección de tesis.	121
4.4.1.5 Módulo de capítulos y subtemas.	122
4.4.1.5.1 Altas capítulos.	123
4.4.1.5.2 Bajas capítulos.	124
4.4.1.5.3 Altas subtemas.	125
4.4.1.5.4 Bajas subtemas.	126
4.4.1.5.5 Actualizaciones de apartados y subtemas.	127
4.4.1.6 Módulo de revisión.	128
4.4.1.6.1 Verificar datos y generar archivo en PDF's.	128

---

---

4.4.1.7 Módulo de archivos.	129
4.4.1.7.1 Copiar y encriptar archivos.	129
4.4.1.8 Utilerías.	130
4.4.1.8.1 Importar índice.	131
4.4.1.8.2 Opciones administrativas.	131
4.4.1.8.3 Desbloquear tesis.	132
4.4.1.9 Módulo de catálogos.	133
4.4.1.9.1 Altas en catálogos.	134
4.4.1.9.2 Bajas en catálogos.	135
4.4.1.9.3 Búsquedas en catálogos.	136
4.4.2 Diseño de la aplicación WEB.	137
4.4.2.1 Módulo de tesis.	139
4.4.2.2 Módulo de alumno.	140
4.4.2.3 Módulo de apartados.	141
4.4.2.3.1 Ingreso de un apartado.	142
4.4.2.3.2 Actualización de un apartado.	143
4.4.2.3.3 Borrado de un apartado.	144
4.5 Características de las herramientas de software utilizado.	144
4.5.1 Interfaz gráfica.	144
4.5.2 Aplicación WEB.	148
<b>CAPÍTULO 5. DESARROLLO DEL SISTEMA</b>	
5.1 Desarrollo de la Base de Datos.	152
5.2 Desarrollo del sistema.	155
5.2.1 Desarrollo de la interfaz gráfica.	155
5.2.2 Desarrollo de la aplicación WEB.	166
<b>CAPÍTULO 6. IMPLEMENTACIÓN, PRUEBAS Y MANTENIMIENTO</b>	
6.1 Implementación del DBMS.	176
6.2 Implementación del Servidor Web.	176
6.3 Implementación de la Interfaz gráfica.	179
6.4 Funcionalidad del Sistema	181
6.5 Implementación para la búsqueda y visualización de las tesis digitales.	182
6.6 Pruebas	187
6.7 Mantenimiento	189
<b>CONCLUSIONES</b>	192
<b>BIBLIOGRAFÍA</b>	196

**CAPÍTULO 1**

---

---

## **ANTECEDENTES Y DEFINICIÓN DEL PROBLEMA**

### **1.1 Introducción**

Las tecnologías de información y comunicación (TIC) están impactando directamente en los hábitos informativos de los usuarios, así como en la organización de los materiales documentales dentro de las bibliotecas y centros de información. El soporte digital ofrece a la Universidad Nacional Autónoma de México (UNAM) una nueva herramienta para preservar, conservar y difundir la gran cantidad de tesis que se generan cada año (aproximadamente 12,000 tesis).

La Dirección General de Bibliotecas día a día se está actualizando en cuanto al desarrollo de sistemas para satisfacer todas las necesidades de la comunidad universitaria, por lo tanto, en esta ocasión se trabajará en el desarrollo de una biblioteca digital.

La colección de tesis digitales de la Biblioteca Central de la UNAM juega un doble papel dentro de la institución. Por un lado, y dadas las características del soporte donde se encuentran contenidas, forman parte de los recursos electrónicos con que cuenta el departamento de consulta de Biblioteca Central y que junto con colecciones como revistas electrónicas, bases de datos en línea y libros electrónicos, entre otros, proporcionan al usuario un acceso ágil a la información. La colección de tesis digitales junto con las que se encuentran impresas y en microfilm conforman uno de los acervos más grandes que se tienen en México y Latinoamérica, ya que se compone de un poco más de 318,000 trabajos recepcionales, generados desde 1900 hasta la fecha tanto en la UNAM como en su sistema de universidades incorporadas. Dichas tesis se encuentran en papel y microfilm.

Todos estos materiales se encuentran registrados en la base de datos TESIUNAM, la cual ofrece varias alternativas de búsqueda. Ahora bien, hay que mencionar que esta base se encuentra disponible en el sitio Web de la Dirección General de Bibliotecas con el objetivo de lograr la mayor difusión de dichos materiales.



La colección de tesis se ha ido acrecentando a través de los años por un hecho significativo: la Biblioteca Central es depositaria de todos los trabajos que, como requisito para titularse, presentan los alumnos de los niveles de licenciatura y postgrado existentes en la Universidad. Lo anterior se encuentra establecido en el *Reglamento General de exámenes de la UNAM*, en cuyo Capítulo IV correspondiente a los exámenes profesionales y de grado, y muy concretamente en los artículos 20 y 26, se especifica: “las tesis profesionales y para obtener el grado de maestro o de doctor, se presentarán mecanografiadas, con un número de copias igual al de los sinodales propietarios y suplentes, otra para la biblioteca de la facultad o escuela y dos más ...” (UNAM, 2001), entendiéndose que éstas dos últimas son para el Departamento de tesis de Biblioteca Central. Así, podemos darnos cuenta de la relevancia de esta colección y del crecimiento que tendrá a futuro.

Si consideramos el volumen actual de la colección y su perspectiva de crecimiento, podemos afirmar que la Biblioteca central se ha visto en la necesidad de implementar nuevos soportes, con la finalidad de preservar, conservar y difundir todo este cúmulo de información. Por ello, ha hecho uso del microfilm y muy recientemente de la digitalización, tecnología de vanguardia que posibilita, entre otras cosas, la agilidad en el acceso a la información y sobre todo el ahorro de espacio físico dentro del recinto destinado a estos importantes materiales.

## 1.2 Antecedentes de las tesis digitales

El concepto *tesis digitales* no es nuevo; de hecho, se sabe que desde mediados de la década de los noventa la empresa denominada UMI (University Microfilms Inc.) ofrece el acceso en formato digital a las tesis doctorales de las Universidades de Canadá y Estados Unidos, previo pago del solicitante. Hay que mencionar que esta empresa ha proporcionado diversos servicios de acceso a tesis en diferentes formatos desde 1938, año de su creación.

John MacColl (2002) menciona que el proyecto denominado ETD (*Electronic Theses and Dissertations*) fue creado desde 1994 por el *Virginia Polytechnic Institute* y en el cual la característica predominante es la de contar con el acceso libre al texto completo de las tesis.

Existe, por otro lado, un proyecto relevante en el cual participan cerca de 132 universidades interesadas en digitalizar sus colecciones, denominado ND LTD (Networked Digital Library of theses and Dissertations), cuyo objetivo es crear una biblioteca digital de tesis y disertaciones, que permita el intercambio de información entre las entidades participantes.

Asimismo, en el Reino Unido existen importantes proyectos como el de "Theses Alive", "Daedalus" y "E-theses in the UK", que refuerzan la trascendencia de este nuevo soporte para las colecciones de este tipo. Por mencionar otro dato, en la Universidad de Sao Paulo se creó, desde el año 2001, la Biblioteca Digital de Tesis y maestría, como de algunos programas de posgrado, y contempla el crecimiento que a futuro tendrá esta colección.

En México existen algunas dependencias que tienen proyectos de digitalización; la Universidad de las Américas (UDLA), por ejemplo, brinda el servicio de consulta libre a estos materiales desde su sitio web.

### **1.2.1 Las tesis digitales en la UNAM**

En la Universidad Nacional Autónoma de México, el proyecto de tesis digitales se tenía considerado desde el año de 1996, dentro de la Dirección General de Bibliotecas, con el objetivo de aplicar las nuevas tecnologías de información dentro de las colecciones y servicios que ésta ofrece. Más tarde, se empezó a trabajar en el tratamiento que se le daría a la información para convertirla a digital y en lo referente a los derechos de autor, a fin de obtener el permiso de los tesisistas para difundir sus trabajos en formato electrónico.

Cabe mencionar que anualmente se reciben en la Biblioteca Central alrededor de 12,000 tesis impresas, de las cuales aproximadamente el 90 % son producidas en la UNAM. El plan de trabajo consistió en digitalizar las tesis de 1998 a 2001.

La conversión de impreso a digital se llevó a cabo en agosto de 2001; para tal efecto, el Departamento de Tesis se responsabilizó de compilar las tesis impresas en ese periodo y entregarlas a la empresa encargada de digitalizar los materiales (UNAM,2002).

Finalmente, en noviembre de 2002 se empezó a proporcionar el acceso a estos trabajos recepcionales en el Departamento de Consulta, como parte de los recursos electrónicos con que cuenta. Por el momento, el acceso únicamente es mediante una red local, pero en un futuro se tiene considerado suministrarlo a través de la Red UNAM y al exterior.

### **1.2.2 Tesis digitales**

Las tesis digitales entran en la designación de las llamadas *publicaciones digitales*, ya que se necesita un medio electrónico para visualizarlas. Pero más allá de una simple publicación electrónica, las tesis digitales son documentos de gran valor para la entidad que las genera, puesto que son el producto de los procesos de enseñanza/aprendizaje y de investigación que se realizan dentro de las aulas de estudio, desde el nivel licenciatura hasta el de doctorado y especialidades, y que a diferencia de otras publicaciones como las revistas no generan ningún costo de adquisición para la entidad.

Al igual que la mayoría de las publicaciones electrónicas, las tesis han recibido un tratamiento que favorece que la información tenga ciertas ventajas a la hora de visualizarlas, como las llamadas ligas, que facilitan acceder más ágilmente y navegar de un párrafo a otro o incluso de un capítulo a otro, entre otras ventajas más.

El acceso a la colección de tesis digitales como novedoso servicio dentro de la Biblioteca Central, ofrece al usuario diversas ventajas de las cuales los materiales impresos y microfilmados adolecen. Por un lado, este nuevo soporte permite la consulta multiusuario de un mismo texto, agiliza la lectura y reproducción de alguna parte del texto y junto con la posibilidad de consultar la base de datos TESIUNAM en texto completo repercute directamente en ahorro de tiempo para el usuario.

### **1.2.3 Tesiunam**

Está integrado por un Catálogo de Tesis de los egresados de nivel licenciatura y posgrado de la UNAM e instituciones de enseñanza superior incorporadas a nuestra universidad.

### 1.2.3.1 Breve Reseña Histórica

**1986** Inicia TESIUNAM.

**1992** Se edita en disco compacto.

**1996** Aparece la segunda edición y se proyecta la digitalización de tesis..

**1997** A partir de este año, está disponible a través del Sitio Web de la Dirección General de Bibliotecas.

**2002** Se concluye la digitalización de más de 40,000 tesis correspondientes a los años 1998 – 2001.

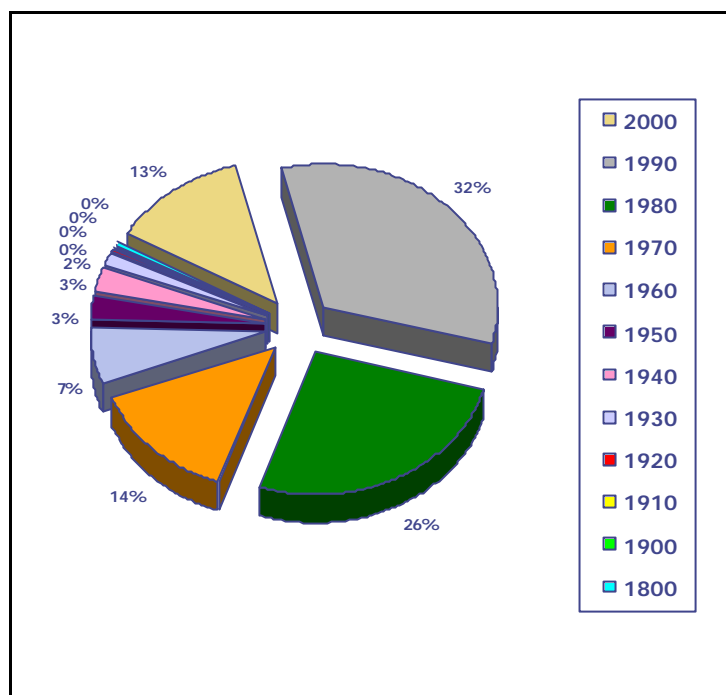
Tesiunam representa los trabajos recepcionales originales de nuestra máxima casa de estudios de los cuales más del 90% se encuentran físicamente en la Biblioteca Central además de encontrarse en cada biblioteca de cada escuela o facultad. Para fines del año pasado se tenían registradas 310,621 tesis cuyo análisis muestra lo siguiente:

- ◆ Antigüedad
- ◆ Grado
- ◆ Procedencia

### 1.2.3.2 Antigüedad

Se poseen 1,437 tesis del siglo XIX que representan el 0.46%; 269,417 tesis del siglo XX representando un 86.74% y 39,767 tesis del 2000 a la fecha cuyo porcentaje es de 12.80%. La descripción anterior se puede apreciar con mayor detalle en el siguiente gráfico:

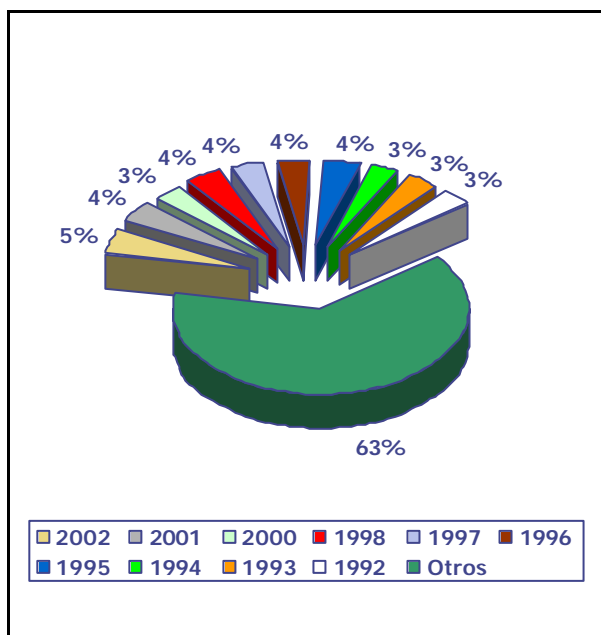
Década	Número	%
2000	39,767	12.80
1990	102,183	32.90
1980	81,120	26.12
1970	43,233	13.92
1960	20,269	6.53
1950	7,794	2.51
1940	8,093	2.60
1930	4,717	1.52
1920	1,243	.40
1910	322	.10
1900	443	.14
1800	1,437	.46
<b>Total</b>	<b>310,621</b>	<b>100</b>



Gráfica 1.1 Tesis por Década

Puede observarse que en los últimos 12 años se poseen 141,950 tesis equivalentes al 45.70% del total. El siguiente gráfico muestra el número de tesis recibidas por año a partir de 1992 y hasta el año 2002.

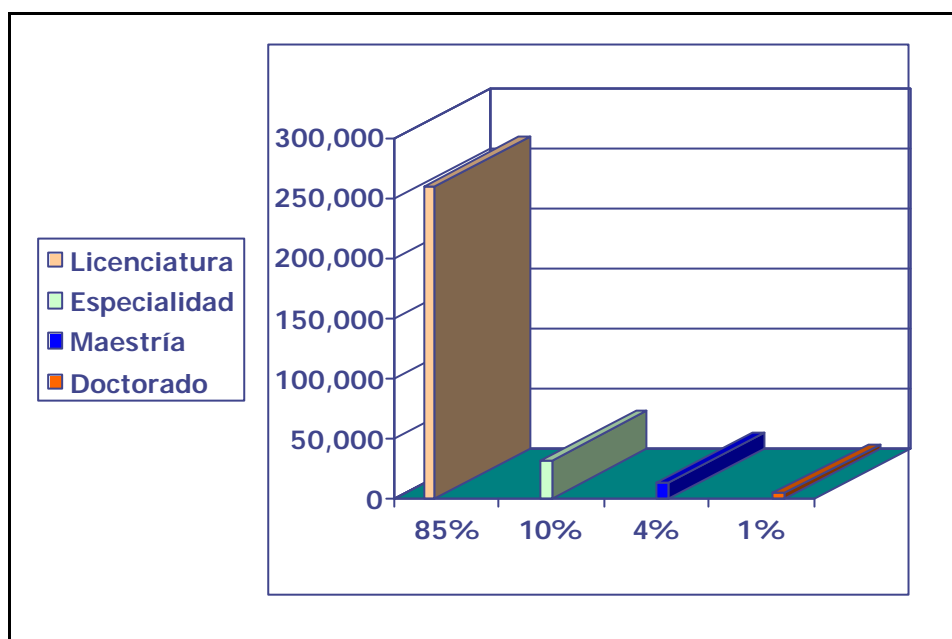
Año	Número	%
2002	14,668	4.72
2001	12,509	4.03
2000	10,115	3.26
1998	12,131	3.90
1997	11,363	3.66
1996	10,987	3.53
1995	11,516	3.71
1994	10,186	3.28
1993	10,518	3.39
1992	10,338	3.33
Otros	196,290	63.19
<b>Total</b>	<b>310,621</b>	<b>100</b>



Gráfica 1.2 Tesis por Año

### 1.2.3.3 Grado

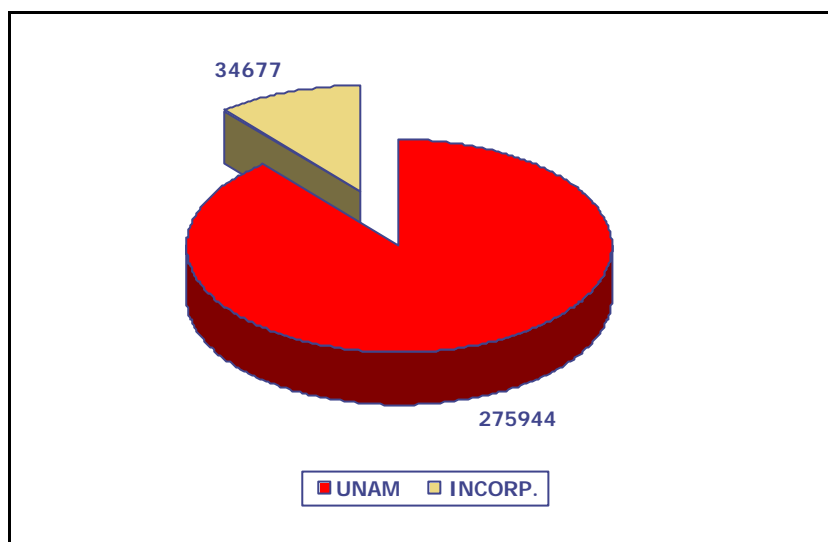
La distribución por grado es la siguiente: de las 310,621 tesis, 260,430 (83.84%) corresponden a licenciatura; 32,252 (10.38%), a especialidades (sobre todo en medicina); 13,416 (4.32%) a maestrías y 4,523 (1.46%) al doctorado.



Gráfica 1.3 Tesiunam-Grado

### 1.2.3.4 Procedencia

Las tesis de egresados de la UNAM representan el 89% con 275,944 trabajos recepcionales mientras que las tesis de egresados de 42 escuelas y universidades incorporadas representan el 11% con 34,677 tesis.

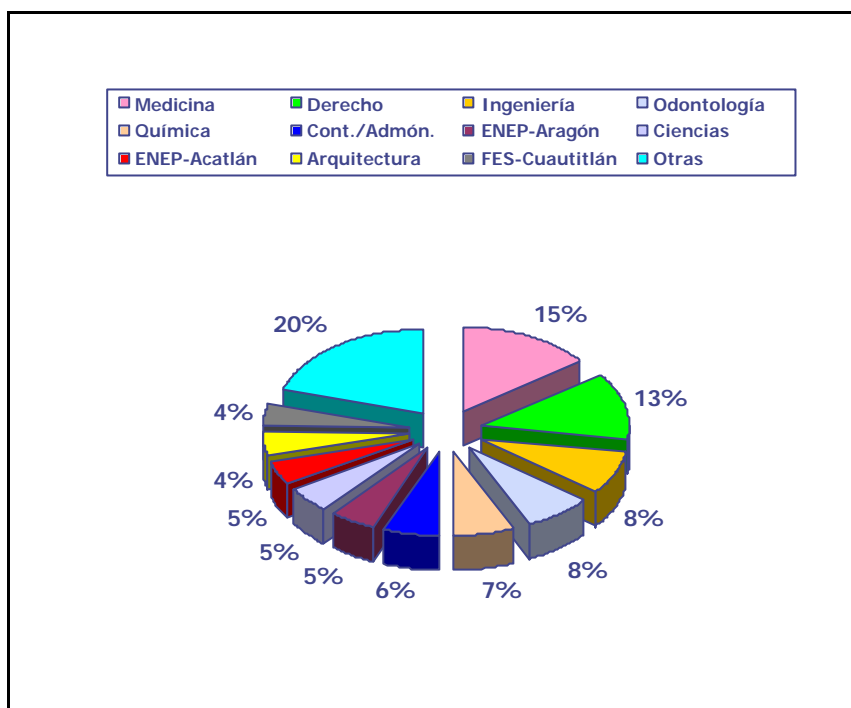


**Gráfica 1.4** Tesiunam-Procedencia

### 1.2.3.5 Escuelas y facultades

La distribución de tesis por escuelas y facultades de la UNAM es la siguiente:

Medicina con 40,262 (15%); Derecho con 35,364 (12%); Ingeniería con 598 (8%); Odontología con 21,171 (8%); Química con 18,521 (7%); Contaduría y Administración con 16,817 (6%); ENEP-Aragón con 13,580 (5%); Ciencias con 13,456 (5%); la ENEP-Acatlán con 13,630 (5%); Arquitectura con 12,276 (4%); FES-Cuautitlán con 10,866 (4%) y 58,043 (20%) tesis consignadas en otras 31 facultades.



Gráfica 1.5 Tesiunam-Escuelas y Facultades

La mayoría de las tesis corresponden a la década de los noventa con un 33%, y por grado las de licenciatura representan el 84%.

El 89% pertenecen a egresados de la UNAM, la Facultad de Medicina –una de las más antiguas de nuestra máxima Casa de Estudios está representada con el 15% de las tesis, seguida por las de Derecho con 12%, Ingeniería y Odontología con el 8% cada una.

### 1.3 Bibliotecas Digitales

Las bibliotecas digitales son las organizaciones que proporcionan los recursos, incluso el personal especializado, para seleccionar, estructurar, ofrecer acceso intelectual, interpretar, distribuir, preservar la integridad y asegurar la persistencia de las colecciones de obras digitales, de manera que éstas se encuentren rápida y económicamente disponibles para ser consultadas por una comunidad definida o por el conjunto de las mismas.



### 1.3.1 Características

- Son la cara digital de las bibliotecas tradicionales, que incluyen colecciones impresas, digitales y multimedia no electrónicas. Por lo tanto, abarcan ambos materiales, los impresos y los electrónicos.
- Cuentan también con materiales digitales que existen más allá de las fronteras físicas y administrativas de cualquier biblioteca digital.
- Incluyen los procesos y servicios que constituyen la columna vertebral y el sistema nervioso de las bibliotecas. Sin embargo, estos procesos tradicionales, a pesar de ser la base del trabajo de la biblioteca digital, tendrán que ser revisados y mejorados para adecuar las diferencias entre los nuevos medios digitales y los tradicionales.
- Idealmente, proporcionan una visión coherente de toda la información contenida en la biblioteca, sin importar su forma o formato.
- Sirven a comunidades u organizaciones determinadas, como lo hacen las bibliotecas tradicionales; no obstante, tales comunidades pueden estar ampliamente dispersas a través de la red.
- Para asegurar su existencia, requerirán de las habilidades tanto de bibliotecarios como de ingenieros en computación.

Algunos de los aspectos a los que se enfrenta el desarrollo de bibliotecas digitales son los siguientes:

### 1.3.2 Arquitectura Técnica

Incluye componentes tales como:

1. Redes de trabajo de alta velocidad y rápida conexión a Internet.
2. Bases de datos relacionales que soporten diversos formatos digitales.
3. Motores de búsqueda de texto completo que indiquen y proporcionen acceso a las fuentes.
4. Una variedad de servidores como los de la Web y los FTP.

5. Funciones de administración de documentos electrónicos, que ayudarán en el manejo integral de los recursos digitales.

### 1.3.3 Construcción de colecciones digitales

Para que cualquier biblioteca digital sea viable, debe contar con una colección digital relevante que sea útil. Principalmente hay tres métodos para conformar colecciones digitales:

1. **Digitalización**, que consiste en convertir las colecciones en papel y en otros medios en formato digital.
2. **Adquisición de obras digitales originales** creadas por editoriales y académicos, algunas podrían ser libros electrónicos, journals y conjuntos de datos.
3. **Acceso a materiales externos** que no se encuentran localmente a través de enrutamientos a sitios Web, colecciones de otras bibliotecas o servidores de editoriales.

La conformación coordinada de colecciones digitales mejora considerablemente la compartición de recursos, e incrementa la riqueza de las colecciones a las que los usuarios tienen acceso.

### 1.3.2 La solidez de la colección

Una biblioteca específica, con una colección muy especializada, podría ser la responsable de digitalizar porciones seleccionadas de ella y agregar nuevas obras digitales a la misma.

#### 1. Colecciones únicas

Si una biblioteca posee las únicas copias de algunas obras, éstas serán las más adecuadas para ser digitalizadas.

#### 2. Las propiedades de las comunidades de usuarios

Tales prioridades justificarían poseer los materiales localmente, debido a las necesidades de un plan de estudios.

### **3. Partes manejables de las colecciones**

Cuando no exista otro criterio, los materiales podrán distribuirse entre varias instituciones, estableciendo lo que sea razonable que cada un albergue o digitalice.

### **4. Arquitectura técnica**

La situación de la arquitectura técnica de la biblioteca también será un factor para decidir quién digitaliza qué. Una biblioteca debe contar con una arquitectura técnica acorde con las tareas de apoyo a una colección digital definida.

### **5. Habilidades del personal**

Las instituciones cuyo personal no cuente con las habilidades necesarias, no podrán ser un enlace significativo en un esquema de bibliotecas de esta naturaleza a nivel nacional.

Las bibliotecas, de manera colectiva, deben asegurarse de que lo que se esté haciendo sea conservado y esté disponible indefinidamente. Asegurar la conservación y el acceso a largo plazo requerirá de políticas y de un esquema por el cual, para su permanencia, copias repetidas sean almacenadas en instituciones designadas para ese propósito.

### **1.3.5 Copyright**

El copyright ha sido llamado “la barrera más vejatoria para el desarrollo de la biblioteca digital” (Chepesuk, 1997:49). Su concepto actual, basado en la literatura impresa, se viene abajo en el ambiente digital debido a que se pierde el control sobre las copias. Los objetos digitales son menos rígidos, más fáciles de ser copiados y accesibles a distancia por muchos usuarios de forma simultánea. El problema es que, a diferencia de las empresas privadas o las editoriales que poseen su propia información, las bibliotecas son, en su mayor parte, sólo preservadoras de ella y no poseen el copyright del material que albergan. Es poco probable que las bibliotecas sean alguna vez plenamente capaces de digitalizar y proporcionar acceso libre a los materiales de

sus colecciones, amparados por las leyes de autoría. Por el contrario, tendrán que desarrollar mecanismos, para la administración del copyright, mismos que les permitan proporcionar la información sin infringir derechos de autor.

Algunas funciones que la administración de derechos de autor podría incluir son:

1. Rastrear patrones de uso.
2. Identificar y validar usuarios.
3. Proporcionara la situación de copyright de cada objeto digital, las restricciones para sus uso y las tarifas pertinentes.
4. Administrar las transacciones de los usuarios para permitirles el acceso sólo a determinado número de copias, cobrarles por cada una o bien trasladar la solicitud de consulta a una editorial.

### 1.3.6 Preservación

Otro aspecto importante es mantener la información digital disponible perpetuamente. En la preservación de los materiales digitales lo que verdaderamente importa es la obsolescencia técnica, que en la era digital es semejante al deterioro del papel en el mundo impreso. Las bibliotecas en la era predigital se preocupaban por el control del clima y la desacidificación de los libros, pero la preservación de la información digital significará encontrar constantemente nuevas soluciones técnicas.

Hay tres formas para preservar a los materiales digitales:

1. **La preservación del medio de almacenamiento.** Los procedimientos utilizados para almacenar materiales digitales se vuelven obsoletos de dos a cinco años antes de ser reemplazados por una mejor tecnología. A largo plazo, los materiales almacenados en medios antiguos podrían perderse debido a que en poco tiempo no existirían el hardware ni el software para leerlos. Así, las bibliotecas tendrán que estar trasladando su información digital de un medio de almacenamiento a otro.
2. **La preservación de acceso al contenido.** La solución al problema de la preservación del acceso al contenido podría resolverse trasladando los datos; esto es, migrar los datos de un formato a otro para asegurar que

los usuarios puedan desplegar su contenido. Sin embargo, en ello también se presentan dificultades: trasladar los datos es costoso, no existen normas para dicha transportación, y distorsionar o perder información resulta inevitable cada vez que se realiza un traslado de un formato a otro. Incluso, si hoy en día se contara con las tecnologías adecuadas para migrar objetos digitales completos, la información tendría que ser trasladada de formato a formato por muchos años, lo que heredaría una enorme y cara responsabilidad a las generaciones venideras.

3. **La preservación de materiales en medios fijos a través de la tecnología digital.** Todavía no existen normas para el uso de medios digitales como técnica de preservación, y tampoco queda claro si éstos están listos para la tarea de la preservación a largo plazo. Las normas de preservación serán necesarias para almacenar y compartir de manera adecuada los materiales preservados digitalmente.

Las bibliotecas podrían crear un esquema coordinado que abarque:

- Políticas para la preservación a largo plazo.
- Aseguramiento de copias permanentes que sean almacenadas en instituciones ex profeso.
- Ayuda y establecimiento de normas de preservación para almacenar y compartir de manera adecuada los materiales preservados digitalmente.

#### 1.4 Definición del problema

Uno de los problemas que tiene en la actualidad la Biblioteca Central, en específico el departamento de tesis, es que sufre de espacio físico debido a que se reciben alrededor de 12,000 tesis anualmente.

El proceso de registrar las tesis en la base de datos (tesiunam) es tardado y se necesita de personal para dicha tarea. Además de utilizar personal para atención a los usuarios que necesitan de consultar una tesis.

Dada la gran diversidad de la comunidad universitaria la Dirección General de Bibliotecas se ve en la necesidad de proporcionarle cada día mejores servicios en

cuanto a consulta de información y en cuanto a la agilización de trámites para titulación.

Por lo tanto se ha dado a la tarea de resolver algunos de los problemas que enfrentan los tesisistas, como son:

- El costo de titulación aumenta debido a que tiene que entregar dos ejemplares más en BC a parte de los que ya tuvo que haber entregado en su institución.
- La difusión de las tesis es tardado y solamente se puede hacer por medio de la Biblioteca.

### **1.5 Objetivos**

Desarrollar un sistema integral capaz de implementar y automatizar el proceso de registro, almacenamiento, recuperación y consulta de las tesis a través de Internet.

Desarrollar un sistema eficaz, para cumplir con las necesidades de consulta de texto completo de la comunidad universitaria.

Agilizar los trámites de titulación del tesisista y solucionar el problema de espacio físico que existe en el Departamento de tesis de la Biblioteca Central.

### **1.6 Alcances**

En general el sistema esta diseñado para dar soporte a dos partes muy importantes. Uno a los tesisistas de la Universidad y otro a los trabajadores de la misma.

Se tiene pensado que en un futuro no muy lejano todos los tesisistas se registren por internet y que manden sus archivos de PDF por medio de correo electrónico ahorrando con esto tiempo, dinero y esfuerzo. Además tendrán la oportunidad de consultar sus tesis en muy poco tiempo por medio de Internet de forma segura y agradable. Prácticamente todo el mundo podría tener acceso a esta información sin el riesgo de que se violen derechos de autor.

En cuanto a los trabajadores de la Biblioteca central (Departamento de tesis) el tiempo que dedicaban en el registro podrían utilizarlo en otras actividades además de que los errores de captura serían mínimos.

## **CAPÍTULO 2**

---

---



## **INGENIERÍA DE SOFTWARE**

### **2.1 Aplicaciones del software**

Para el estudio de las aplicaciones del software es necesario establecer categorías genéricas, ya que conforme aumenta su complejidad es más difícil de encasillar en una u otra categoría. De esta manera se pueden mencionar las siguientes:

- *Software de Sistemas*: Conjunto de programas creado como herramienta para otros programas (S.O's, Compiladores, CASE, Editores, etc.). Se caracteriza por existir una fuerte interacción con el hardware de la computadora.
- *Software de tiempo real*: Conjunto de programas que mide, analiza y controla sucesos del mundo real a medida que ocurren. Se caracterizan por los siguientes componentes:
  1. Adquisición de Datos: recoge y formatea los datos que le entran.
  2. Análisis: transforma la información dependiendo de la aplicación.
  3. Control/Salida: responde a la entrada (entorno externo).
  4. Monitorización: coordina el resto de los componentes, de forma que pueda mantenerse la respuesta en tiempo real (entre 1 ms y 1 s).
- *Software de gestión*: Realiza tareas de procesamiento de datos y operaciones de cálculo interactivo (aplicaciones de gestión de empresas, etc.).
- *Software de Ingeniería y Científico*: Utiliza algoritmos de manejo de números (Astronomía, Cálculo, Biología molecular, Fabricación automática, etc.). El CAD y la simulación de sistemas están tomando características de software en tiempo real e incluso de software de sistema.
- *Software empotrado*: Reside en memorias ROM y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo. (Lavadoras, Microondas)

- *Software de PC's*: Ejemplos: procesadores de textos, hojas de cálculo, gestores de Bases de Datos, juegos, aplicaciones financieras, etc.
- *Software de inteligencia artificial*: Es aquel que hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados el cálculo o el análisis directo. Forman parte de este grupo, aplicaciones de sistemas expertos, reconocimiento de patrones (OCR), redes neuronales artificiales, prueba de teoremas, juegos, etc.

## 2.2 Ingeniería de software

El término Ingeniería de Software (Isw) se introdujo por primera vez a finales de la década de 1960 en una conferencia celebrada para analizar la llamada "crisis de el software", que fue resultado directo de la aparición del hardware de computadoras de la tercera generación. Se propusieron varias definiciones pero sus elementos comunes son que la ingeniería de software trata de la construcción de sistemas que son más grandes de lo que normalmente podía manejar un individuo, utiliza principios de ingeniería para el desarrollo de estos sistemas y tiene aspectos técnicos y no técnicos.

Algunas de estas definiciones son:

"La Isw: es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales" (Fritz Bauer).

"La Isw es la disciplina que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza" (Sommerville 2002).

La Ingeniería del Software abarca tres elementos clave:

1. Métodos.
2. Herramientas.

### 3. Procedimientos.

Los cuáles facilitan el proceso del desarrollo del software y permiten establecer unas bases para construir software de alta calidad de una forma productiva.

Los **Métodos** nos enseñan cómo construir técnicamente el Software. Estos métodos abarcan:

- Planificación y estimación de proyectos
- Análisis de requisitos del sistema y del software
- Diseño de estructuras de datos
- Diseño de programas y procedimientos algorítmicos
- Codificación
- Prueba
- Mantenimiento

Las **Herramientas** son el soporte automático o semiautomático de aplicación de los métodos. Existen herramientas para cada uno de los métodos anteriores. Cuando la información creada por una herramienta puede ser utilizada por otra se establece un sistema para el desarrollo del software llamado Ingeniería del Software Asistida por Ordenador (CASE: Computer Aided Software Engineering). Cada método tiene su propia herramienta.

Aplicamos varios pasos para crear un producto de software. La salida de uno es la entrada de otro.

Los **Procedimientos** son la aplicación de los métodos utilizando las herramientas. Son los elementos de unión entre los métodos y las herramientas. Definen la secuencia en la que se aplican los métodos, ayudan a asegurar la calidad y coordinar los cambios, etc.

*La Ingeniería del Software está compuesta por una serie de pasos que incluye los métodos, herramientas y procedimientos: Análisis, Diseño y Codificación*

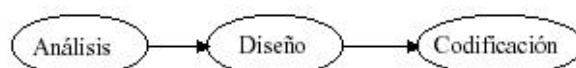


Figura 2.1 Modelo General de Ingeniería de Software

### 2.3 Modelos de procesos de software

Definición de Procesos de Software.

Es un grupo de actividades cuya meta es el desarrollo o evolución de software. Los procesos genéricos de todos los procesos de software son:

- Especificación -- qué debe hacer el sistema y las restricciones de desarrollo.
- Desarrollo -- la producción del sistema de software.
- Validación -- asegurando que el software es lo que el cliente desea.
- Evolución -- cambiando el software como respuesta a las demandas cambiantes.

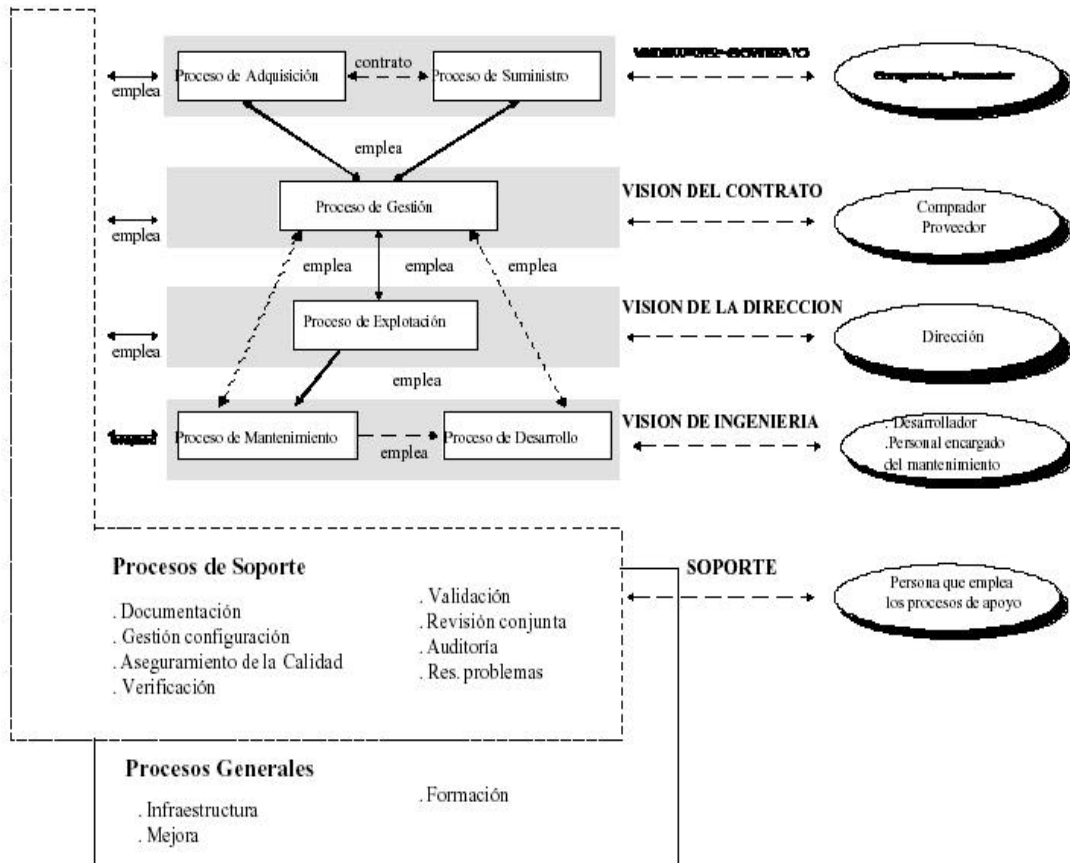


Figura 2.2 Visión general de proceso de software.

### 2.3.1 Modelo Lineal Secuencial

Es un refinamiento altamente influenciado por el modelo de etapas. Existe, para este modelo, un reconocimiento de los ciclos de retroalimentación entre etapas, y una guía para confinar las retroalimentaciones a las etapas sucesivas con el objetivo de minimizar el costo del retrabajo involucrado en retroalimentaciones a través de muchas etapas.

Este modelo considera las siguientes partes:

1. **Planificación del sistema.**
2. **Análisis.**
3. **Diseño.**
4. **Codificación.**
5. **Pruebas.**
6. **Mantenimiento.**

En ocasiones se denomina de cascada porque los productos pasan de un nivel a otro con facilidad. Este es el ciclo de vida clásico y más antiguo, usado en el desarrollo de productos de software.

#### ***Planificación del Sistema:***

Es la etapa en la que se determina si el proyecto es o no factible de realizar y se determinan tiempos y costos aproximados, estableciendo así la ruta crítica de cada actividad. Esto es porque la falta de planeación de un sistema es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de software.

Uno de los principales propósitos de esta fase es aclarar los objetivos, problemas o necesidades y restricciones.

#### ***Análisis:***

Para comprender la naturaleza del programa a construir, el ingeniero debe comprender el dominio de información del software, así como la función requerida, su comportamiento, rendimiento e interconexión.

Es indispensable comprender perfectamente los requisitos del software, para que éste no fracase. Pero abundan los casos en que se puede llegar a malas interpretaciones o falta de información.

**Diseño:**

El diseño del software es realmente un proceso multipaso que se enfoca sobre cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

**Codificación:**

El diseño debe traducirse en una forma legible para la máquina. Si el diseño se realiza de una manera detallada, la codificación puede realizarse mecánicamente, para desarrollar con una mejor lógica la programación.

**Pruebas:**

Una vez que se ha generado el código, comienza la prueba del programa. La prueba se centra en la lógica interna del software, asegurando que todas las sentencias se han probado, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

En el desarrollo de un sistema grande, las pruebas incluyen varias etapas la primera incluye la prueba conocida como *prueba de módulo*, componente o unidad en la cual, se verifica que el componente funciona apropiadamente con los tipos de entrada especificados en la etapa del diseño.

La siguiente etapa incluye la *prueba de integración* que es el proceso de verificación de que los componentes del sistema trabajan juntos como esta descrito en las especificaciones del diseño del programa.

Posteriormente se llevan a cabo las *pruebas de funcionamiento* que evalúan que el sistema determine si las funciones descritas en la especificación de requerimientos son realmente desarrolladas por el sistema integrado. El resultado es un sistema funcional.

Después, se continúa con una *prueba de desempeño* que compara con el resto de esos requerimientos de hardware y software. De una prueba exitosa de este tipo resulta un sistema validado.

A continuación se desarrolla una *prueba de aceptación* en donde el sistema es checado con respecto a la descripción de requerimientos del cliente. Hasta que este completa la prueba de aceptación se puede instalar el sistema en el ambiente en el cual será usado; una *prueba final de instalación* se lleva a cabo para asegurarnos de que el sistema todavía funciona como debería.

### **Mantenimiento**

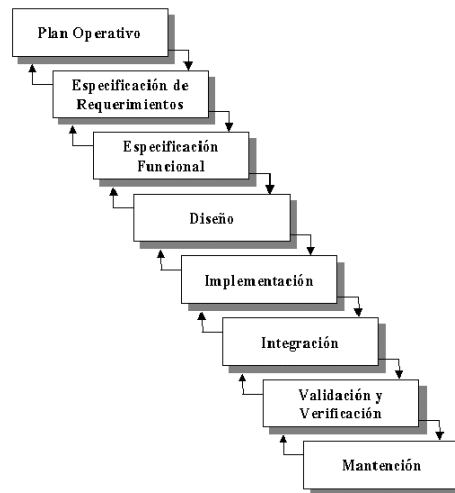
Es indudable que el software una vez entregado al cliente sufrirá cambios (posible excepción es el software empotrado). Los cambios ocurrirán debido a que se hayan encontrado errores, a que el software deba adaptarse a posibles cambios. Existen cuatro tipos de mantenimiento que son:

*Correctivo*: un programa no realiza correctamente la aplicación para la que ha sido diseñado, y, por tanto, debe ser modificado.

*Perfectivo*: modificaciones a los programas para conseguir mayor adecuación a los requisitos, mayor eficiencia, o simplemente recoger nuevas funcionalidades no expresadas en la fase de definición del sistema

*Adaptativo*: Adaptar los programas para acomodarlos a los cambios de su entorno externo (modificaciones en la legislación, CPU, SO, las reglas de negocio, etc.)

*Preventivo*: El software se deteriora con los cambios, y este tipo de mantenimiento hace cambios en los programas para que se puedan corregir, adaptar y mejorar más fácilmente (**Reingeniería del software**).



**Figura 2.3** Representativo del Modelo Lineal Secuencial (Cascada)

Las principales ventajas de este modelo son:

- Es muy útil ya que ayuda a los desarrolladores a determinar exactamente lo que necesitan hacer en cada una de las etapas del desarrollo del sistema.
- Su simplicidad facilita a los desarrolladores explicar a los clientes todo lo relacionado al desarrollo del sistema, ya que ellos no están familiarizados con el tema.
- Hace explícitos los productos intermedios que son necesarios en orden de empezar la siguiente etapa del desarrollo.

El principal problema con este modelo es que no refleja la manera en la que se codifica el sistema de una manera real. Sin embargo este problema se soluciona con un buen entendimiento de los problemas debido a que de esta manera el sistema se desarrolla con un buen grado de iteración.

Por otro lado, el modelo de prototipos (que se analiza en la siguiente sección) funciona solo como un subproceso en el modelo secuencial. Debido a que un prototipo es un producto parcialmente desarrollado que nos permite examinar algunos aspectos del sistema propuesto y decidir si es apropiado o no para el producto terminado.



### **2.3.2 Modelo de Prototipos**

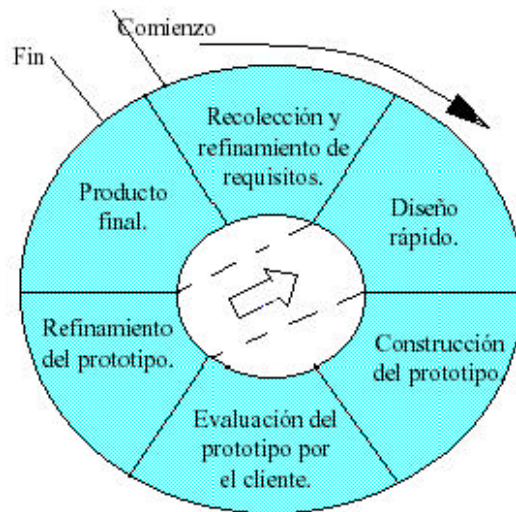
Un prototipo es una representación o modelo del producto de programación que, a diferencia de un modelo de simulación, incorpora componentes del producto real. Por lo regular, un prototipo tiene un funcionamiento limitado en cuanto a capacidades, confiabilidad o eficiencia.

Hay varias razones para desarrollar un prototipo; una de ellas es ilustrar los formatos de datos de entrada, mensajes, informes y diálogos al cliente, este es un mecanismo adecuado para explicar opciones de procesamiento y tener un mejor entendimiento de las necesidades de él.

Los pasos a seguir en este ciclo son:

1. El técnico informático y el cliente se reúnen; se definen los objetivos globales para el software, se identifican los requisitos y se perfilan las áreas que requieren mayor definición.
2. Se hace un diseño rápido, enfocado a la representación de los aspectos del software visibles al usuario, por ejemplo los mecanismos de entrada y formatos de salida.
3. Se construye un prototipo.
4. El prototipo es evaluado por parte del cliente o el usuario, para refinar los requisitos del software a desarrollar.

Idealmente el prototipo sirve como mecanismo para identificar los requisitos del software. Realmente el primer sistema construido no es utilizable., ya que suele estar sin depurar y su calidad suele ser muy baja.



**Figura 2.4** Modelo de Prototipos

Al igual que el ciclo de vida clásico, la construcción de prototipos puede ser problemática por las siguientes razones:

- 1.- El cliente ve funcionando lo que parece ser una primera versión del software, ignorando que, por hacer que funcione, no hemos considerado los aspectos de calidad o de mantenimiento del software a largo plazo. Cuando se le informa de que el producto debe ser reconstruido, el cliente solicita que se apliquen "cuantas mejoras" sean necesarias para hacer del prototipo un producto final que funcione. Y el desarrollador del producto cede demasiado a menudo.
- 2.- El técnico de desarrollo, frecuentemente, impone ciertos compromisos de implementación con el fin de obtener un prototipo que funcione rápidamente. Puede que utilice un sistema operativo inapropiado, o un lenguaje de programación equivoco o simplemente porque ya está disponible y es conocido, puede que implemente ineficientemente un algoritmo, sencillamente para demostrar su capacidad.

La clave está en definir al comienzo las reglas del juego; esto es, el cliente y el técnico deben estar de acuerdo en que el prototipo se construya para servir sólo como un mecanismo de definición de los requisitos

### 2.3.3 Modelo en espiral

El Modelo Espiral de Boehm para Ingeniería de Software agrupa las mejores características del modelo del ciclo de vida clásico y de prototipos. Pero también agrega nuevas funciones que no están incluidas en los otros modelos, como el análisis de riesgo.

Los pasos a seguir en este modelo son:

1. **Planificación:** Se determinan los objetivos, alternativas y restricciones.
2. **Análisis de riesgo:** Se identifican los riesgos, analizando alternativas y resolución de los mismos.
3. **Ingeniería:** Se desarrolla el producto del siguiente nivel.
4. **Evaluación del cliente:** Se evalúan los resultados del paso anterior.

A cada vuelta de la espiral, se construyen sucesivas versiones del software, cada vez más depuradas y completas, hasta llegar a obtener el producto final. Con este modelo nos ayudamos con un prototipo para verificar el sistema, y si los riesgos encontrados durante el desarrollo son muy grandes, se puede dar por terminado el proyecto. Este método utiliza un enfoque evolutivo, permitiendo al técnico informático y al cliente entender y reaccionar a los riesgos aparecidos a cada nivel de desarrollo.

## Modelo de Proceso de Espiral

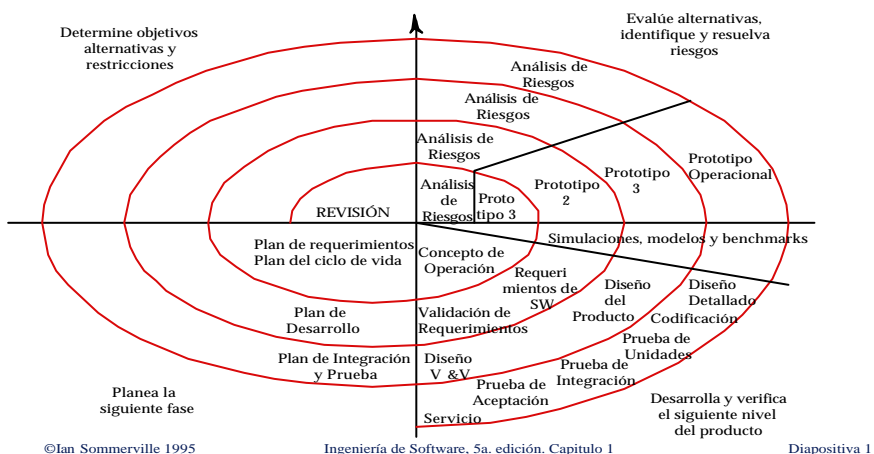


Figura 2.5 Modelo en Espiral

### 2.3.4 Otros modelos

#### EL PROTOTIPADO EVOLUTIVO.

Este modelo tiene las siguientes características primordiales:

- Construcción de una implementación parcial que cubre los requisitos conocidos, para ir aprendiendo el resto y, paulatinamente, incorporarlos al sistema.
- Reduce el riesgo y aumenta la probabilidad de éxito.
- No se conocen niveles apropiados de calidad y documentación.
- Problemas de gestión de configuración.
- Construir software para que pueda ser modificado fácilmente es un "arte desconocido".

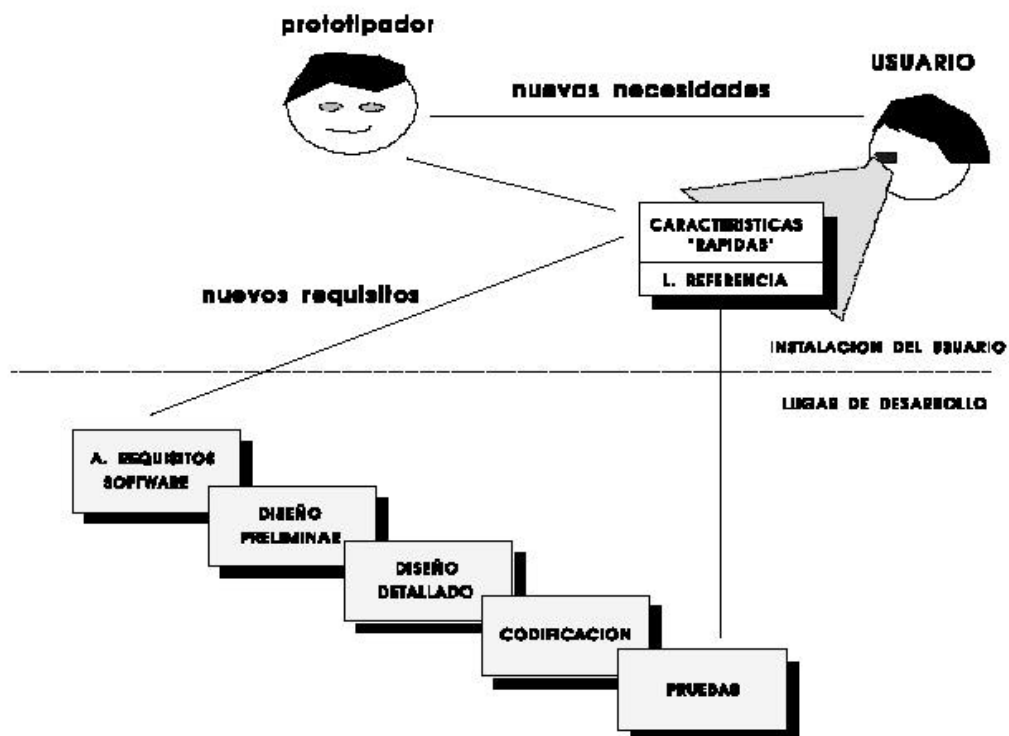


Figura 2.6 Modelo prototipado operacional

EL MODELO DE DESARROLLO EVOLUTIVO

El desarrollo evolutivo es una metodología de desarrollo de software muy relacionada con, pero claramente distinta de, el desarrollo por prototipos. El énfasis está puesto sobre la importancia de obtener un sistema de producción flexible y expandible. Así, si los requerimientos cambian durante el desarrollo del sistema, entonces con un mínimo de esfuerzo y tiempo se puede desarrollar un sistema de trabajo flexible.

La diferencia fundamental entre desarrollo evolutivo y prototipos de software es que el desarrollo evolutivo busca reemplazar el viejo sistema con uno nuevo que tendría la propiedad de satisfacer los nuevos requerimientos lo más rápido posible. En contraste, prototipos usa un enfoque iterativo solo para determinar los requerimientos organizacionales. Por lo tanto el tiempo tomado entre cada iteración es mucho más importante para el desarrollo evolutivo. En la siguiente figura se puede ver gráficamente esta diferencia.

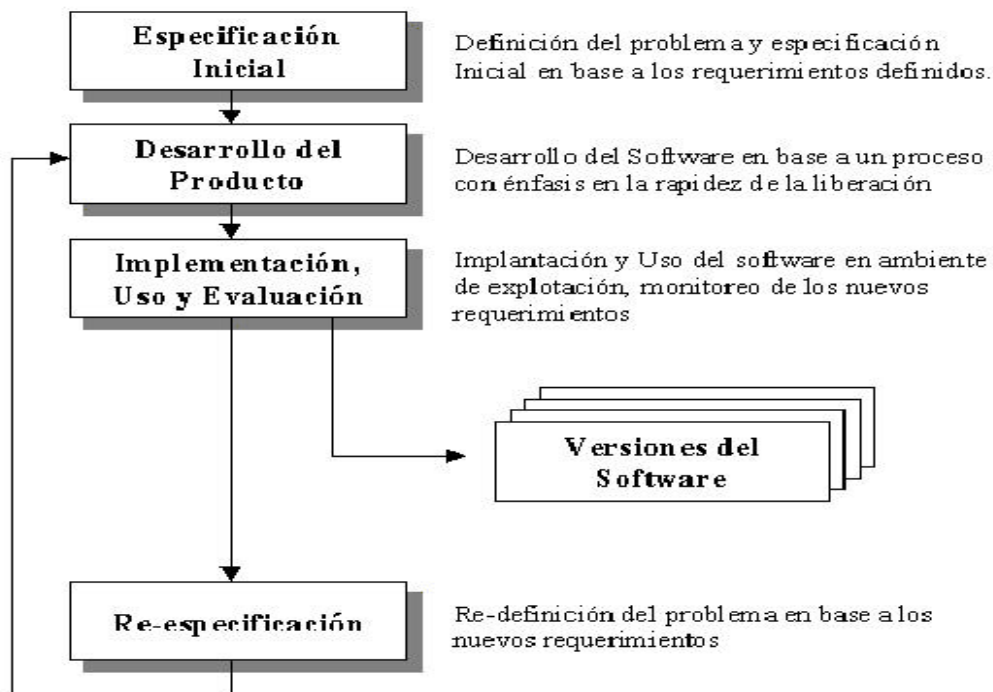


Figura 2.6 Modelo Desarrollo Evolutivo

El desarrollo evolutivo asume que los requerimientos de un proyecto están sujetos a cambios continuos, por lo cual es necesario definir una estrategia de desarrollo que refleje esta situación. En cambio, el desarrollo orientado a prototipos, así como los anteriores, asume que los requerimientos "reales" existen y se vale de las iteraciones del prototipo para establecerlos y modelarlos.

La idea entonces de la metodología de desarrollo evolutivo es estar liberando constantemente una nueva versión del sistema que sea completamente funcional; así, cada sistema producto de las iteraciones sucesivas del método tendría incorporado los nuevos requerimientos que ha sido posible identificar y que no estarían considerados en la anterior versión.

Así, las etapas del desarrollo evolutivo tienen por objetivo extender los incrementos de un producto de software operacional, en las direcciones determinadas por la evolución de la experiencia operacional.

El modelo de desarrollo evolutivo puede ser idealmente asociado a un lenguaje de aplicación de cuarta generación y mejor aún a situaciones en que el usuario dice, "yo no puedo hablarte sobre lo que yo quiero, pero yo lo reconocería si lo viese". Así, este método entregaría al usuario rápidamente una capacidad operativa inicial y, además, establecería una base real para determinar las mejoras subsecuentes en el producto.

Pero, existirían algunas dificultades técnicas que no pueden dejar de ser mencionadas, por ejemplo:

- No facilita la integración de aplicaciones que han sido desarrolladas como sistemas independientes.
- Facilita la posibilidad de que existan casos de "esclerosis de información", en el sentido que trabajos temporales alrededor de algunas deficiencias del software se solidifican como poderes inmodificables a la evolución. Es decir, en la medida que se evoluciona, esta misma facilidad a la evolución llevaría a que no sea posible seguir evolucionando.
- Pueden ocurrir que el software nuevo es un reemplazo incremental de un subsistema dentro de un gran sistema existente. Si el sistema existente está

pobremente modularizado, entonces es obvia la dificultad en hacer que la nueva versión se acople con facilidad al resto.

- El método evolutivo tiene la gran ventaja de reconocer la existencia de una constante de cambios en los requerimientos y, desde esta premisa, propone una solución, la cual es válida para la solución de ese problema pero que no resolvería la inquietud original, esto es que el método no facilita elementos que permitan reducir la distancia conceptual entre los dominios del desarrollador y del usuario. Con la existencia del método evolutivo se configura una nueva problemática en el desarrollo de sistemas, es decir, la crisis se expande ahora en el sentido que no sólo se requiere reflejar lo más fielmente posible las necesidades del usuario, sino que ahora los ambientes en que el sistema está inserto están sujetos a cambios y estos cambios inciden en la efectividad del software desarrollado.

#### MÉTODO DE SISTEMAS "SOFT"

Checkland desarrolla la metodología de sistemas "soft" en respuesta a las fallas de los enfoques más convencionales para abordar problemas "espinudos", problemas que son complicados de definir, conocidos como problemas "soft". Los problemas "soft" son encontrados frecuentemente en organizaciones y no pueden ser resueltos por las mismas técnicas usadas en resolver problemas más formales.

La metodología de sistemas "soft" considera el término "el problema" como inapropiado, considera que es una visión errada de la situación. Los sistemas "soft" reconocen que no existe un problema, pero si muchos problemas, lo que se describe como una "situación problema". El énfasis es puesto más sobre la caracterización del problema que sobre cómo se resuelve. Así, las metas del sistema "soft" son mucho más complejas que las metas de los sistemas tradicionales que pueden ser alcanzadas y medidas.

El proceso involucrado en la metodología de sistemas "soft" es el siguiente:

1. El analista (en adelante el *solucionador* del problema), con la ayuda de los administradores y usuarios dentro de la organización (el *dueño* del problema), crean un cuadro de la situación problema. El cuadro es una percepción subjetiva del problema en un esquema informal. Incluirá sistemas de la organización bajo revisión,

la gente dentro de ello, las tareas ejecutadas, etc. Este cuadro es usado como ayuda a las discusiones con el o los dueños del problema.

2. Los temas problemas son extraídos desde el cuadro por el *solucionador*. Él agregará, desde su óptica propia, los conflictos entre el personal o funciones dentro de la organización, problemas de comunicación u otro tipo de problemas que detecte. Este cuadro es utilizado para identificar problemas e informa al *dueño* de la situación, antes que se desarrollen posibles soluciones. El uso de este cuadro es mucho mejor que los métodos tradicionales para estimular a administradores y usuarios para revelar los problemas "reales" dentro de la organización.

3. La definición raíz es creada usando una lista de chequeo llamada el criterio CATWOE (Clients, Actors, Transformations, WorldView, Owner, Environment). Que por orden de importancia debería ser TWECO. Ellos determinan quién está haciendo qué para quienes, y a quienes ellos están preguntando. Qué supuestos están siendo usados y en qué medio ambiente está sucediendo todo aquello.

Si existen muchas versiones diferentes del cuadro, entonces serán creadas tantas definiciones raíz como versiones existan.

4. Cuando los dueños y los solucionadores del problema estén satisfechos en que las definiciones raíz están bien formadas, se crea un modelo conceptual. Este describe en forma gráfica que es lo que el sistema hará. El modelo conceptual tomará en cuenta todas las definiciones raíz consideradas.

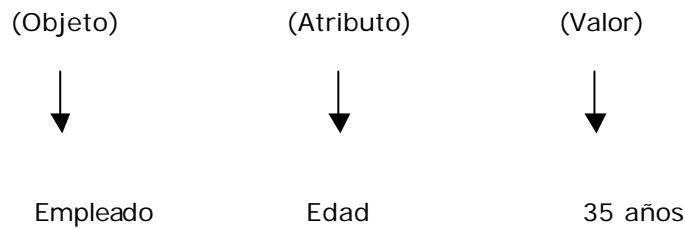
5. El modelo conceptual es usado como la base de todos los cambios hechos por el sistema para eliminar los problemas.

Como esta metodología no considera cualquier procedimiento preconcebido, el análisis es mucho mejor y permite un mejor entendimiento del problema.

## 2.4 Conceptos básicos de bases de datos

Un **dato** es la unidad mínima de información. Un dato, dentro de una base de datos, responde a la función:





La **información** es un conjunto de datos interrelacionados entre sí, que tienen un significado del cuál se puede obtener conocimientos para una futura toma de decisiones.

**Campo** es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto.

**Registro** es la colección de campos de iguales o de diferentes tipos.

**Archivo** es el nombre que recibe la colección de registros almacenados siguiendo una estructura homogénea.

Definición de **Base de Datos**:

- Una base de datos es un conjunto de datos relacionados entre sí con un objeto común.
- Una base de datos puede tener más de un uso y los múltiples usos pueden satisfacer múltiples "enfoques" de los datos almacenados.

Una base de datos debe cumplir con las dos condiciones siguientes:

1. Los datos son compartidos por diferentes usuarios y programas de aplicación; existe un mecanismo común para inserción, actualización, borrado y consulta de los datos.
2. Tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento.

Un **Administrador de una Base de Datos** (DBA) es la persona que tiene el control centralizado sobre el sistema de base de datos y controla tanto los datos como los programas que tienen acceso a ellos. Sus principales funciones se puede enumerar como sigue:

- Decidir el contenido de la base de datos.
- Crear la estructura de almacenamiento y los métodos de acceso.
- Administrar y controlar la seguridad física y lógica de la base de datos.
- Monitorear el comportamiento y crecimiento de la base de datos.
- Procedimientos de respaldos y depuración de la base de datos.
- Salvaguardar la documentación, respaldos y diccionario de datos tanto de la base de datos como de sus aplicaciones.
- Procedimientos de contingencia y recuperación de la base de datos.
- Modificar la base de datos o la descripción de la organización física.
- Otorgar permisos de acceso y prioridades a los diferentes usuarios.
- Especificar las limitaciones de integridad.
- Ser el enlace con los usuarios.

#### **2.4.1 Objetivos de las Bases de Datos.**

Los objetivos principales de un sistema de base de datos es disminuir:

##### **Redundancia:**

- La redundancia de datos se refiere, a la existencia de información repetida o duplicada en diferentes tablas dentro de una base de datos.
- La redundancia conduce a muchos problemas que tienen que ver con la integridad y la consistencia de los datos. La redundancia de los datos requiere múltiples procedimientos de entrada y actualización de los mismos.
- Dentro de una base de datos relacional la redundancia debe ser mínima y controlada.

##### **Integridad:**

- La integridad de una base de datos se refiere no solo a que los datos sean consistentes dentro de la base, sino además, que los valores que posean los

datos sean válidos de acuerdo a las dependencias funcionales entre tablas y de acuerdo a las políticas de negocio.

- La integridad de la base de datos se puede lograr:
  1. Manteniendo una redundancia mínima y controlada.
  2. Estableciendo llaves primarias.
  3. Estableciendo reglas de validación durante la creación y edición de los datos.
  4. Estableciendo procedimientos que validen la dependencia funcional entre tablas relacionadas (integridad diferencial).

**Consistencia:**

- Frecuentemente los problemas de consistencia de datos se deben a la redundancia de éstos.
- Es muy probable que surjan incongruencias al almacenar la misma información en más de un lugar; ya que al modificar, eliminar o agregar un dato, en esas condiciones, debe realizarse en cada una de las instancias del mismo con el riesgo de no realizarlo en su totalidad, generando en este caso datos inconsistentes.

**Seguridad:**

- Hoy en día se considera a la información de una empresa como uno de los activos más valiosos, por lo que la seguridad de la misma es muy importante.
- La seguridad de una base de datos se refiere principalmente al control de acceso, modificación y definición, tanto de los datos como de la estructura de la base de datos, a los diferentes usuarios de la misma.
- Algunos sistemas operativos proporcionan algún nivel de seguridad en el control de acceso a usuarios, sin embargo ésta debe radicar principalmente en el DBMS o en la aplicación que maneje la base de datos.

Además, es recomendable considerar las siguientes cuestiones:

*Seguridad de Objetos:* Se refiere a los permisos de los diferentes usuarios para poder hacer uso de tablas, procedimientos almacenados, triggers, etc.

*Seguridad de operaciones:* Aquí se manejan permisos para poder modificar (insertar, borrar, actualizar) la base de datos.

**Independencia de datos:**

- Se conoce como independencia de datos a la capacidad para utilizar la base de datos sin conocer los detalles de su representación conceptual (*Independencia lógica de datos*) o de su almacenamiento físico en el hardware (*Independencia física de datos*).
- La independencia de datos depende tanto del diseño de la base de datos como del sistema manejador de base de datos. Actualmente los DBMS no permiten lograr una completa independencia de datos.

Mediante la *independencia lógica de datos*, el programa de aplicación puede cambiar sin afectar a los datos almacenados.

Mediante la *independencia física de datos*, las aplicaciones permanecen inalteradas sin importar los cambios efectuados en el almacenamiento o en los métodos de acceso.

**2.4.2 Modelos de Datos.**

Modelo es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación se elabora de forma gráfica. Y Modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. Se dividen en tres grupos:

1. Modelos lógicos basados en objetos.
2. Modelos lógicos basados en registros.
3. Modelos físicos de datos.

**MODELOS LÓGICOS BASADOS EN OBJETOS.**

Se usan para describir datos en los niveles conceptual y de visión, es decir, con este modelo se representan los datos en la forma como los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permite especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

*Modelo Entidad-Relación.*

Denominado por sus siglas como: E-R; Este modelo representa a la realidad a través de entidades, que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como lo son: el nombre o el número de control asignado al entrar a una institución educativa; así mismo, un empleado, una materia, etc. Las entidades pueden ser de dos tipos:

Tangibles: Son todos aquellos objetos físicos que podemos ver, tocar o sentir.

Intangibles: Todos aquellos eventos u objetos conceptuales que no podemos ver, aún sabiendo que existen, por ejemplo: la entidad materia, sabemos que existe, sin embargo, no la podemos visualizar o tocar.

Las características de las entidades en base de datos se llaman atributos, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno; clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de relaciones.

*MODELOS LÓGICOS BASADOS EN REGISTROS.*

Se utilizan para describir datos en los niveles conceptual y físico. Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de la base de datos y para proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son:

1. Modelo Relacional.
2. Modelo de Red.
3. Modelo Jerárquico.

*Modelo relacional.*

En este modelo se representan los datos y las relaciones entre estos a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

Características de una tabla.

- Cada tabla debe tener un nombre único.
- El orden de los renglones no es significativo.
- El orden de las columnas no es significativo al aplicarles un alias o nombre del atributo.
- Todos los valores son indivisibles, esto es, no se pueden descomponer sin pérdida de información.
- No debe haber renglones repetidos.
- Debe haber uno o más atributos que identifiquen unívocamente a cada renglón (llave primaria).

Tipo de llaves.

- *Llave Candidata.* Es un atributo o grupo de atributos que puede ser elegido como llave primaria.
- *Llave Primaria.* Es un atributo o grupo de ellos que identifica unívocamente un renglón y ha sido elegido como identificador único.
- *Llave Alterna.* Es una llave candidata que no ha sido elegida como una llave primaria.
- *Llave Foránea.* Es un atributo o grupo de atributos que se encuentra en la tabla A y es llave primaria en la tabla B. Hace referencia a una tupla en otra tabla. Generalmente forma parte de una PK compuesta.

Tipos de Relaciones.

*Relación uno a uno.*

Se presenta cuando existe una relación como su nombre lo indica uno a uno, denominado también relación de matrimonio. Una entidad del tipo A solo se puede relacionar con una entidad del tipo B, y viceversa; por ejemplo: la relación asignación de automóvil que contiene a las entidades EMPLEADO, AUTO, es una relación 1 a 1, ya que asocia a un empleado con un único automóvil por lo tanto ningún empleado posee más de un automóvil asignado, y ningún vehículo se asigna a más de un trabajador.

*Relación uno a muchos.*

Significa que una entidad del tipo A puede relacionarse con cualquier cantidad de entidades del tipo B, y una entidad del tipo B solo puede estar relacionada con una entidad del tipo A.

*Relación muchos a uno.*

Indica que una entidad del tipo B puede relacionarse con cualquier cantidad de entidades del tipo A, mientras que cada entidad del tipo A solo puede relacionarse con solo una entidad del tipo B.

*Relación muchos a muchos.*

Establece que cualquier cantidad de entidades del tipo A pueden estar relacionados con cualquier cantidad de entidades del tipo B.

Existen dos formas de representar las relaciones entre las entidades.

1. Haciendo una tabla que contenga cada una de las llaves primarias (se definirá más adelante) de las entidades involucradas en la relación.
2. Incluyendo en alguna de las tablas de las entidades involucradas, la llave de la otra tabla.

*Modelo de red.*

Este modelo representa los datos mediante colecciones de registros y sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

*Modelo jerárquico.*

Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

*MODELOS FÍSICOS DE DATOS.*

Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son:

1. Modelo unificador.
2. Memoria de elementos.

**2.4.3 Sistemas Manejadores de Bases de Datos (DBMS).**

*Definición.*

Es un programa o grupo de programas que basados en un modelo de datos que permiten la definición, manipulación y control de una base de datos.

Entre las ventajas que generan se encuentran:

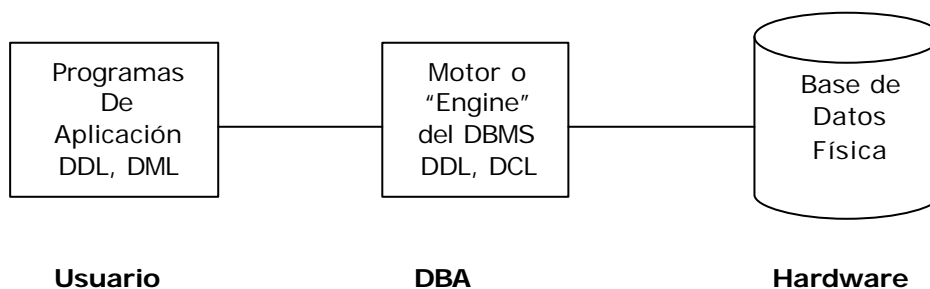
1. Facilitan la integridad, seguridad y concurrencia de los datos.
2. Los datos se almacenan con mínima de redundancia.
3. Las aplicaciones son independientes del almacenamiento físico de los datos.



*Componentes.*

- DDL o Lenguaje de Definición de Datos. Con el que se define la estructura (tablas, índices, vistas, etc.) de la base de datos mediante comandos como crear, eliminar o alterar.
- DML o Lenguaje de Manipulación de Datos. Con el se realiza la consulta y edición de la información contenida en la base de datos, esto implica: insertar, borrar, modificar.
- DCL o Lenguaje de Control de Datos. Permite la definición de los privilegios de control de acceso y edición a los elementos que componen la base de datos (seguridad), es decir, permitir o revocar el acceso.
- DD o Diccionario de Datos. Va a tener la información de todos los objetos de la base de datos. Sus principales funciones son las siguientes:
  1. Describe todos los elementos en el sistema.
  2. Los elementos se centran en los datos.
  3. Comunica los mismos significados para todos los elementos del sistema.
  4. Documenta las características del sistema.
  5. Facilita el análisis de los detalles para evaluar las características y determinar cómo deben realizarse los cambios.
  6. Localiza errores y omisiones en el sistema.

*Arquitectura.*



#### 2.4.4 Formas Normales.

Son las técnicas para prevenir las anomalías en las tablas. Dependiendo de su estructura, una tabla puede estar en primera forma normal, segunda forma normal o en cualquier otra.

##### *Primera forma normal (1FN).*

Definición formal: una relación R se encuentra en 1FN si y solo si por cada renglón columna contiene valores atómicos, es decir un solo valor por cada uno de los campos que conforman a los atributos de las entidades.

Abreviada como 1FN, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

- Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
- Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
- Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.
- Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal.

##### *Segunda forma normal.*

Para definir formalmente la segunda forma normal requerimos saber que es una dependencia funcional: consiste en edificar qué atributos dependen de otro(s) atributo(s).

Definición formal: una relación R está en 2FN si y solo si está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria.

Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, esta en segunda forma normal. Se representa por dependencias funcionales.

## 2.5 Sistemas operativos

### *Definición de sistema operativo*

El objetivo fundamental de los sistemas operativos es gestionar y administrar eficientemente los recursos hardware, permitiendo que se ejecuten concurrentemente varios programas, sin que haya conflictos en el acceso de cada uno de ellos a cada uno de los recursos que necesite, y sin que ningún programa monopolice alguno de ellos.

Un sistema operativo es un programa (o conjunto de programas) de control que tiene por objeto facilitar el uso de la computadora y conseguir que ésta se utilice eficientemente.

Es un programa de control, ya que se encarga de gestionar y asignar los recursos hardware que requieren los programas. Tomando en cuenta una computadora que es utilizada desde diversas terminales por varios usuarios, los recursos de hardware son: el procesador (CPU), la memoria principal, los discos, y otros periféricos. Obviamente, si varios usuarios están utilizando la misma computadora, debe haber alguien o algo que asigne los recursos y evite los conflictos que puedan surgir cuando dos programas requieran los mismos elementos (la misma unidad de disco, o la impresora, por ejemplo). Esta es una de las funciones del sistema operativo. Además de esta función de asignar los recursos a cada programa, el sistema operativo se encarga de contabilizarlos y de la seguridad (que un usuario no pueda acceder sin autorización a la información de otro, por ejemplo).

El sistema operativo facilita el uso de la computadora. Se presenta a continuación un ejemplo de cómo lo hace. Cada dispositivo de E/S, para ser utilizado, requiere varias instrucciones máquina que establezcan un dialogo entre las unidades centrales y el periférico, enviando o captando el dato de salida o de entrada, respectivamente. Estas instrucciones dependen considerablemente de las

características concretas del periférico. Si se trata, por ejemplo, de una unidad de disco, hay que considerar el ancho de los buses, el tamaño de su memoria intermedia, el arranque (y parada) de los motores de la unidad, el código identificador de la posición a donde hay que acceder, etc. Por otra parte, un dispositivo dado, estas instrucciones u operaciones son comunes para grabar o leer cual tipo de información (programas o datos), sea cual sea su naturaleza. El sistema operativo, con objetivo de facilitar el trabajo de los programadores, contiene módulos de gestión de entradas/salidas que evitan a los usuarios tener que incluir esas instrucciones cada vez que hacen una operación de entrada o salida. Se puede decir que esos programas del sistema operativo hacen transparente al usuario las características hardware concretas de los dispositivos.

El sistema operativo también hace que la computadora se utilice eficientemente. Para poner de manifiesto cómo el sistema operativo puede incrementar la eficiencia se considerará un ejemplo sencillo. Los programas tradicionalmente se ejecutan secuencialmente; es decir, hasta que no concluye la ejecución de una instrucción no se ejecuta la siguiente. Por ejemplo, dentro de un programa hay una instrucción que implica la escritura de una página en una impresora láser. Hasta que no acabe de imprimirse dicha página el programa no puede continuar ejecutándose.

En resumen, el sistema operativo efectúa, entre otras, las siguientes funciones:

- Facilita el uso de la computadora, en general, la comunicación computadora/usuario.
- Gestiona y asigna recursos hardware (procesador, memoria y periféricos) a los distintos programas o tareas.
- Gestiona y mantiene los archivos en dispositivos de memoria masiva.
- Apoya a otros programas.
- Protege los datos y los programas, cuestión especialmente compleja en sistemas multiusuario.
- Identifica y autentifica a los usuarios que hacen uso de la computadora. Contabiliza la utilización de los recursos realizada por los distintos usuarios.

### 2.5.1 Conceptos Básicos de Sistemas Operativos.

Llamadas al sistema. Son una interfaz entre el sistema operativo y los programas del usuario, que se definen como un conjunto de "instrucciones ampliadas". Su función es crear, eliminar y usar objetos del software, como son los procesos y los archivos.

Proceso. Es un programa en ejecución. Consta de programa ejecutable, registros, pila y contador de programa.

Intérprete de comandos o shell. Lee los comandos desde una terminal y los traduce en instrucciones para el hardware de la computadora.

Archivo. Conjunto de información importante para el usuario.

Memoria. Conjunto de palabras o bytes donde cada uno tiene asignado una dirección.

### 2.5.2 Componentes del Sistema Operativo.

*Administración de Procesos.*

- Crear y eliminar procesos de usuarios y de sistema.
- Suspender y reanudar la ejecución de procesos.
- Proporcionar mecanismos para la sincronización de procesos.
- Proporcionar mecanismos para la comunicación entre procesos.
- Mecanismos para evitar bloqueo mutuo (cuando se quiere ejecutar dos procesos al mismo tiempo, alguno de ellos o los dos se bloquean).

*Administración de Memoria (principal).*

- Llevar un control de las zonas que se está usando y quién las usa.
- Decidir qué proceso entra a memoria cuando haya espacio disponible.
- Asignación y recuperación del espacio en memoria cuando se requiera.

*Administración de Memoria (secundaria).*

- Administración del espacio libre.
- Asignación del almacenamiento.
- Planificación de las operaciones del disco.

*Administración del sistema E/S.*

Consiste de un sistema de memoria caché formado por buffers; una interfaz general con los manejadores de los dispositivos; manejadores para dispositivos de hardware específico.

*Administración de Archivos.*

- Crear y eliminar archivos, directorios.
- Manejo de operaciones primitivas (lenguaje de bajo nivel) para el manejo de archivos y directorios.
- Correspondencia entre archivos de almacenamiento secundario.
- Copia de seguridad de archivo en medios de seguridad estable (no volátiles: cintas, discos).

*Sistema de Protección.*

Protección de archivos contra posibles pérdidas o acceso de otros usuarios.

*Redes.*

*Sistema de Interpretes de Comandos.*

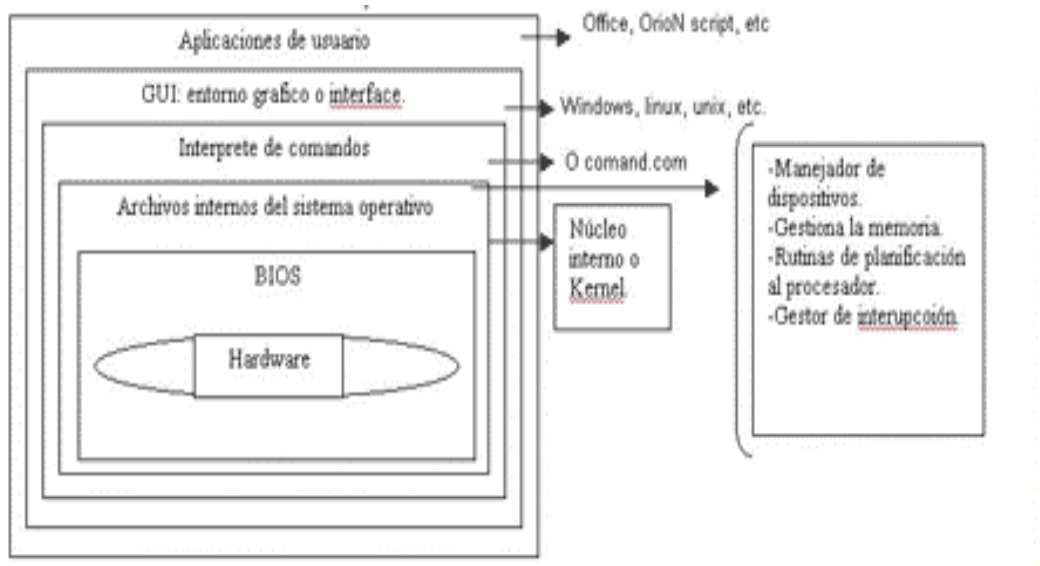
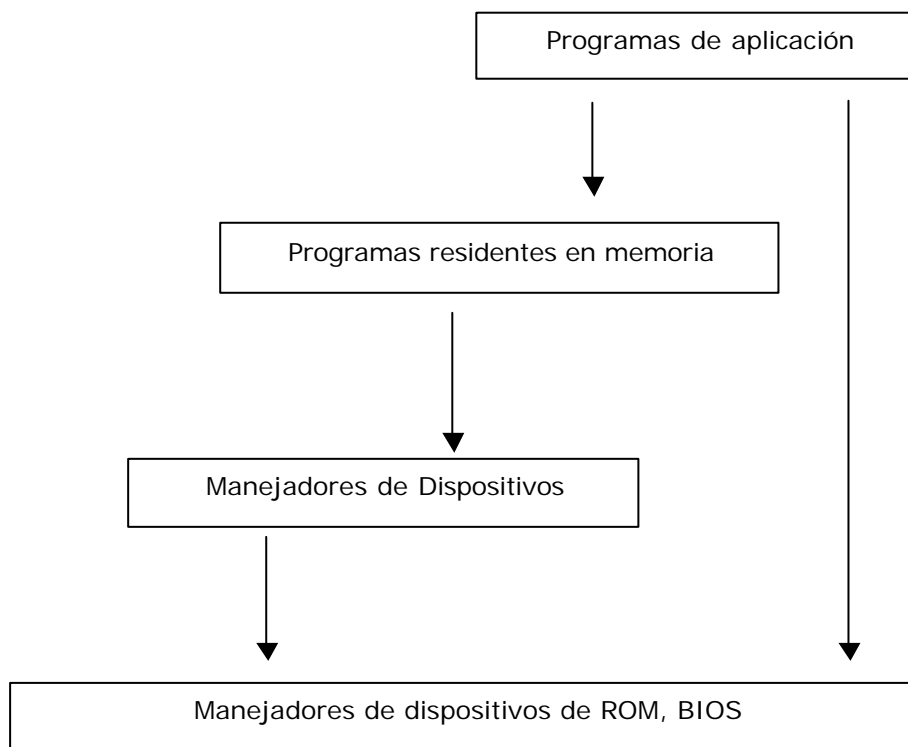


Grafico interno de la fusión entre el software y el hardware.

### 2.5.3 Evolución de la estructura de los Sistemas Operativos.

Sistemas Operativos Monolíticos.



*Sistemas Operativos por capas.*

5	El operador
4	Programas de usuario
3	Control E/S
2	Comunicación operador-proceso
1	Administración de la memoria y disco
0	Asignación del procesador y multiprogramación

*Sistemas por bloques.*

Características:

- Remitir programas, datos, etc., en forma de un bloque de trabajo.
- Se tiene poca o ninguna interacción con el usuario.
- Se usa para procesos largos y complejos.
- Procesamiento en batch (lotes).
- Administración de procesos sencilla.
- Administración E/S sencilla.
- Administración de archivos sencilla.

*Sistemas multiprogramados.*

Características:

- Ejecutan procesos en forma concurrente.
- Administración de procesos compleja. Requiere de algoritmos de planificación de procesos y de sincronización y comunicaciones de procesos.
- Administración de memoria compleja y requiere de algoritmos.

*Sistemas Multiusuario o Compartido.*

Características:

- Multiprogramado, multiusuario, reparte de forma equitativa los recursos, protección a los recursos de los usuarios.
- Se utilizan en entornos donde deban de ser aceptados un número de sucesos, la mayoría externos al sistema informático, en breve tiempo o en plazos cortos.

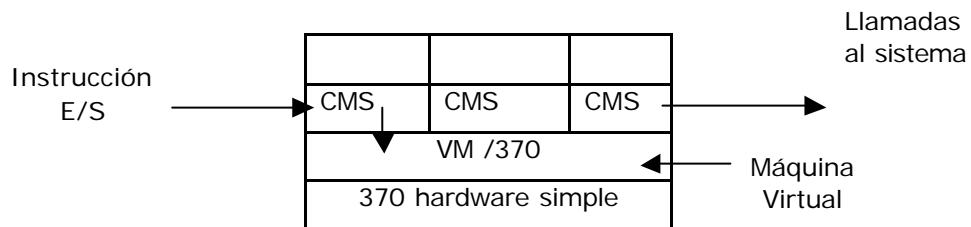


Objetivos: proporcionar rápidos tiempos de respuesta; la utilización de recursos es secundaria, la conveniencia del usuario es secundaria; procesa miles de ráfagas de interrupciones por segundo sin perder un solo suceso, políticas y técnicas para ejecutar su trabajo, multiprogramados.

*Sistemas Distribuidos.*

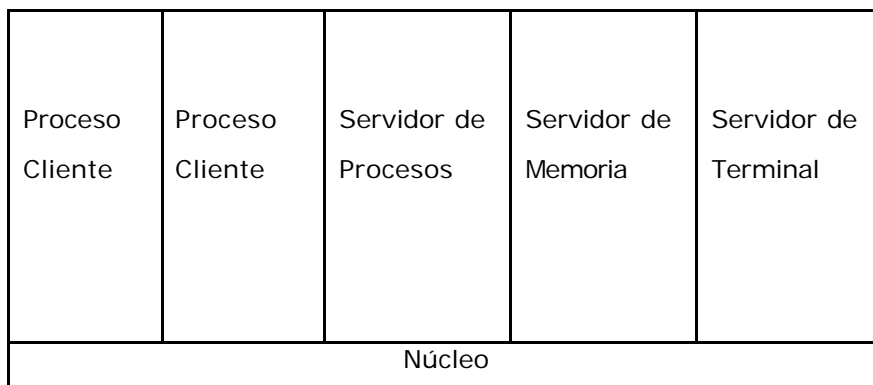
Es una colección de sistemas informáticos autónomos capaces de comunicarse y cooperar mediante interconexiones de hardware-software.

*Máquina Virtual.*

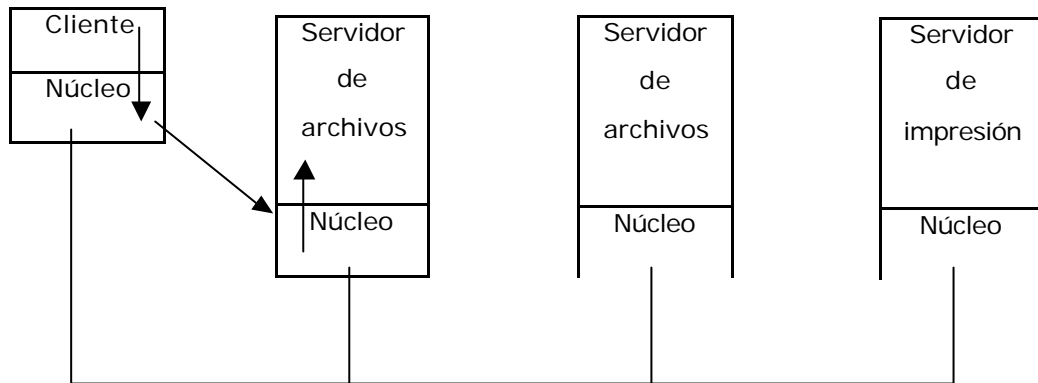


Se trata de una copia idéntica del hardware (disco, procesador, interrupciones).

*Modelo Cliente-Servidor (en una máquina).*



Modelo Cliente/Servidor Distribuido.



#### 2.5.4 Historia de los Sistemas Operativos: MS-DOS, Unix, Macintosh, Windows, Linux.

##### MS-DOS

Tim Paterson, principal artífice del que sería uno de los sistemas operativos más conocidos del mercado. En Seattle Computer, Tim Paterson trabajó en varios proyectos. No obstante, las cosas empezaron a cambiar cuando acudió a un seminario sobre el chip 8086 de Intel en julio de 1978.

De hecho, el primer diseño de la placa 8086 se terminó a finales de enero pero él ya tenía el prototipo en mayo de 1979. Una vez el prototipo 8085 estuvo funcionando, Seattle se acercó a Digital Research para comprobar si podía lograr que CP/M trabajara con él. Junto a esto, Microsoft, que se había trasladado a Seattle en enero de 1979, quería observar si algunos de sus programas funcionaban. Y, a finales de mayo de 1979, Paterson se incorporó a Microsoft.

En la Conferencia National Computer celebrada en Nueva York en el año 1979, Seattle Computer fue invitado por Microsoft y Lifeboat. Estas mostraron Basic-86, entonces el único software diseñado para el 8086. En abril de 1980, Paterson empezó a trabajar por primera vez en un sistema operativo. Y en la primavera de 1980, Paterson empezó a trabajar en lo que se convertiría en MS-DOS. En julio, ya se había

terminado cerca del 50 por ciento del sistema conocido como QDOS 0.10 aunque rápidamente se presentó QDOS 0.11. Una de las diferencias más significativas entre CP/M-86 y MS-DOS fue el sistema de gestión de archivos. CP/M generalmente ofrece una ventana de no más de 16 ó 32K; sin embargo, con MS-DOS, hay más capacidad.

A finales de agosto de 1980, QDOS 0.11 trabajaba y se comercializaba bastante bien. Paterson trabajó duro para incluir todas las características propias de un sistema operativo completo. En diciembre de 1980, Paterson y compañía presentaron 86-DOS, 0.33, que representaba mejoras significativas sobre QDOS. El ensamblador, originalmente escrito en el Z-80, se hizo más rápido. También se hicieron algunos cambios en las denominaciones del sistema. Así, a finales de 1980, Seattle Computer vendió 86-DOS a OEMs (integradores) y a otras compañías como Microsoft. En abril de 1981, mientras Paterson se centraba en algunos cambios internos del 86-DOS, decidió abandonar Seattle Computer. En mayo, se fue a trabajar a Microsoft a tiempo completo en la versión PC-DOS de 86-DOS.

A pesar de todo, Paterson terminó PC-DOS en julio, un mes antes de que el ordenador fuera anunciado oficialmente a nivel mundial. En ese momento, 86-DOS se había convertido en MS-DOS.

## **Unix**

Desarrollado en los Laboratorios Bell de AT&T a finales de los 60, este sistema operativo es el más popular dentro de los sistemas de gama alta. Unix es un sistema operativo multiusuario y multitarea ampliamente utilizado para controlar programas en estaciones de trabajo y especialmente en servidores. Un sinnúmero de aplicaciones comerciales se ejecuta en servidores Unix, y muchos sitios web funcionan bajo Unix. Hay diferentes versiones del sistema operativo y, excepto en el mundo de los ordenadores personales, donde el que domina es Windows, casi todos los vendedores de hardware lo ofrecen como sistema operativo primario o secundario. Sun ha sido uno de los instrumentos más importantes en la comercialización de Unix con su sistema operativo Solaris, antiguamente llamado SunOS. HP, IBM y SCO también han sido grandes vendedores y promotores de Unix.

Unix está escrito en C y ambos fueron desarrollados por AT&T y distribuidos libremente a instituciones gubernamentales e instituciones académicas, lo que supuso

que fuera incluido en una mayor variedad de familias de máquinas que cualquier otro sistema operativo. Con estos inicios, Unix llegó a ser un sinónimo de sistema abierto.

Los orígenes del sistema Unix se remontan al desarrollo de un proyecto iniciado en 1968. Este proyecto fue realizado por General Electric, AT&T Bell y el MIT (Instituto Tecnológico de Massachusetts), llevaron a cabo el desarrollo de un sistema operativo con nuevos conceptos como la multitarea, la gestión de archivos o la interacción con el usuario. El resultado de estas investigaciones se bautizó como Multics.

El proyecto resultó ser demasiado ambicioso, por lo que no llegó a buen fin y terminó abandonándose. Posteriormente la idea de este proyecto se vuelve a retomar y conduce al gran desarrollo en 1969 del sistema operativo Unix. Entre los investigadores destacaban Ken Thompson y Dennis Ritchie. En principio, este sistema operativo recibió el nombre de Unics, aunque un año después pasa a llamarse Unix, como se conoce hoy en día.

Inicialmente Unix fue considerado como un proyecto de investigación, hasta el punto de distribuirse de forma gratuita en algunas universidades, pero después la demanda del producto hizo que los laboratorios Bell iniciaran su distribución oficial. Para 1974 Unix se había convertido en un eficaz sistema operativo que se ejecutaba principalmente en máquinas PDP, llegando a ser muy popular en entornos académicos y científicos.

Incluso a pesar de sus muchas versiones, Unix es ampliamente utilizada en aplicaciones de misión crítica para servidores y sistemas de procesamiento de transacciones.

### **Macintosh**

El sistema operativo Macintosh está directamente relacionado con la familia de ordenadores de Apple, la primera del mercado que no era compatible con IBM. Fue la primera computadora que popularizó la interfaz gráfica de usuario (GUI) que, junto con su hardware, se hizo famoso en el mundo de la informática por su facilidad de uso.

Otra de las características de Macintosh es que fue en sus tiempos, y sigue siendo, la serie de ordenadores personales no compatibles con IBM. Es decir los Macintosh, o Mac, sólo ejecutan aplicaciones Mac, aunque pueden acceder a Windows

a través de emuladores como SoftWindows, que les permiten ejecutar aplicaciones de Windows y de Dos.

La primera Macintosh, lanzada al mercado en 1984, contaba únicamente con una disqueteera, una memoria de 128K y una pantalla monocroma que formaban un único bloque. Mantenido durante cierto número de años, los sucesivos diseños se centraron en suavizar las curvas en sus modelos Classic y separar los monitores de las CPUs. A principios de los años 90, Apple vuelve a sus raíces y lanza la línea iMac y posteriormente el G4 Cube, regresando a los diseños compactos.

Los primeros modelos contaban con una familia de CPUs de 32bits 680x0 de Motorola. En 1994 Apple presentó en el mercado los PowerMacs, que utilizaban chips PowerPC de alto rendimiento diseñados por Apple, Motorola e IBM. De esta forma, los PowerMac ejecutaban aplicaciones nativas PowerPC y emulaban las tradicionales aplicaciones de los Mac 680x0.

Lanzado al mercado en 1999, el chip G4 permitía que Macintosh ejecutara 1.000 millones de operaciones en coma flotante por segundo, lo que suponía una máquina extremadamente rápida para las aplicaciones basadas en gráficos e imágenes tales como Photoshop.

Uno de los principales logros de Macintosh fue que popularizó las interfaces gráficas de usuario (GUI) y simuló escritorios en las pantallas de los ordenadores. Fue el primer sistema que permitía ejecutar archivos arrastrándolos desde una carpeta a una aplicación directamente, lo que atrajo, desde el momento de su lanzamiento, a las personas no muy habituadas a los ordenadores. Y sigue siendo una de las razones, su facilidad de uso, por lo que en Estados Unidos los Macintosh son ampliamente utilizados en colegios.

La interfaz gráfica de usuario fue realmente desarrollada por Xerox, que lo introdujo en su estación de trabajo Star en 1981. Apple se apropió de Star y, consecuentemente, otros copiaron a Mac, trasladando la interfaz gráfica de usuario al sistema operativo Windows y Unix.

## **Windows**

El 10 de noviembre de 1983 Microsoft anunció Windows, una extensión del sistema operativo MS-DOS. Con Windows 1.0 se inició la era de la interfaz gráfica de usuario (GUI) en las PCs. La primera versión de Windows supuso un hito en la industria porque permitió a los usuarios de PCs cambiar el método propio del MS-DOS de teclear los comandos a utilizar un ratón para apuntar y hacer click para ejecutar las funciones, como iniciar las aplicaciones.

Windows 1.0 también permitía a los usuarios moverse entre varios programas sin tener que salir de ella y reanudarlas a cada cambio. Asimismo, el producto incluía un conjunto de aplicaciones como un calendario, un bloc de notas, calculadora, reloj, etc.

En 1987 Microsoft lanzó al mercado Windows 2.0, que tomaba ventaja del aumento de la velocidad de procesamiento del procesador Intel 286, del aumento de la memoria y de la capacidad de comunicación entre las aplicaciones utilizando Dynamic Data Exchange (DDE). Además del soporte al estándar gráfico VGA y los atajos de teclado, es decir, utilizar una combinación de teclas para moverse rápidamente a través de las operaciones de Windows.

En 1990 le tocó el turno a Windows 3.0, que ofrecía un rendimiento de 32-bits, gráficos avanzados, y total soporte para los procesadores 386 de Intel más potentes. Ofrecía nuevas características y capacidades, como administrador de programas, de archivos y de impresoras, además de un entorno de desarrollo de aplicaciones.

1993 fue el año de lanzamiento de Windows for Workgroups 3.11, que añadía soporte para trabajo en grupo peer-to-peer y conexión de redes. Por primera vez, las PCs basados en Windows estaban conectados en red y se convirtieron en parte integrante de la evolución informática del momento: cliente/servidor.

El lanzamiento de Microsoft Windows NT el 27 de julio de 2003. Windows NT fue el primer sistema operativo Windows en combinar el soporte para aplicaciones de negocios cliente/servidor con las aplicaciones de productividad personal que lideraban la industria. El sistema operativo rompía esquemas en seguridad, potencia, rendimiento, escalabilidad y fiabilidad con una serie de nuevas características clave.

Windows 95 fue el sucesor de los tres sistemas operativos de Microsoft existentes para propósitos generales: Windows 3.1, Windows for Workgroups y MS-DOS. Esta versión incluía el soporte de 32-bit, TCP/IP para la conexión a Internet y capacidades plug and play para facilitar a los usuarios la instalación de hardware y software. También ofrecía capacidades multimedia mejoradas, características más potentes para la informática móvil y conexión de redes integrada.

Windows Workstation 4.0 llegó en 1996 y fue una actualización del sistema operativo para computadoras de sobremesa para corporaciones que incrementó la facilidad de uso y una gestión simplificada, además de una mayor capacidad de conexión de redes, y un completo conjunto de herramientas para el despliegue y gestión de Intranets.

Windows 98 fue la actualización de Windows 95 y fue la primera versión de Windows diseñada específicamente para los consumidores. Daba a los usuarios la oportunidad de abrir y cerrar las aplicaciones más rápidamente, además de incluir soporte para leer discos DVD y conectar dispositivos a través de puertos USB. En 1999 Microsoft lanza al mercado Windows 98 SE, o Second Edition, que ofrecía a los usuarios una variedad de nuevas y mejores compatibilidades y otras características relacionadas con Internet.

Windows ME o Millenium Edition, lanzado en 2000, ofreció a los consumidores numerosas mejoras con los archivos de música, vídeo y conexiones de redes. Windows Movie Maker permitía editar, grabar y compartir vídeos; Windows Media Player 7 facilitaba la búsqueda, organización y reproducción de archivos digitales. Por otra parte, Windows ME fue el último sistema operativo de Microsoft basado en el kernel de Windows 95. Tras el lanzamiento de esta versión, Microsoft anunció que todos sus futuros sistemas operativos estarían basados en los kernel de Windows NT y Windows 2000.

Windows 2000 Professional fue la actualización de Windows NT Workstation 4.0. Windows 2000 Professional fue diseñado para reemplazar a Windows 95, Windows 98 y Windows NT Workstation 4 en todos los desktops de negocio y computadoras portátiles. Construido sobre el probado código base de Windows NT Workstation 4.0, Windows 2000 añadió mejoras en la fiabilidad, facilidad de uso, compatibilidad con Internet e informática móvil. También hacía que la instalación de hardware fuera

mucho más fácil que antes añadiendo soporte para una amplia variedad de nuevos plug and play para hardware, incluidos productos inalámbricos y conexión de redes avanzada, dispositivos USB y dispositivos infrarrojos.

Finalmente, Windows XP, lanzado en 2001, supuso un salto hacia los sistemas operativos para computadoras de sobremesa. Con el lanzamiento del Windows XP Home Edition y Windows XP Professional en octubre de 2001, Microsoft fusionó sus dos líneas de sistemas operativos para consumidores y de negocio, uniéndolos en torno al código base de Windows NT y Windows 2000.

Con Windows XP, los consumidores y usuarios del hogar tenían ahora el rendimiento, estabilidad y seguridad de las que los usuarios de negocio se habían beneficiado en Windows 2000.

### **Windows para servidores**

Los sistemas operativos Windows para servidores tienen una historia más corta que la de los PCs, aunque comparten la misma herencia.

En 1988 Microsoft concibió el objetivo de desarrollar un sistema operativo de 32-bits multiusuario a partir de Microsoft Windows NT. En 1991 Microsoft mostró Windows Advanced Server para gestión de redes LAN, un sistema operativo de gama alta que posteriormente sería renombrado como Microsoft NT. El producto ofrecía la familiar interfaz de usuario y el modelo de programación de Windows, y era capaz de ejecutar todas las aplicaciones desarrolladas para Windows 3.0.

En julio de 1993, se lanzó el primer sistema operativo para servidores. Windows NT Advanced Server 3.1 fue diseñado para actuar como un entorno de cliente/servidor en un servidor dedicado, además de ofrecer potencia, escalabilidad, mejora a la tolerancia a fallos e interoperabilidad basada en estándares. Además, también era una potente plataforma para los servidores de bases de datos como Microsoft SQL Server, servidores de comunicaciones como Microsoft SNA Server y servidores de correo electrónico como Microsoft Mail.

El siguiente lanzamiento de Windows NT Server, la versión 3.5 lanzada en 1994, fue construido sobre la versión estable 3.1, pero con grandes mejoras respecto a la velocidad de procesamiento y una mejor conectividad con otros sistemas. En 1995 le



llegó el turno a la versión 3.51, que incluía una herramienta para ayudar a gestionar las Licencias de Acceso de Cliente para la familia BackOffice de Microsoft.

*Mejoras palpables.*

Windows NT Server 4.0, lanzada al mercado en 1996, se convirtió en un sistema operativo de 32-bits, además de añadir características avanzadas para usuarios técnicos, como servicios de impresión y archivo más rápidos, una mayor capacidad de conexión de redes, comunicación basada en estándares, un completo conjunto de herramientas para el despliegue y gestión de intranets. Además, también se ofrecían otras características como soporte a smart cards o capacidad de clustering.

En octubre de 1998 Microsoft anunció que eliminaría el sufijo NT en la próxima gran versión del sistema operativo Windows NT, que llevaría el nombre de Windows 2000. El mismo año 1998 Microsoft presentó Windows NT Server 4.0, Terminal Server Edition, que ofreció al sistema operativo Windows NT Server la capacidad de servir aplicaciones basadas en un sistema operativo Windows de 32-bit a terminales. El entorno de Terminal Server era, por definición, una arquitectura de cliente donde todos los procesos de las aplicaciones se realizaban en el servidor.

### **Linux.**

El padre de Linux es Linus Torvalds, un programador finlandés de 21 años que inicialmente no tenía más pretensión que divertirse creando un sistema operativo para su uso personal.

En los años 80 el rey supremo de los sistema operativos era DOS y Unix que era lo suficientemente caro como para mantener alejados a los pequeños usuarios de PCs y se dirigía fundamentalmente a entornos corporativos.

A finales de los años 90 apareció lo que podría ser la solución: Minix, una versión de Unix limitada escrita por el profesor holandés Andrew S. Tanenbaum, que quería enseñar a sus estudiantes el funcionamiento de un sistema operativo. Fue diseñado para funcionar con los microprocesadores 8086, que por aquellos tiempos inundaron el mercado mundial. Como sistema operativo, Minix no era magnífico, pero la ventaja es que el código fuente estaba disponible. Cualquiera podía acceder a las

12.000 líneas de código, escritas en el lenguaje ensamblador C. Y uno de esos estudiantes fue Linus Torvalds.

En 1991, Linus Benedict Torvalds era estudiante de segundo año de Ciencia Informática de la Universidad de Helsinki. Su intención inicial no era crear un sistema operativo, sino experimentar creando software para manejar ciertas funciones básicas del PC. Por esa misma fecha, programadores de todo el mundo comenzaron a sentirse muy atraídos por el proyecto GNU, dirigido por Richard Stallman, que proporcionaba software libre y gratuito. La idea de Stallman era que a diferencia de otros productos, el software debería estar libre de restricciones para poder hacer programas informáticos mejores y más eficaces. Con su famoso manifiesto de 1983 que declaraba los comienzos de proyecto GNU, inició un movimiento para crear y distribuir software que encarnara su filosofía. De esta forma, a principios de 1984, Stallman comenzó a escribir el GNU C Compiler (GCC) y en poco tiempo él solo superó a grupos enteros de programadores de las compañías de software comercial y GCC fue considerado como uno de los compiladores más eficaces y robustos jamás creado.

En 1991, el proyecto GNU creó un montón de herramientas. El esperado compilador Gnu C ya estaba disponible, aunque todavía no era un sistema operativo.

El 25 de agosto de 1991 Linus Benedict Torvalds envió por correo electrónico un mensaje que resultó ser histórico, en el que saludaba a los usuarios de Minix y les explicaba que estaba haciendo, como hobby, un sistema operativo no profesional para procesadores 386(486), y pedía a la gente que le informara sobre qué características les gustaría que tuviera, recibiendo de buena gana cualquier sugerencia.

La versión de Linux 0.01 fue lanzada a mediados de septiembre de 1991 y fue colocada en la red. Los entusiastas descargaron los códigos, que probaron, retocaron y se devolvieron a Linus, que el 5 de octubre lanzó la versión 0.02. Pocas semanas después aparece la versión 0.03 y en diciembre la 0.10 ya estaba preparada.

Sólo ofrecía soporte para discos duros AT. La versión 0.11 fue mucho mejor, con soporte para teclados multilingües, disqueteras, y soporte para VGA, EGA, Hercules, etc. En pocos meses, el código daba la vuelta al mundo.

La mejor ventaja de Linux por aquellos tiempo, y también ahora, es que cada vez que sale una nueva pieza de hardware, el kernel de Linux es retocado para

adaptarse a ella. Por ejemplo, a las pocas semanas de lanzarse el Intel Xeon Microprocessor, Linux ya era capaz de ejecutarlo.

Actualmente Linux es uno de los sistemas operativos con mayor índice de crecimiento del mundo. Ha pasado de contar con unos cuantos fanáticos en 1991-1992 a tener a millones de usuarios hoy en día.

Después de IBM han sido muchas las compañías que actualmente ofrecen soporte para Linux, desde las más grandes hasta las más pequeñas y desde las creadoras de software de bases de datos hasta las compañías de seguridad que desarrollan antivirus. Pero lo que es realmente destacable y asombroso es que grupo de desarrolladores esté constantemente incrementándose a lo largo de todo el mundo.

En establecimientos educacionales se han instalado grandes redes y clusters de servidores con cientos de máquinas. Linux también está siendo utilizado en PDAs, thin clients y dispositivos más pequeños.

Linux ha vencido en el mercado de servidores por su escalabilidad y robustez. No venció debido al marketing, porque no ha habido mucho, con excepción del dibujo del pingüino y la ayuda de IBM. Ha vencido por calidad y gran proliferación.

Habiendo estudiado los paradigmas de la ingeniería de software llegamos a la conclusión que el paradigma a utilizar será el de cascada.

## **CAPÍTULO 3**

---

---

## **ANÁLISIS DEL SISTEMA**

### **3.1 Análisis de los recursos disponibles**

#### **3.1.1 Recursos informáticos disponibles**

Para poder llevar a cabo la tarea de desarrollo del nuevo sistema, se requieren ciertos recursos mínimos, tanto de hardware como de software. A continuación se hará mención de los recursos con los cuales se cuenta y que han sido puestos a nuestra disposición en el Departamento de Análisis, Desarrollo y Mantenimiento de sistemas, instancia por medio de la cual se desarrollará el sistema.

En cuanto a hardware son:

- Servidor sunblade para base de datos.
- PC Pentium IV

En cuanto a software:

- Sistema operativo Windows 98, Windows 2000 Server, Windows XP.
- Sistema operativo Unix.
- Manejador de bases de datos Oracle 8i.
- Servidor WEB Apache Tomcat

### **3.2. Análisis de la base de datos**

#### **3.2.1 Modelo Entidad Relación**

Como ya se mencionó, uno de los objetivos de esta tesis es desarrollar un sistema integral capaz de implementar y automatizar el proceso de registro, almacenamiento, recuperación y consulta de las tesis a través de Internet para cumplir con las necesidades de consulta de texto completo de la comunidad universitaria. De esta forma se pretende agilizar los trámites de titulación del tesista y solucionar el problema de espacio físico que existe en el Departamento de tesis de la Biblioteca Central.

En primer lugar, es necesario recabar toda la información posible acerca del proceso que se quiere automatizar. Por tal motivo realizamos una serie de entrevistas y cuestionarios con las personas involucradas en este proceso y que actualmente lo llevan a cabo de manera manual. Sin embargo, en esta etapa surgieron numerosos problemas debido a que, en la mayoría de las ocasiones, se trabaja con personas con una formación profesional muy distinta, por lo que las especificaciones resultan muy ambiguas y en muchos casos hasta contradictorias. Por esto fue necesario normalizar el proceso y definir qué partes y qué información de este eran realmente necesarias y cuáles no.

Basándonos en este análisis, a continuación se detalla qué información necesita ser almacenada en la base de datos.

- RFC del Alumno
- Apellido Paterno del Alumno
- Apellido Materno del Alumno
- Nombre(s) del Alumno
- Teléfono del Alumno
- Número de Cuenta del Alumno
- Carrera del Alumno
- Facultad del Alumno
- Institución del Alumno
- RFC del Asesor
- Apellido Paterno del Asesor
- Apellido Materno del Asesor
- Nombre(s) del Asesor
- Teléfono del Asesor
- Matriz de la Tesis
- Título de la Tesis
- Año de la Tesis
- Número de Páginas
- Adicional
- Ilustraciones
- Fecha de Registro de la Tesis
- Capítulos

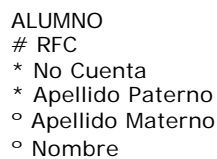
- Subtemas
- Subtítulos
- Grado

Ahora surge la necesidad de almacenar la información anterior de una manera eficiente para de esta forma poder recuperar tanto su información bibliográfica, cómo su texto completo.

Esta información debe agruparse en forma de entidades, normalizarse y relacionarse siguiendo una serie de especificaciones y restricciones que se detallan a continuación:

De primera instancia podríamos obtener las siguientes entidades:

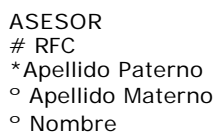
- Alumno. Se refiere a aquellas personas que son autoras de una o más tesis.



ALUMNO  
# RFC  
\* No Cuenta  
\* Apellido Paterno  
◦ Apellido Materno  
◦ Nombre

**Figura 3.1** Entidad de Alumno.

- Asesor. Aquellas personas que se encargan de asesorar a uno o más alumnos en una o más tesis.



ASESOR  
# RFC  
\* Apellido Paterno  
◦ Apellido Materno  
◦ Nombre

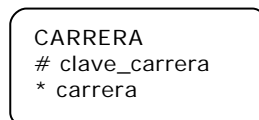
**Figura 3.2** Entidad de Asesor.

- Tesis. Aquellos trabajos realizados para la obtención de un título.



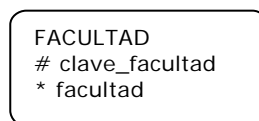
**Figura 3.3** Entidad de Tesis.

- Carrera. El detalle de la carrera, especialidad, maestría o doctorado del alumno.



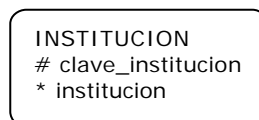
**Figura 3.4** Entidad de Carrera.

- Facultad. La escuela o facultad a la que pertenece el alumno.



**Figura 3.5** Entidad de Facultad.

- Institución. La institución a la que pertenece el alumno.



**Figura 3.6** Entidad de Institución.

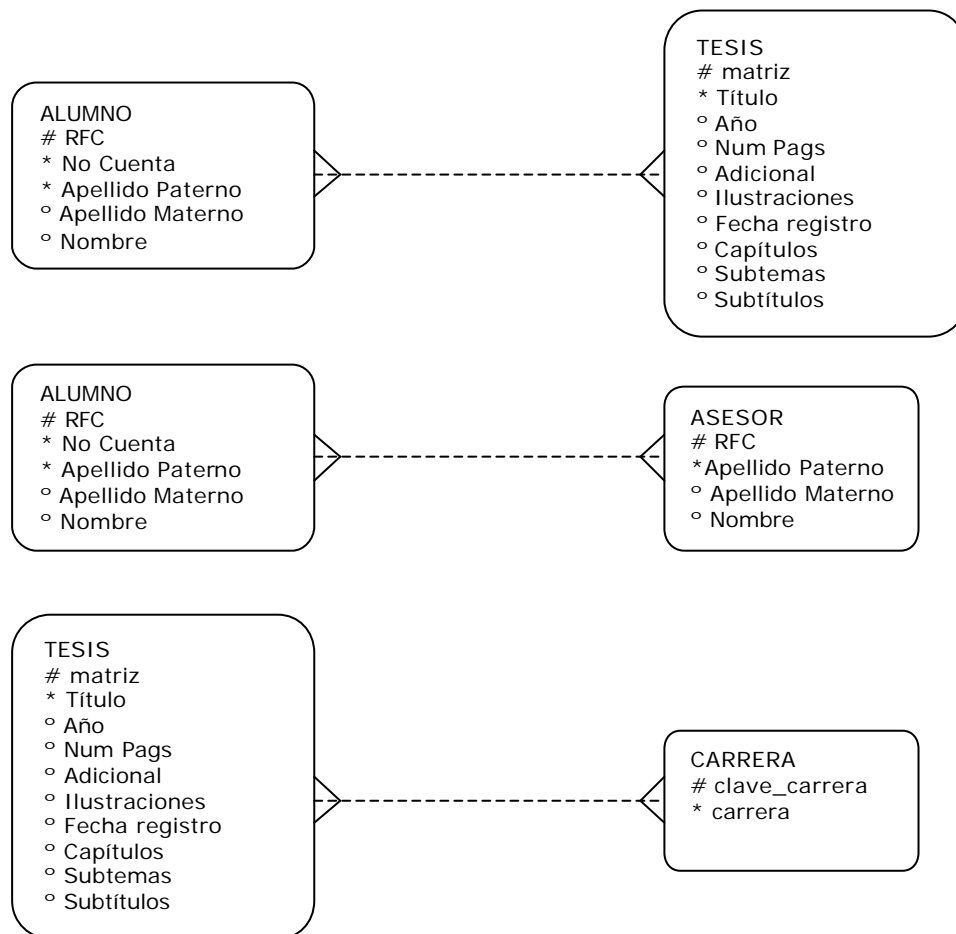
Ahora es preciso definir las relaciones entre estas entidades, basándonos en las siguientes especificaciones:

- Un alumno puede tener una o mas tesis. Esto si cursa otra carrera, posgrado o algún grado de especialización
- Una tesis puede ser realizada por uno o mas alumnos de la misma o de distintas carreras



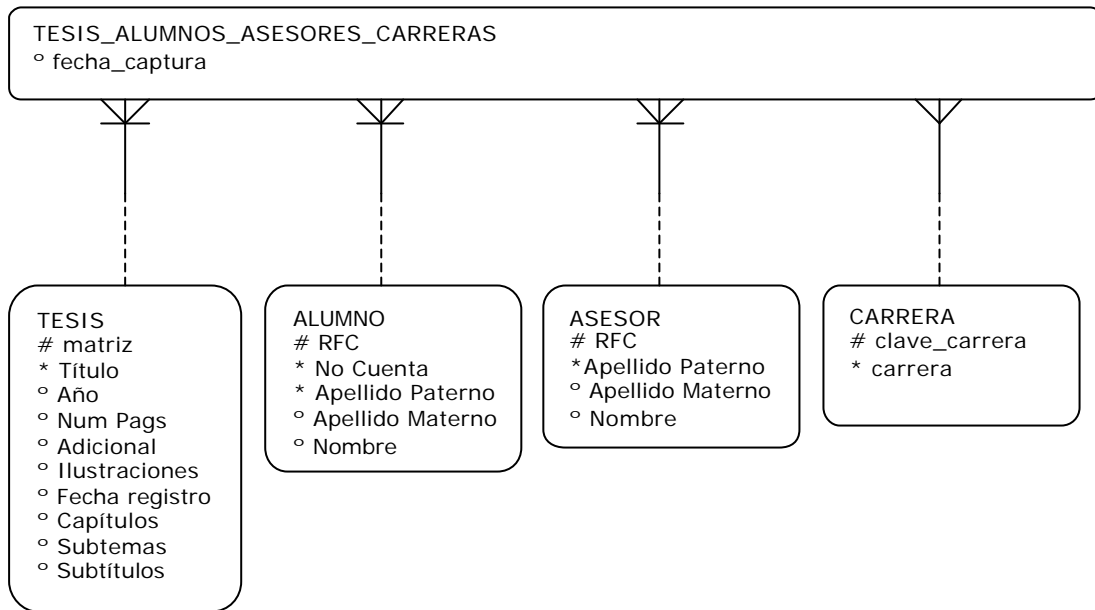
- Un alumno puede tener uno o mas asesores por cada tesis que realice.
- Un asesor puede serlo de uno o mas alumnos y a su vez de una o mas tesis.
- Un alumno puede tener una o mas carreras.
- Una carrera tiene muchos alumnos.

De lo anterior se concluye que existe una relación de muchos a muchos entre las entidades: Alumno, Asesor, Tesis y Carrera, la cual se ilustra a continuación:



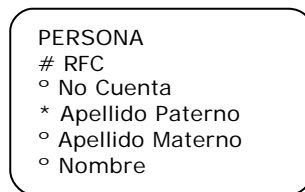
**Figura 3.7** Relaciones de muchos a muchos entre Alumno, Tesis, Asesor y Carrera.

No es posible dejar este tipo de relaciones muchos a muchos en las base de datos, ya que se generaría una redundancia en la información. Las tres relaciones anteriores las resolvemos con una entidad intersección que rompe la cardinalidad de muchos a muchos y queda de la siguiente manera:



**Figura 3.8** Resolución de la relación de muchos a muchos entre Alumno, Tesis, Asesor y Carrera.

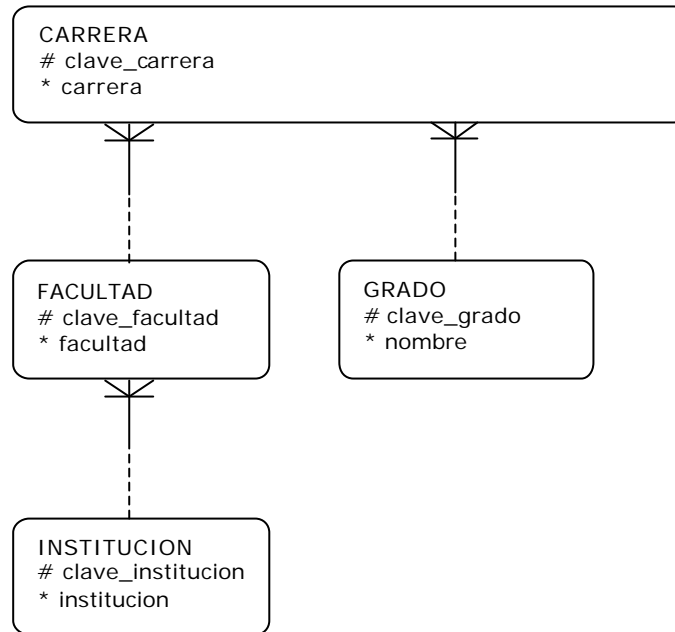
- Un alumno puede ser, a su vez, asesor de uno o mas alumnos. En este caso se trata de un problema de manejo de roles en la base de datos, en el cual una entidad juega mas de un rol. Aquí vemos que el mismo alumno es al mismo tiempo asesor, por lo que se resuelve generando una sola entidad para Alumno y Asesor denominada Persona y que asume ambos roles.



**Figura 3.9** Entidad de Persona.

- Existen instituciones propias de la UNAM e incorporadas a esta.
- Una institución puede tener una o mas facultades o escuelas.
- Una facultad puede tener una o mas carreras.
- Una carrera debe tener un grado.
- Un grado puede asignarse a muchas carreras.

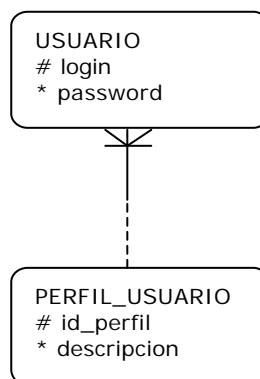
De lo anterior surgieron las siguientes relaciones:



**Figura 3.10** Relaciones entre Carrera, Facultad, Institución y Grado.

Es necesario que el sistema disponga con algún tipo de seguridad para restringir los accesos a catálogos y a ciertas utilerías del sistema.

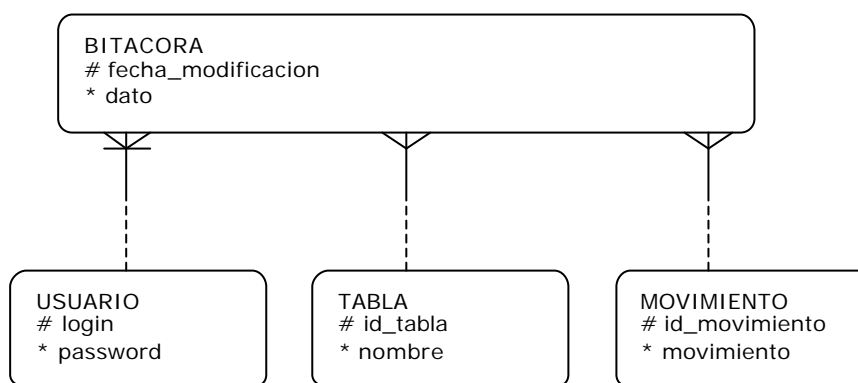
Se resolvió creando una entidad de Usuario y Perfil\_Usuario las cuales como su nombre lo indican, almacenan la información de cada usuario que ingresa al sistema, a los cuales se les asigna cierto perfil guardado en la segunda tabla. De esta forma se puede administrar qué usuario tiene qué permisos.



**Figura 3.11** Relación entre Usuario y Perfil de Usuario.

Surge la necesidad de llevar un registro de los principales movimientos o cambios realizados en la base de datos.

Para esto se crearon las entidades de Bitacora, Movimiento, Tabla y se aprovechó la de Usuario. Aquí mediante el uso de triggers o disparadores de la base de datos, se lleva una bitácora de qué usuario modificó las principales tablas de la base de datos.



**Figura 3.12** Relaciones entre las entidades que conforman la bitácora.

En estas entidades concentramos toda la información requerida, sin embargo necesitan pasar por un proceso de normalización para evitar redundancia e inconsistencia en la base de datos.

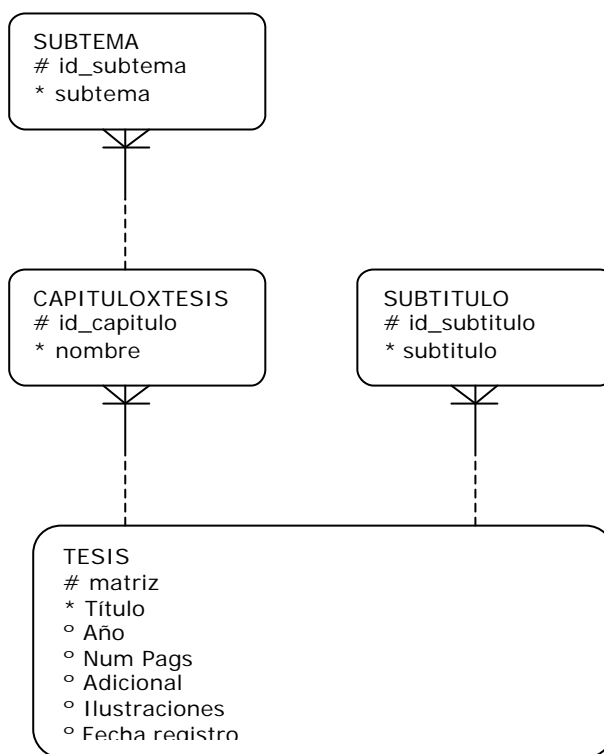
La **primera forma normal** nos dice: “*todos los atributos deben de tener un solo valor*”.

La **segunda forma normal** dice: “*Un atributo debe ser dependiente del identificador único completo*”.

La **tercera forma normal** dice “*Ningún atributo que no es identificador único puede ser dependiente de otro atributo no identificador único*”.

Examinando las entidades generadas observamos que la única que no cumple con estas tres formas normales es la entidad de tesis. Ya que los atributos subtítulos, capítulos y subtemas son multivalor. Esto se deba a que una tesis puede tener muchos capítulos y muchos subtemas asociados a estos capítulos, además de que puede tener muchos subtítulos, por tal motivo se modifica la entidad de teisis y se generan las

nuevas entidades: CapituloXTesis, Subtitulo y Subtema. Estas se relacionan de la siguiente manera:



**Figura 3.13** Relaciones entre la Tesis y sus Capítulos, Subtemas y Subtítulos.

### 3.3 Análisis del formato MARC 21

Es imposible hoy en día leer una revista bibliotecológica, asistir a una conferencia o aún sostener una conversación informal con otros bibliotecarios, sin escuchar las frases "formato MARC," registros MARC," o "compatible con MARC." Muchos bibliotecarios profesionales no han tenido oportunidad de tomar cursos formales donde se expliquen los temas más importantes de la automatización de bibliotecas y el papel desempeñado por MARC, aún cuando los sistemas automatizados serán parte importante de sus bibliotecas.

### 3.3.1 Introducción

Un registro MARC es un registro catalográfico legible por máquina (MACHINE-Readable Cataloging).

**Legible por máquina:** "Legible por máquina" significa que un tipo particular de máquina, una computadora, puede leer e interpretar los datos contenidos en un registro catalográfico.

**Registro catalográfico:** Un registro catalográfico es un registro bibliográfico, o sea, la información que tradicionalmente se presenta en una ficha de catálogo de biblioteca. Un registro puede incluir (no necesariamente en este orden): 1) una descripción del ítem, 2) el asiento principal y los asientos secundarios, 3) los encabezamientos de materia y 4) la clasificación o signatura topográfica. (Los registros MARC contienen con frecuencia mucha información adicional).

- 1) **Descripción:** Los bibliotecarios compilan la descripción bibliográfica de los materiales mediante la aplicación de las *Reglas de Catalogación Angloamericanas*, 2a. ed. (RCAA2), revisión 2002. Esta "descripción" presenta las secciones (compuestas por párrafos) de cada ficha, incluyendo: el título, la mención de responsabilidad, la mención de edición, los detalles específicos del material, la información sobre la publicación, la descripción física, la serie, las notas y los números normalizados.
- 2) **Asiento principal y asientos secundarios:** Las RCAA2 contienen también reglas para determinar cuáles serán los "puntos de acceso" a la información del registro (a los cuales llamamos habitualmente "asientos principales" y "asientos secundarios"); y para establecer la forma que éstos adoptarán. Los puntos de acceso son los puntos de recuperación de datos en el catálogo de la biblioteca que los usuarios necesitarán buscar para localizar los materiales. Dicho de otra manera, las reglas de las RCAA2 se utilizan para contestar preguntas tales como: ¿debe haber, en el caso de un libro en particular, más de un asiento de autor y más de un título?, ¿debe anotarse el título de la

serie?, ¿Cómo debe escribirse el nombre del autor?, ¿debe un ítem (sin autor) asentarse bajo título?

- 3) **Encabezamientos de materia (asientos secundarios temáticos):** El bibliotecario usa la lista de Sears (*Sears List of Subject Headings*), la Lista de Encabezamientos de la Biblioteca del Congreso (*LCSH*) u otras listas normalizadas de encabezamientos de materia, para seleccionar los encabezamientos bajo los cuales se asienta cada ítem. La utilización de una lista normalizada es importante para asegurar la consistencia y para garantizar que todos los materiales que tratan sobre un tema se asienten bajo un encabezamiento y se encuentren en un mismo lugar en el catálogo. Por ejemplo, si la lista indica que todos los libros sobre gatos deben asentarse bajo el encabezamiento GATOS; la aplicación de este encabezamiento autorizado elimina la posibilidad de que unos libros se asienten bajo GATOS y otros bajo FELINOS. Aún cuando un libro se titule Todo sobre felinos, el encabezamiento de materia será GATOS, de esa forma todos los libros sobre ese tema se encontrarán en un solo lugar para que el usuario los pueda localizar. El usuario no tendrá que imaginar todos los sinónimos posibles de la palabra que busca.
- 4) **Signatura topográfica:** El bibliotecario utiliza los esquemas de clasificación del Sistema Decimal de Dewey o de la Biblioteca del Congreso (LC) para seleccionar la signatura topográfica de un ítem. El propósito de dicha signatura es colocar juntos en los estantes los materiales sobre un mismo tema. La mayoría de los materiales se subarreglan en orden alfabético por autor. La segunda parte de la signatura topográfica, que representa generalmente el nombre del autor, sirve para facilitar dicho subarreglo.

### 3.3.2 MARC 21

La Biblioteca del Congreso de Washington sirve como repositorio oficial de las publicaciones de los Estados Unidos de América y constituye una fuente primaria de registros catalográficos de publicaciones de los Estados Unidos y de publicaciones

internacionales. Cuando la Biblioteca del Congreso comenzó a usar computadoras en la década de los sesenta, desarrolló el Formato LC MARC, como un sistema de aplicación de números, letras y símbolos en registros catalográficos que permitiera marcar diversos tipos de información. El formato original LCMARC se transformó en MARC 21 y ha llegado a ser la norma utilizada por la mayoría de los sistemas bibliotecarios automatizados. El formato bibliográfico MARC 21 (así como su documentación oficial) es preservado por la Biblioteca del Congreso; y se publica bajo el título *MARC 21 Format for Bibliographic Data*.

### 3.3.3 Origen de los registros MARC

Compartir información catalográfica (es decir registros bibliográficos) ha sido una realidad desde hace años. Muchos bibliotecarios han elegido utilizar al máximo sus limitados recursos mediante la adquisición de registros de catalogación, en vez de compilarlos ellos mismos. Desde hace casi un siglo ha sido posible comprar juegos de fichas catalográficas a la Biblioteca del Congreso, y más recientemente a través de distribuidores de libros.

Otros crearon sus propios registros utilizando los datos encontrados en la Catalogación en la Publicación (CIP) incluida en el reverso de la portada de muchos libros. Los bibliotecarios han también adaptado los registros catalográficos completos o parciales contenidos en revistas de la profesión o en bibliografías.

Una vez que hubo fácil acceso a las computadoras, los bibliotecarios ya no tuvieron que "reinventar la rueda." No había razón para que cientos de catalogadores tuvieran que compilar registros casi idénticos para los mismos materiales, cuando un solo catalogador podía crear un registro y compartirlo. ¿Qué razón había para que cientos de capturistas volvieran a escribir el mismo registro en fichas cuando una computadora podía ser programada para imprimirlas?

Después de que se desarrollaron los registros MARC, las bibliotecas pudieron compartir los beneficios de la catalogación legible por máquina, independientemente de que tuvieran o no sistemas en línea. Las fichas que se compraban a la Biblioteca del Congreso o a distribuidores comerciales fueron entonces impresas por computadoras a partir de los registros bibliográficos de los ficheros de registros MARC de la Biblioteca del Congreso.



Las universidades con grandes presupuestos para sus bibliotecas se unieron a organismos bibliográficos tales como OCLC, WLN, RLIN y A-G Canada. Mediante una cuota periódica más los gastos de comunicaciones sus departamentos de servicios técnicos iniciaron operaciones en línea con el apoyo de los poderosos sistemas de computación de los organismos bibliográficos. Estas negociaciones conjuntas permitieron que las bibliotecas utilizaran y compartieran registros MARC. Las fichas catalográficas podían requerirse por pedido y una computadora podía conectarse al sistema para recibir los datos a través de las líneas telefónicas e imprimir las etiquetas para el lomo de los libros.

Vivimos en una época emocionante en la que contamos con computadoras poderosas y poco costosas que pueden ayudar en la administración de las operaciones de las bibliotecas de todo tipo y tamaño. Los paquetes de programación para computadoras le permiten a las bibliotecas tener su propio sistema autónomo de circulación o su catálogo de acceso público en línea. Muchos de estos programas pueden leer, almacenar e imprimir registros MARC. Las computadoras actuales utilizan discos duros y blandos para almacenar información y no cintas magnéticas como los grandes sistemas de computación. Las computadoras pueden también grabar y recuperar datos de un disquete; y copiar registros al disco duro.

La tecnología automatizada para bibliotecas ha ido transformándose conforme han ido cambiando las necesidades de la catalogación; y así mismo los formatos MARC 21 y su documentación. A pesar de que la Biblioteca del Congreso desarrolló originalmente el Formato MARC 21 y de que mantiene y publica su documentación, la Biblioteca no realiza cambios o revisiones en forma unilateral.

Existen dos grupos responsables de analizar y revisar la documentación del Formato MARC 21, estos son MARBI y el Comité Consultivo MARC. El Comité para la Información Bibliográfica Legible por Máquina (MARBI) pertenece a la Asociación Norteamericana de Bibliotecas (ALA) y está compuesta por tres representantes de cada una de las divisiones funcionales de la ALA: ALCTS (función de servicios técnicos); LITA (automatización); y RUSA (referencia). La ALA procura garantizar que todo tipo de experiencia y pericia estén representadas en MARBI. El Comité Consultivo MARC está compuesto por representantes de las bibliotecas nacionales, los organismos bibliográficos, grupos de distribuidores, y otras asociaciones bibliotecarias y académicas. MARBI lleva a cabo reuniones en conjunto con el Comité Consultivo MARC

en cada conferencia (anual y semianual) de la ALA. La principal actividad del Comité consiste en la revisión de los documentos para discusión y las propuestas expuestas por o a través de la Biblioteca del Congreso para realizar cambios o adiciones a los formatos MARC, o sobre el desarrollo de nuevos soportes del Formato MARC en áreas emergentes.

En 1987 la Biblioteca del Congreso publicó la primera edición del documento *MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media* con el propósito de ayudar a las bibliotecas y organizaciones que creaban o adquirirían registros MARC; y publicó ediciones subsecuentes en 1990, 1994 y 2000. Este documento ofrece información técnica sobre la estructura de los registros MARC, los conjuntos de caracteres utilizados en ellos, y el formato usado en las cintas magnéticas para distribuir los registros MARC; y está destinado al personal dedicado al diseño y mantenimiento de sistemas para el intercambio y procesamiento de registros MARC.

La proliferación de sistemas de computadora para bibliotecas puso de manifiesto la necesidad de tener una norma para el intercambio de registros MARC 21 en disquete. MARBI y el Comité Consultivo MARC, en colaboración cercana con los distribuidores de datos bibliográficos y de sistemas de automatización de bibliotecas, hicieron diversas propuestas que dieron como resultado cambios y adiciones al formato MARC 21 que permiten especificar en forma completa la transferencia de registros en disquete.

Poco después se desarrollaron también especificaciones para la distribución de registros MARC via archivos de transferencia (FTP). Las especificaciones para la distribución y transferencia a partir de disquetes y FTP se incorporaron en la edición de 1994 de las especificaciones MARC 21.

Los bibliotecarios, los distribuidores comerciales, y quienquiera que distribuye datos catalográficos, deberían poder proveer catalogación MARC 21 en un formato estandar. Los distribuidores de sistemas de automatización de bibliotecas deberían poder ofrecer sistemas diseñados para recibir y procesar en forma correcta registros normalizados MARC. Los bibliotecarios que utilizan computadoras para administrar sus bibliotecas son los beneficiarios directos de todo este trabajo con las normas MARC 21.

### 3.4 Análisis de la Estructura y el Flujo de Datos del Sistema.

#### 3.4.1. Diagrama jerárquico-funcional.

El sistema se divide en varios módulos, los cuales se detallan a continuación en el diagrama jerárquico-funcional.

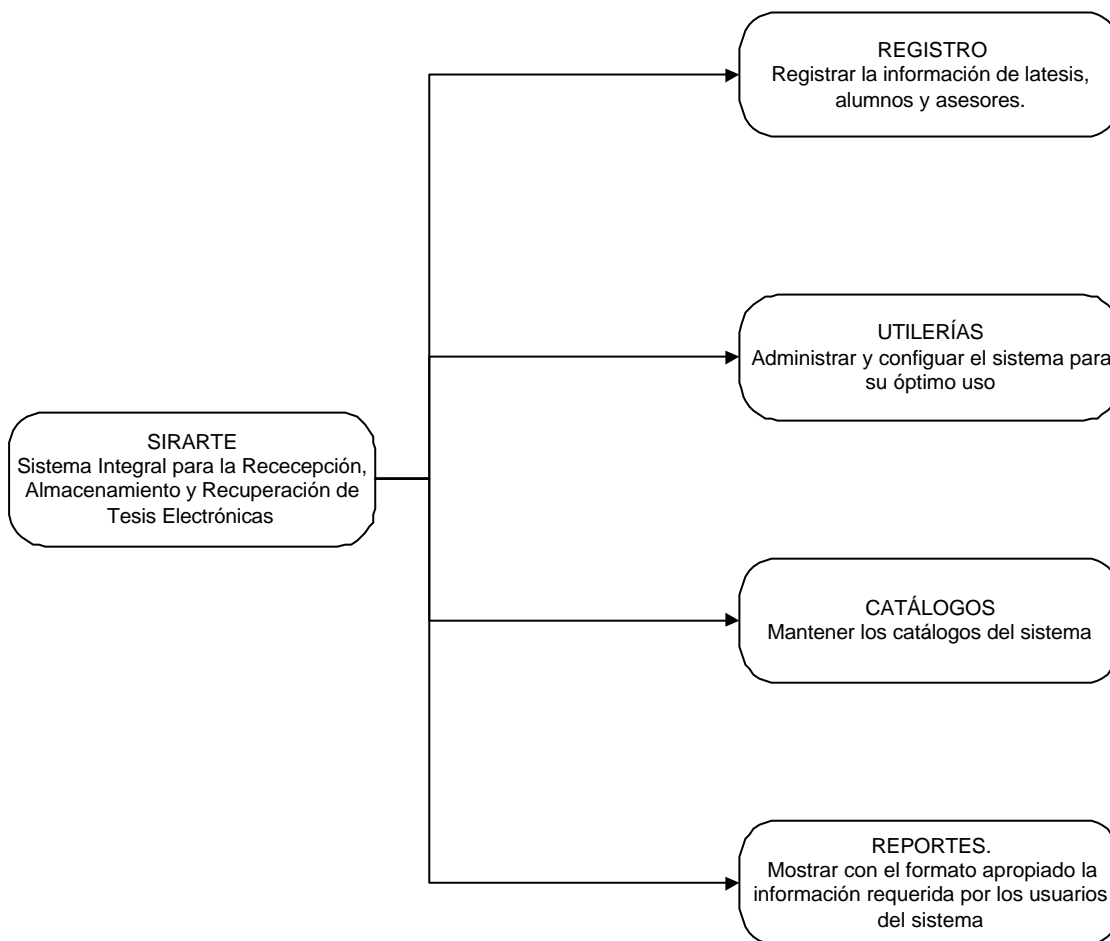


Diagrama Jerárquico –Funcional **Nivel 1.**

A continuación se detalla cada uno de los módulos presentado anteriormente.

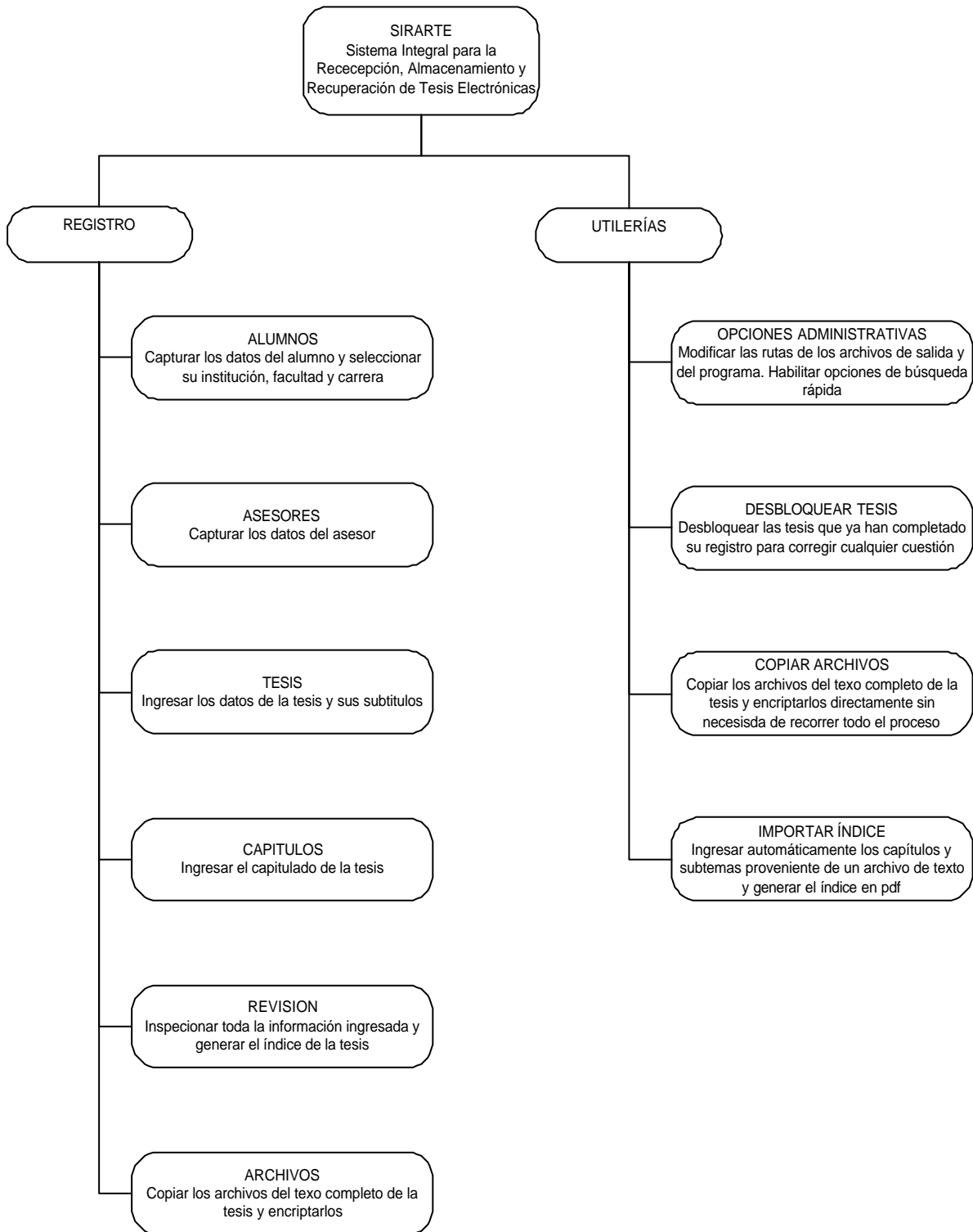


Diagrama 1. Diagrama Jerárquico-Funcional Nivel 2

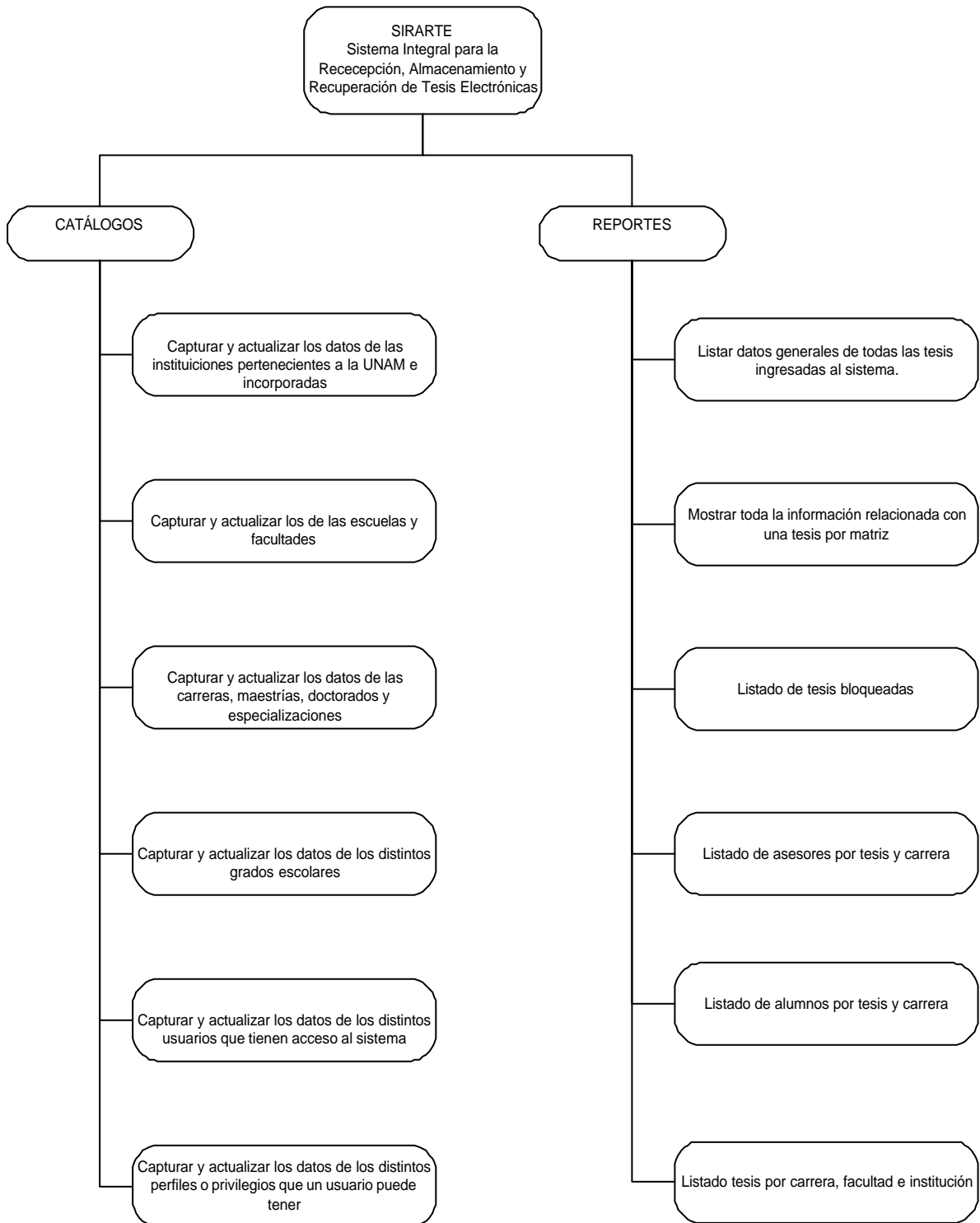


Diagrama 2. Diagrama Jerárquico-Funcional Nivel 2

### 3.4.2. Diagrama de flujo de datos

El Diagrama de Flujo de Datos (DFD) puede ser utilizado para mostrar el procesamiento a distintos niveles de abstracción, desde un bajo nivel de abstracción hasta muy detallado. Puede ser también usado para la descripción arquitectural mostrando el intercambio de datos entre los subsistemas que componen al mismo.

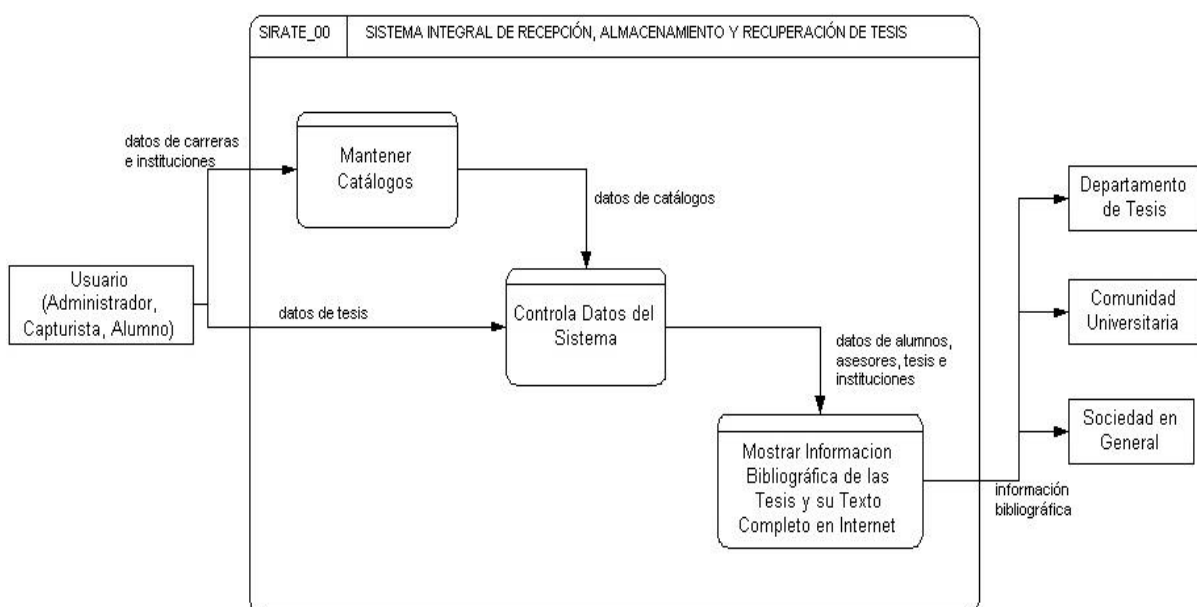


Diagrama 3. Diagrama de Flujo de Datos Nivel 0

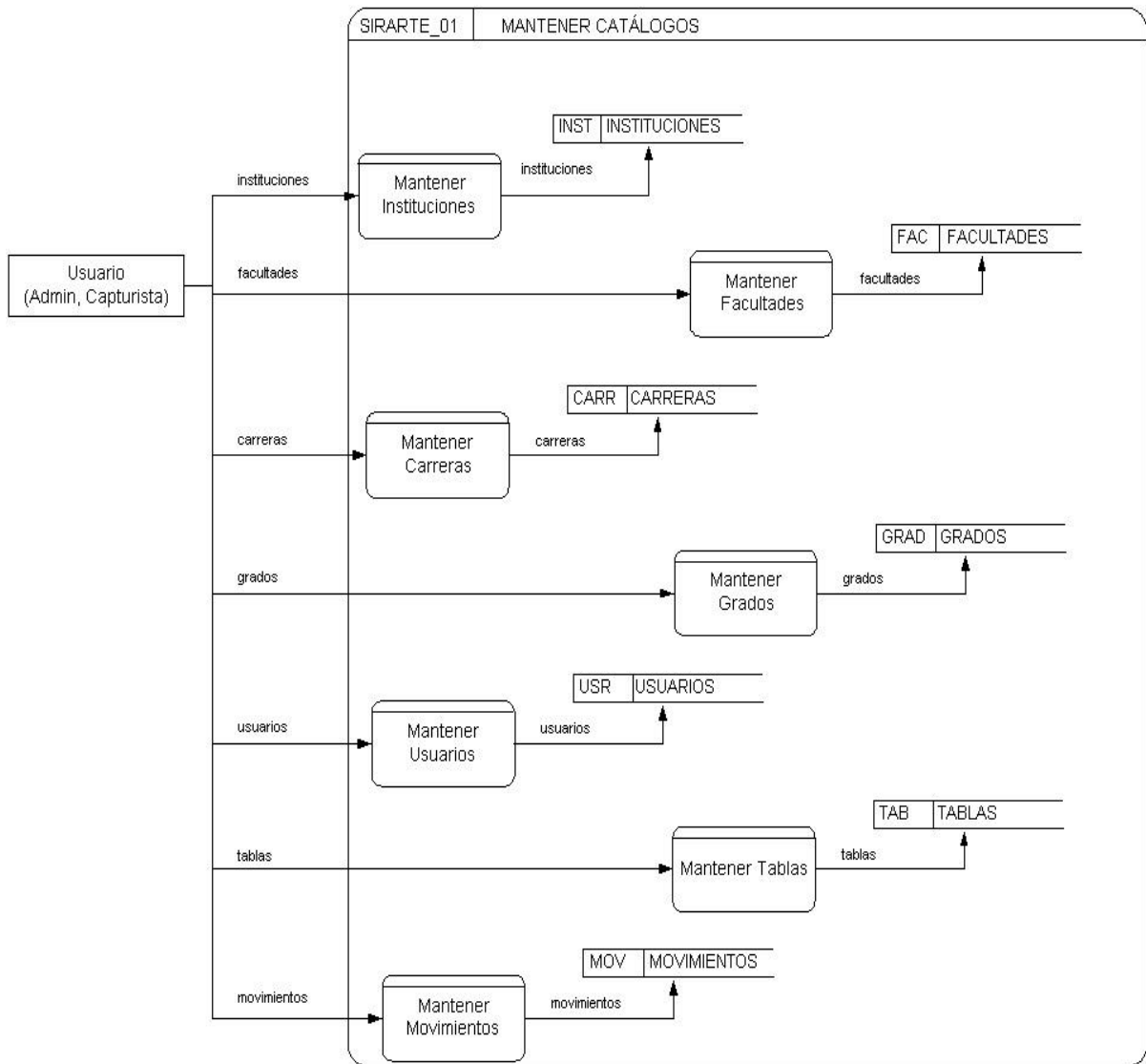


Diagrama 4. Diagrama de Flujo de Datos Nivel 1

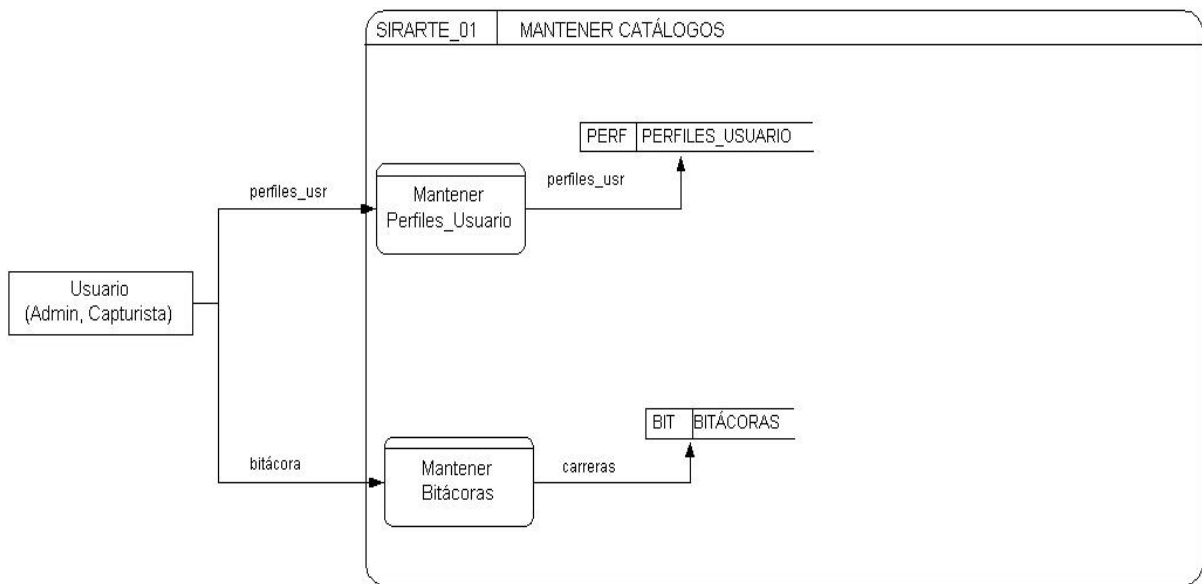


Diagrama 5. Diagrama de Flujo de Datos de Nivel 1

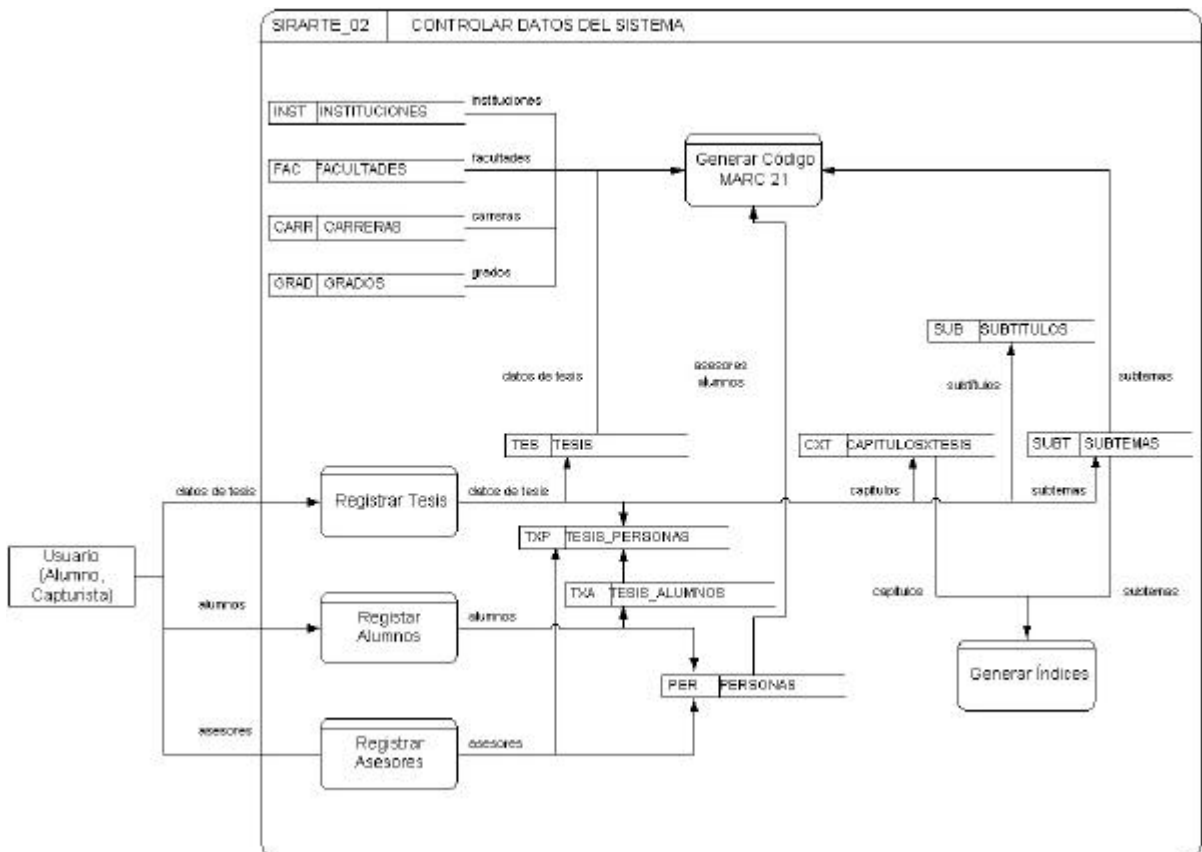


Diagrama 6. DFD de Nivel 2



**CAPÍTULO 4**

---

---

## **DISEÑO DEL SISTEMA**

### **4.1 Descripción del sistema**

El sistema para la recepción, administración y recuperación de tesis electrónicas tiene como principal propósito el dar a conocer a la comunidad universitaria el material que se genera en esta casa de estudios. Para tal propósito se ha desarrollado un sistema integral que sea capaz de encargarse de algunos puntos clave como el registro de los datos de la tesis del tesista y del asesor, modificación de bs datos, otorgar seguridad a archivos PDF, administrar los archivos que el tesista entrega, generar el registro bibliográfico y la visualización de las tesis.

El sistema esta integrado por dos aplicaciones: la aplicación WEB y la interfaz gráfica:

- La aplicación WEB realizado con JSP's se encarga principalmente del registro de los datos de los tesistas. Los datos que se registran en esta parte son los datos de la tesis, los datos del alumno y los datos de los apartados de la tesis (se describe la partición o división de una tesis en el punto 3.4.1).

Los datos que se capturan de la tesis son:

1. Título
2. Subtitulo(s)
3. Número de páginas
4. Año
5. Material adicional
6. Ilustraciones (si/no)

Los datos que se capturan del alumno son:

1. Número de cuenta
2. Nombre
3. Apellido Paterno
4. Apellido Materno
5. Teléfono

Los datos que se capturan de los apartados de la tesis son:

1. Nombre del apartado
  2. Subtemas (no es necesario)
- La interfaz gráfica tiene una estructura cliente-servidor y fue desarrollado en Power Builder. Se utilizará en la Biblioteca Central. Servirá para verificar datos, generar el índice de la tesis en PDF, otorgar seguridad a los archivos en PDF y generar el registro bibliográfico de la misma. Con este proceso quedará terminado el registro de la tesis en la Biblioteca Central y por lo tanto la visualización de la tesis en texto completo será posible al día siguiente.

Estas dos partes están conectadas entre sí ya que utilizan la misma base de datos.

Los datos que se registran en el WEB son visualizados y corregidos en la interfaz gráfica. Este sistema local toma los datos que ya están en la base de datos y solamente se integran otros como los datos del asesor y la institución, la facultad y la carrera para cada uno de los alumnos. En el caso de que ya se encuentre el asesor, solamente se selecciona y se integra a la tesis. Terminando de revisar todos los datos solamente se tendría que generar lo que ya se menciona, que es el índice de la tesis en PDF, generar el registro bibliográfico de la tesis y dar seguridad a los archivos en PDF.

Los datos que se capturan del asesor son:

1. Apellido paterno
2. Apellido Materno
3. Nombre
4. Teléfono

Los datos que se capturan adicionales para cada alumno son:

1. Institución
2. Facultad
3. Carrera

Finalmente se genera un reporte con todos los datos de la tesis en donde se da constancia de que el registro fue completado satisfactoriamente.

#### 4.4.1 División de una tesis

Para mantener un patrón en la división de una tesis se utilizó el ISO 7144 que contiene ciertas reglas establecidas para la correcta partición de una tesis.

La división de una tesis siguiendo la norma ISO 7144 es:

1. Preliminares.
  - a) Cubierta, si se requiere.
  - b) Portada.
  - c) Página de erratas, si procede.
  - d) Resumen.
  - e) Prefacio.
  - f) Índice general.
  - g) Lista de ilustraciones y lista de tablas, si procede.
  - h) Lista de abreviaturas y símbolos, si se necesita.
  - i) Glosario, si es necesario.
2. Texto principal
  - a) Texto principal con las ilustraciones y tablas esenciales.
  - b) Lista de referencias.
3. Anexos.
4. Partes finales
  - a) Índice(s), si hubiere.
  - b) Currículo vitae del autor, si procede.
  - c) Interior y exterior de la cubierta posterior, si se requiere.
  - d) Eventualmente, material anexo.

Basándonos en esta forma de división se hizo una división que utiliza la Dirección General de Bibliotecas hasta el momento, y es la siguiente:

Nombre del Archivo (.doc)	Elementos
A1	<ul style="list-style-type: none"> <li>•Portada, cubierta, pasta o portadilla.</li> <li>•Agradecimientos o dedicatorias.</li> <li>•Pensamientos.</li> </ul>
A2	<ul style="list-style-type: none"> <li>•Tabla de contenido o índice.</li> </ul>
A3	<ul style="list-style-type: none"> <li>•Introducción.</li> <li>•Prefacio.</li> <li>•Prólogo</li> <li>•Resumen o abstract.</li> </ul>
A4 – A(n)	<ul style="list-style-type: none"> <li>•Capítulos (cuantos contenga la tesis).</li> </ul>
A(n+1)	<ul style="list-style-type: none"> <li>•Conclusiones.</li> </ul>
A(n+2)	<ul style="list-style-type: none"> <li>•Anexos o apéndice.</li> <li>•Complementos y/o suplementos.</li> <li>•Glosario y/o glosario de términos.</li> </ul>
A(n+3)	<ul style="list-style-type: none"> <li>•Bibliografía, obras de consulta o referencias.</li> </ul>
A(n+4)	<ul style="list-style-type: none"> <li>•Otros (partes de una tesis no contempladas en los apartados anteriores)</li> </ul>

#### 4.2 Diseño de la base de datos

Para poder manejar la gran cantidad de datos de todos los tesis, de sus asesores y de todos los apartados de la tesis se hace necesaria la utilización de una base de datos que satisfaga los requerimientos del sistema, es decir, que pueda almacenar y relacionar todos esos datos, entre otros más.

La base de datos se diseño de tal manera que la recuperación de los datos se hiciera de una manera fácil y rápida. Además esta adecuada para realizar todas las acciones que el sistema requiera y toma en cuenta todas las reglas para el diseño de bases relacionales.

Para dar una descripción completa de la base de datos, se utilizará el Modelo Relacional que corresponde a la categoría de los Modelos Lógicos basados en Registros, en el cuál se describe la información que contendrá la base de datos por medio de una colección de tablas y de las respectivas relaciones existentes entre ellas.

#### 4.2.1 Definición de tablas

En primer lugar se describirán cada una de las tablas

#### PERSONAS

Una de las tablas más importantes de la base de datos es la de personas, que es la tabla que contiene la información tanto de los tesistas como de los asesores. Se hace lo anterior debido a que un asesor también puede ser en algún momento tesista.

Tipo de llave	Campos	Tipo de datos
PK	RFC (rfc)	char(20)
	Apellido paterno (ape_pat)	varchar2(75)
	Apellido materno (ape_mat)	varchar2(75)
	Nombre (nombre)	varchar2(75)
	Teléfono (telefono)	varchar2(20)
	Número de cuenta (num_cuenta)	varchar2(10)
	Password (password)	varchar2(30)

#### TESIS

Otra de las tablas con bastante importancia para el desarrollo del sistema es la tabla de tesis. En esta tabla se almacenan los datos más importantes de una tesis. El campo denominado matriz es una referencia única para identificar a cada una de las tesis. A continuación se muestran los datos de esta tabla.

Tipo de llave	Campos	Tipo de datos
FK	Matriz de la tesis (matriz)	number(7)
	Título (titulo)	varchar2(250)
	Número de páginas (num_paginas)	varchar2(50)
	Año (anio)	number(4)
	Material adicional (adicional)	varchar2(100)
	Ilustraciones (ilustraciones)	varchar2(10)
	Fecha de registro	Date

## CAPITULOSXTESIS

Esta tabla tiene la característica de almacenar todos los apartados de una tesis. Además podemos identificar que capítulos pertenece a que tesis.

Tipo de llave	Campos	Tipo de datos
FK	Clave del apartado (clave_capitulo)	number(4)
PK,FK	Matriz de la tesis (matriz)	number(7)
	Nombre del apartado (nombre)	varchar2(120)

## SUBTÍTULOS

Debido a que una tesis puede tener uno o más subtítulos se creo esta tabla que tiene los subtítulos que se asocian a cada una de las tesis. Al igual que en la tabla de CapituloXTesis con esta tabla podemos saber que subtítulos pertenecen a que tesis.

Tipo de llave	Campos	Tipo de datos
PK	Identificador de subtítulo (id_subtitulo)	number(4)
PK, FK	Matriz de la tesis (matriz)	number(7)
	Nombre del Subtítulo (subtitulo)	varchar2(120)

## SUBTEMAS

En esta tabla se almacenan los subtemas de cada apartado que pudiera contener subtemas, es decir, cuando un apartado sea un capítulo. Con esta tabla es relativamente fácil saber cuales son los subtemas para cada uno de los capítulo.

Tipo de llave	Campos	Tipo de datos
PK, FK	Identificador para cada apartado de la tesis (clave_capitulo)	number(4)
PK, FK	Matriz de la tesis (matriz)	number(7)
PK	Identificador para cada subtema (id_subtema)	number(4)
	Nombre de cada subtema (subtema)	varchar2(250)

## TESIS\_ALUMNOS

En la tabla de tesis\_alumnos se van a almacenar los alumnos que se den de alta para poder registrarse por medio de la WEB. En esta tabla se hace la relación de que tesis corresponde a que alumno.

Tipo de llave	Campos	Tipo de datos
PK, FK	Matriz de la tesis (matriz)	number(7)
PK, FK	RFC del alumno (rfc_alumno)	char(20)
	Fecha de registro (fecha_registro)	Date

## INSTITUCIONES

En la tabla de instituciones están todas las instituciones de la UNAM y también de las escuelas incorporadas. Por ejemplo, Universidad del Valle de Mexico. Plantel San Rafael, Universidad Femenina de México, etc. Como ya se menciono, el alumno al registrarse en la aplicación gráfica selecciona una institución.

Tipo de llave	Campos	Tipo de datos
FK	Clave para la institución (clave_institución)	number(5)
	Nombre de la institución (nombre_institución)	varchar2(100)
	Clave única de la institución que servirá para formar el archivo con formato MARC (clave_marc_ins)	number(5)

## FACULTADES

Todas las facultades se almacenan en esta tabla. De esta tabla el alumno también selecciona alguna.

Tipo de llave	Campos	Tipo de datos
FK	Clave para la facultad (clave_facultad)	number(9)
FK	Clave para la institución (clave_institución)	number(5)
	Nombre de la facultad (nombre_facultad)	varchar2(100)
	Clave única de la facultad que servirá para formar el archivo con formato MARC (clave_marc_fac)	number(5)

## CARRERAS

Todas las carreras existentes en la UNAM y sus escuelas incorporadas se encuentran almacenadas en esta tabla. Al igual que en la tabla de instituciones y facultades, el alumno al registrarse selecciona una carrera.



Tipo de llave	Campos	Tipo de datos
FK	Clave para la carrera (clave_carrera)	number(9)
FK	Clave del grado a obtener (clave_grado)	Integer
FK	Clave para la facultad (clave_facultad)	number(9)
	Nombre de la carrera (nombre_carrera)	varchar2(100)
	Clave única de la carrera que servirá para formar el archivo con formato MARC (clave_marc_car)	number(5)

## GRADOS

En esta tabla se almacenan todos los grados existentes, en este caso: licenciatura, maestría, doctorado y especialidad.

Tipo de llave	Campos	Tipo de datos
PK	Clave que identifica el grado (clave_grado)	Integer
	Nombre de cada grado (nombre_grado)	varchar2(50)

## TESIS\_PERSONAS

Esta tabla nos ofrece una visualización en cierta forma completa de la relación existente entre asesores, tesis, alumnos y la carrera de cada alumno. Se formo esta tabla porque puede darse el caso de que un asesor sea también tesista o de que un tesista tenga una o más tesis y con esta tabla nos permite darnos cuenta de este tipo de casos.

Tipo de llave	Campos	Tipo de datos
PK, FK	Matriz de la tesis (matriz)	number(7)
PK, FK	RFC del alumno (RFC_alumno)	char(20)
PK, FK	RFC del asesor (RFC_asesor)	char(20)
FK	Clave de la carrera (clave_carrera)	number(9)
	Fecha de captura (fecha_captura)	Date

## USUARIOS

En esta tabla se almacena la información de los usuarios que pueden entrar al sistema. Por lo tanto se les otorga un login y un password además de un perfil en la tabla de perfiles\_usuario.

Tipo de llave	Campos	Tipo de datos
PK	Login asignado al usuario (login)	varchar2(15)
FK	Identifica el perfil del usuario (id_perfil_usuario)	number
	Password asignado al usuario (password)	char(15)

## PERFILES\_USUARIOS

En esta tabla se definen todos los perfiles que pudieran tener los diversos usuarios que utilicen la interfaz gráfica. Hasta el momento solo se tienen usuarios tipo administrador y normales. Los administradores pueden ver todas las partes de la aplicación y los normales solo algunas. Con lo anterior se le otorga seguridad al sistema.

Tipo de llave	Campos	Tipo de datos
PK	Identifica el perfil del usuario (id_perfil_usuario)	number
	Perfil del usuario (perfil)	varchar2(35)

## MOVIMIENTOS

Siguiendo con ciertas políticas de seguridad y control de la información se formo la tabla de movimientos, la cual almacena los movimientos que podrían llegar a darse en la base de datos. Los principales movimientos serían: altas, bajas y cambios.

Tipo de llave	Campos	Tipo de datos
PK	Identificador para cada movimiento (id_movimiento)	integer
	Tipo de movimiento (movimiento)	varchar2(50)

## TABLAS

En esta tabla se almacenan los nombres de las tablas que están involucradas en los movimientos que ya se mencionaron. Los nombres de las tablas son: personas, tesis, subtemas, subtítulos y capitulosxtesis.

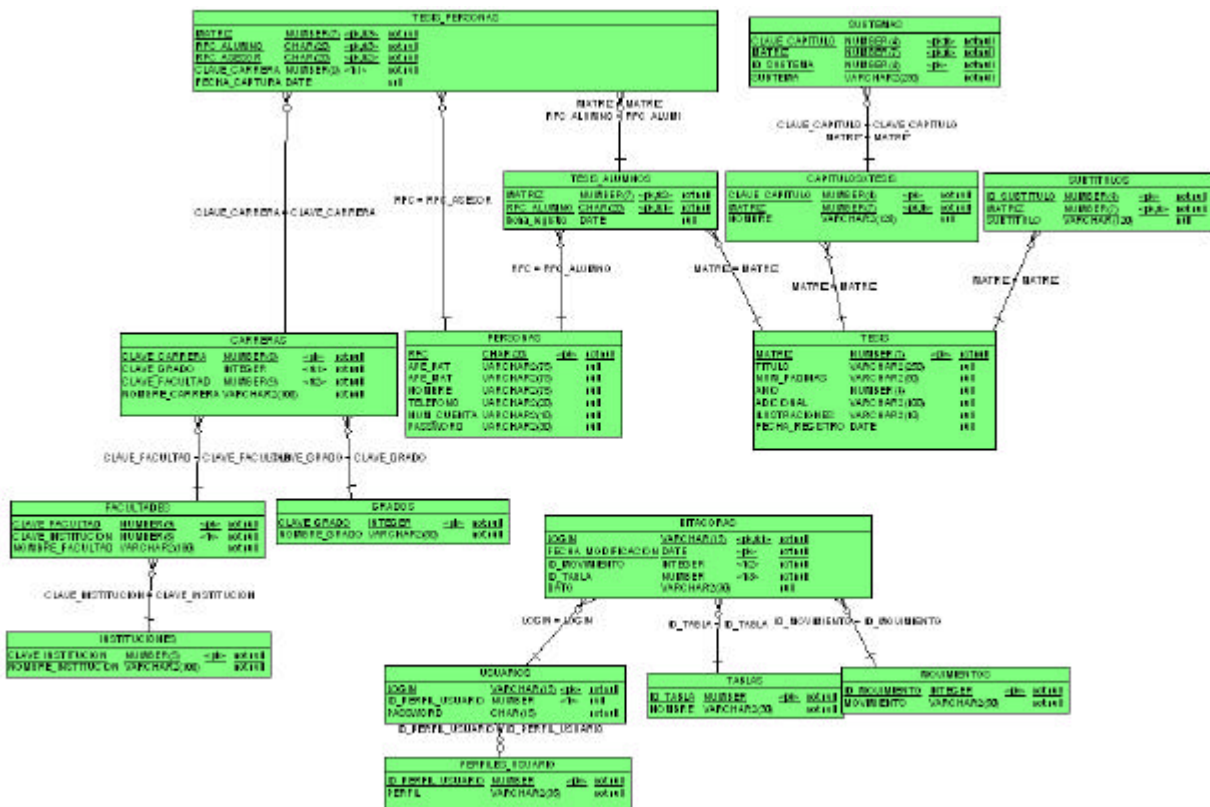
Tipo de llave	Campos	Tipo de datos
PK	Identificador para cada tabla (id_tabla)	number
	Nombre de las tabla (nombre)	varchar2(30)

## BITACORAS

En la tabla de bitácoras se tienen todos los datos necesarios para poder identificar que persona hizo algún cambio, la fecha de modificación del cambio, el tipo de movimiento que se hizo, la tabla que fue afectada y el identificador para cada tabla.

Tipo de llave	Campos	Tipo de datos
PK, FK	Login de la persona que hizo el movimiento (login)	varchar2(15)
FK	Fecha de modificación (fecha_modificación)	date
FK	Identificador del movimiento (id_movimiento)	integer
FK	Identificador de la tabla (id_tabla)	number
	Identificador de la tabla afectada (dato)	varchar2(30)

### 4.2.2 Diagrama Entidad Relación



### 4.3 Características de los sistemas manejadores de bases de datos

PostgreSQL, Oracle, Informix y Sybase están sumamente parejos en manejo RDBMS en plataformas de 32 bits. En plataformas de 64 bits, PostgreSQL e Informix son notablemente superiores que el resto. En determinadas circunstancias de modelo de datos y tráfico de transacciones PostgreSQL es notablemente superior a Informix y Oracle; y en situaciones comunes no tiene nada que envidiarles. Manejan cómodamente bases de datos grandes superiores a los 40 Gb.

SQL Server no posee la potencia o la funcionalidad que Oracle, por lo que la velocidad, exceptuando en entornos puramente de Microsoft se ve bastante decelerada al ejecutar scripts de cierta extensión ( esto es debido, a su gran facilidad de uso, que contrariamente impide que puedas configurarlo a la perfección ). Tampoco es multiplataforma, no se puede instalar en Novell NetWare, Linux, Solaris, UX, etc, por lo que este DBMS deja mucho que desear.

Si hablamos de una base de datos ORDBMS, Informix es superior a PostgreSQL solamente porque incluye una familia de tipos agregados más numerosa (pero igual los puedes programar en PostgreSQL); de hecho, lo que Informix es hoy en ORDBMS lo debe a Illustra, versión comercial de PostgreSQL 95.

PostgreSQL es la base de datos de fuente abierta que más se aproxima a las capacidades ofrecidas por sistemas manejadores de bases de datos propietarios. En este momento, su falta de replicación y de manejo de información textual puede ser un impedimento para su amplia aceptación. Sin embargo, si estas capacidades no son fundamentales, es un sistema muy estable y sólido.

MySQL, por el otro lado, siempre ha tenido un fuerte sesgo hacia la eficiencia en tiempo, a costa de capacidades avanzadas. Sin embargo es un simple manejador de archivos. La ausencia de Integridad Referencial, transacciones, sub-selects, triggers y vistas lo hacen poco adecuado para proyectos grandes. Acces y Fox Pro también entran dentro de esta categoría.

InterBase se origina en un ambiente comercial y por lo tanto provee gran parte de las capacidades esperadas por un sistema manejador de base de datos propietario. Sin embargo, su desarrollo comenzó relativamente tarde y no provee todas las capacidades que se esperarían.

Las bases de datos propietarias tales como Oracle, Informix y Sybase le llevan una gran ventaja a las bases de datos de fuente abierta en el área de la administración del sistema manejador de bases de datos. Las herramientas incluidas con el producto y las herramientas de tercera parte para administración son muy poderosas, cuentan con interfaces gráficas, y cubren virtualmente todos los aspectos posibles a administrar. Las bases de datos de fuente abierta, cuentan con herramientas de administración de línea de comandos y en esencia tiene una interfaz muy poco amigable.

### **4.3.1 Oracle**

#### **4.3.1.1 Introducción a Oracle**

Oracle Systems Corporation, fabrica productos software y proporciona servicios para la gestión electrónica de la información. Oracle es un proveedor de software a nivel mundial, su software funciona en más de 100 plataformas diferentes. Oracle8 es la primera versión de la base de datos de Oracle que incorpora la tecnología orientada a objetos. Se trata de una base de datos objeto-relacional, dado que esta implementación no es una base de datos orientada a objetos pura, ni tampoco es una base de datos relacional, representa un híbrido de ambas.

#### **4.3.1.2 Arquitectura de Oracle8**

La arquitectura de Oracle8 es cliente-servidor: los usuarios acceden a la base de datos desde su computadora personal (cliente) a través de una red, y la base de datos se encuentra en una computadora diferente (servidor). El servidor es la piedra angular de este sistema que a continuación se describe brevemente.

### **Oracle8 Server**

El servidor de Oracle8 es un entorno avanzado de gestión de la información. Permite almacenar grandes cantidades de datos y proporciona a los usuarios acceso

rápido a los mismos. También, permite la compartición de datos entre aplicaciones; la información se almacena en un cierto lugar y puede ser utilizada por muchos sistemas.

#### **4.3.1.3 Funciones de Oracle8.**

Oracle8 posee una amplia gama de funcionalidades, las más importantes se comentan a continuación:

- Mecanismos de seguridad

Los sofisticados mecanismos de seguridad de Oracle controlan el acceso a los datos utilizando un conjunto de privilegios.

- Realización de copias de seguridad y recuperación

Oracle proporciona sofisticados procedimientos de realización de copias de seguridad y recuperación de los datos.

- Gestión del espacio

Oracle ofrece una gestión flexible del espacio. Se puede asignar un cierto espacio de disco para el almacenamiento de los datos, y controlar las subsiguientes asignaciones instruyendo a Oracle sobre cuánto espacio debe reservar para los requerimientos futuros. También tiene una serie de características que fueron diseñadas teniendo en cuenta las necesidades de las bases de datos de gran tamaño.

- Conectividad de carácter abierto

Oracle proporciona conectividad hacia y desde paquetes software de otros fabricantes. Utilizando extensiones a la base de datos Oracle, se puede trabajar con información almacenada con otros sistemas de bases de datos, como DB2 de IBM o Microsoft Access. También se pueden almacenar los datos en la base de datos de Oracle y acceder a ellos desde otros paquetes software, como Visual Basic.

- Herramientas de desarrollo

El servidor Oracle, al que normalmente se denomina motor de la base de datos, funciona con un amplio conjunto de herramientas de desarrollo, herramientas de consulta para usuario final, aplicaciones comerciales y herramientas de gestión de la información de ámbito corporativo.

- Mecanismos de integridad

El servidor Oracle también se encarga de la integridad de los datos. Si se produce cualquier tipo de fallo mientras un usuario está cambiando los datos en una base de datos, este tiene la capacidad de deshacer o cancelar cualquier transacción sospechosa.

- Componente procedimental

A partir de Oracle7, esta opción pasó a formar parte del núcleo del Servidor. El fundamento de esta opción es el lenguaje de programación de Oracle PL/SQL. Con esta opción se pueden implementar las funcionalidades siguientes:

1. Procedimientos almacenados. Se pueden almacenar programas (o segmentos de código) en la base de datos.

2. Disparadores de bases de datos (triggers). Son segmentos de código almacenados en la base de datos, y que se disparan como respuesta a sucesos que tienen lugar en las aplicaciones.

3. Paquetes. Los procedimientos se suelen agrupar, almacenándose el código como una única unidad de programación en la base de datos.

- Componente de procesamiento distribuido

En muchos sistemas hay datos que residen en diferentes computadoras, Oracle permite gracias a la transparencia de ubicación, que todos los usuarios puedan acceder a dicha información sin necesidad de saber en que ordenador se encuentra la información.

- Componente de consulta en paralelo:

Se utiliza con computadoras de múltiples UCP, y permite que el procesamiento de las consultas sea repartido entre múltiples procesos servidores.

#### **4.3.1.4 Estructura física de una Base de datos en Oracle**

Una base de datos es una colección de datos relacionados que son utilizados y recuperados conjuntamente por uno o más sistemas de aplicaciones. La situación física e implementación de la base de datos es transparente para los programas de aplicaciones, y en realidad, se puede mover y reestructurar la base de datos física sin afectar a los programas.

Físicamente, en su forma más simple, una base de datos Oracle no es más que un conjunto de archivos que se encuentran en alguna parte del disco. La situación física de estos archivos es irrelevante para la función de la base de datos (aunque es importante para su rendimiento). Los archivos son binarios y sólo se puede acceder a ellos utilizando el software del núcleo de Oracle. La consulta de los datos de los archivos de la base de datos se realiza generalmente con una de las herramientas de Oracle (tales como SQL \* Plus).

Lógicamente, la base de datos está dividida en un conjunto de cuentas de usuario de Oracle (esquemas), cada una de las cuales está identificada por un nombre de usuario (username) y una contraseña (password) única, exclusiva para esa base de datos.

Con frecuencia se confunde una cuenta de usuario Oracle con una base de datos, pero esto no es estrictamente correcto. Se pueden usar dos cuentas de usuario Oracle para contener datos para dos sistemas de aplicaciones completamente diferentes; se tendrían dos bases de datos lógicas implementadas en la misma base de datos física utilizando dos cuentas de usuario Oracle.

### **Archivos Oracle**

Hay tres conjuntos de archivos en el disco que componen una base de datos, son los siguientes:

- Archivos de bases de datos.
- Archivos de control.
- Registros de rehacer.

Los más importantes son los archivos de bases de datos, en los cuales reside la base de datos. Los archivos de control y los registros de rehacer soportan la funcionalidad de la arquitectura en sí.

Los tres conjuntos de archivos deben estar presentes, abiertos y disponibles para que Oracle pueda utilizar cualquier dato de la base de datos.



#### 4.4 Diseño de la interfaz de usuario

##### 4.4.1 Diseño de la interfaz gráfica

La aplicación cliente del sistema deberá contar con una interfaz gráfica amigable al usuario e intuitiva, para de esta forma reducir el tiempo dedicado a la capacitación de los usuarios, pero al mismo tiempo deberá tener una serie de atajos y accesos rápidos para los usuarios expertos. La lógica con la que debe contar cada una de estas pantallas se detalla a continuación:

En primer lugar debe existir un proceso de validación del usuario, el cual deberá evaluar los permisos o privilegios con los que este fue dado de alta. Con esta información habilita o deshabilita ciertas opciones del sistema, como se muestra en la figura:

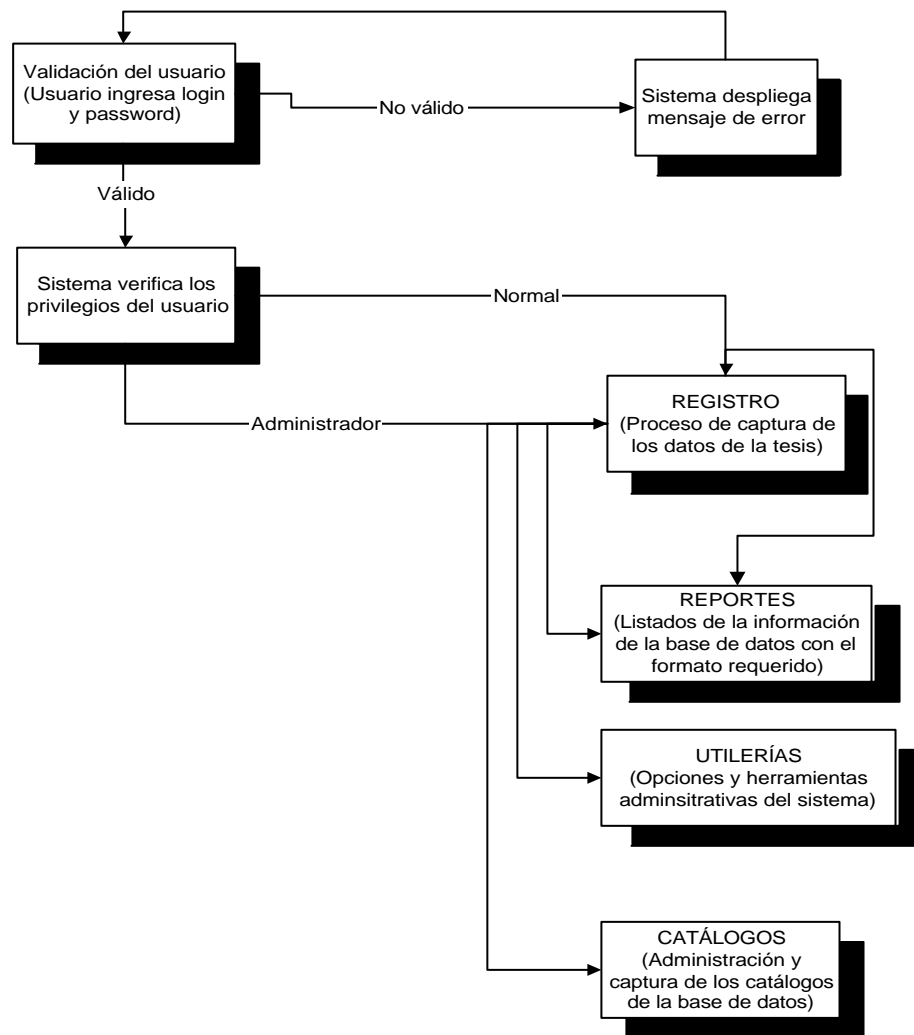
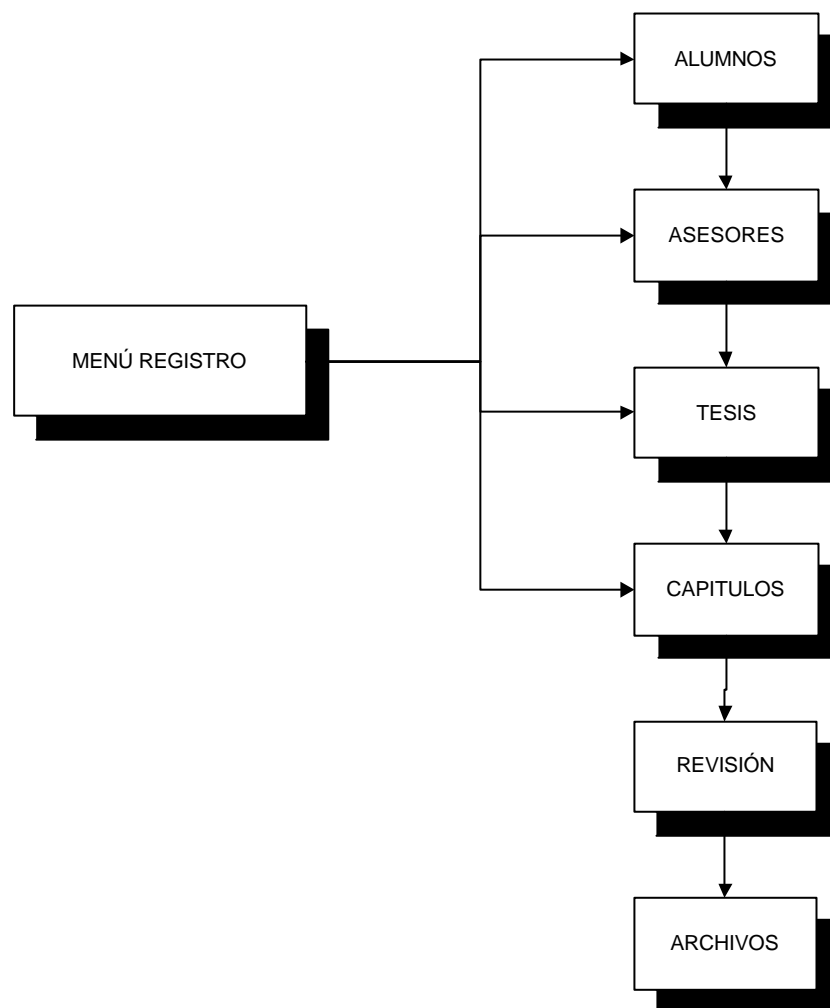


Diagrama 4.1 Flujo de bloques del sistema.

#### 4.4.1.1 Módulo de registro

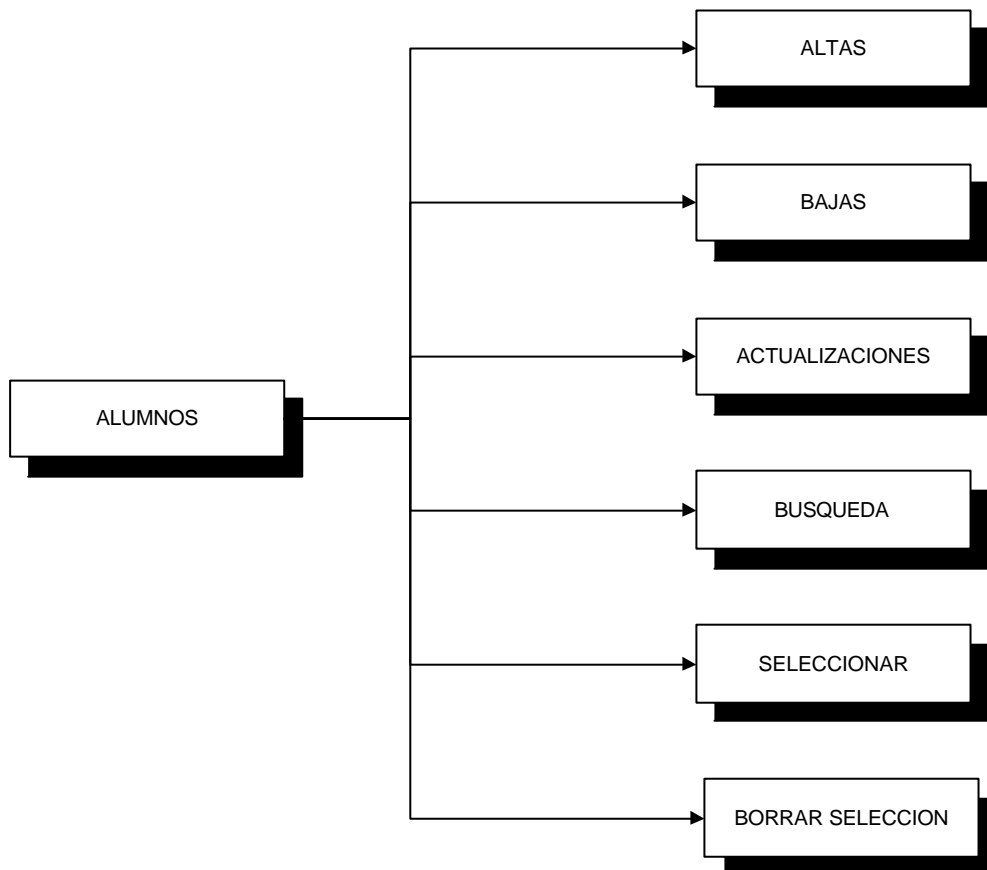
Este constituye el módulo principal del sistema, ya que a partir de este se sigue todo el proceso para el registro completo de una tesis. Como se observa en el diagrama, es posible acceder directamente a algunos módulos, esto permite modificar algún dato específica que hubiera que corregir.



*Diagrama 4.2 Flujo del módulo de registro.*

#### 4.4.1.2 Módulo de alumnos

Este módulo permite capturar toda la información referente al alumno, y además realizar bajas, actualizaciones búsquedas y selección de uno o mas alumnos.



**Diagrama 4.3** Flujo del módulo de alumnos.

4.4.1.2.1 Altas alumnos

Altas de uno o mas alumnos, verificando los datos insertados y evitando la redundancia de los mismos.

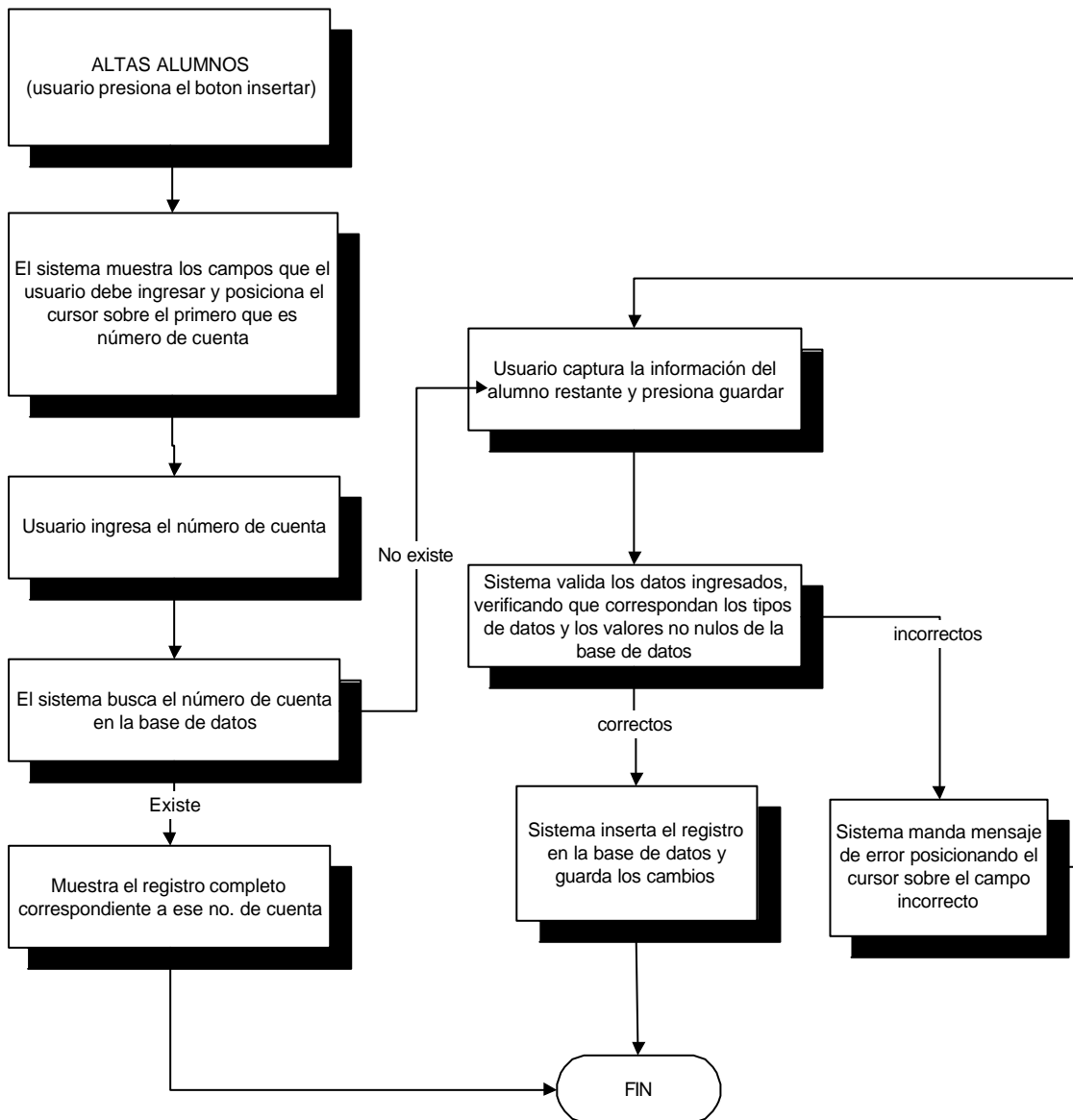


Diagrama 4.4 Flujo de la alta de alumnos.

#### 4.4.1.2.2 Bajas alumnos

Permite al usuario eliminar uno o más alumnos siempre y cuando no tengan una o más matrices asociadas a estos.

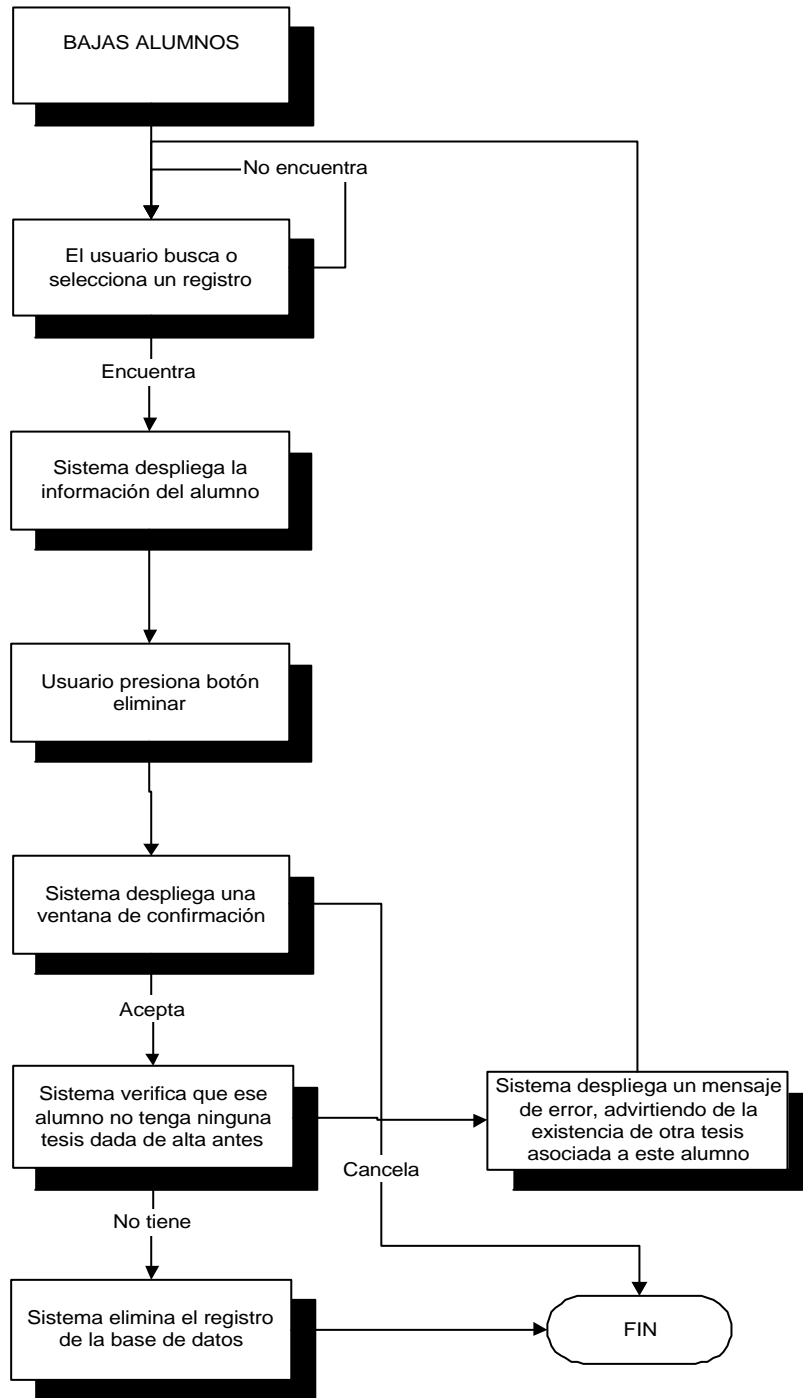


Diagrama 4.5 Flujo de la baja de alumnos.

### 4.4.1.2.3 Actualizaciones alumnos

Permite hacer modificaciones a registro dados alta en la base de datos.

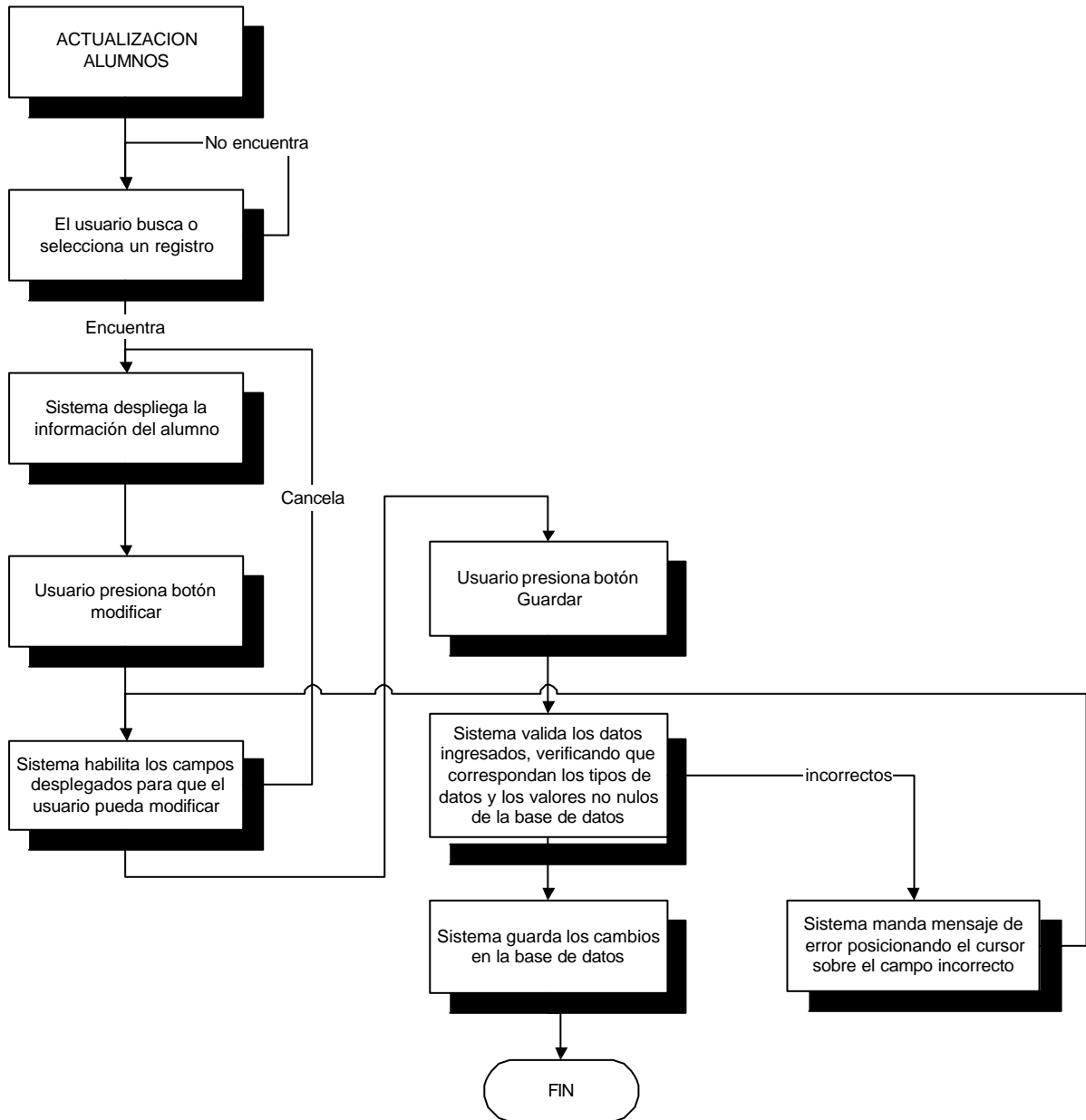


Diagrama 4.6 Flujo de las actualizaciones de alumnos.

4.4.1.2.4 Búsqueda alumnos

Permite realizar búsquedas directas y filtrados la tabla de alumnos.

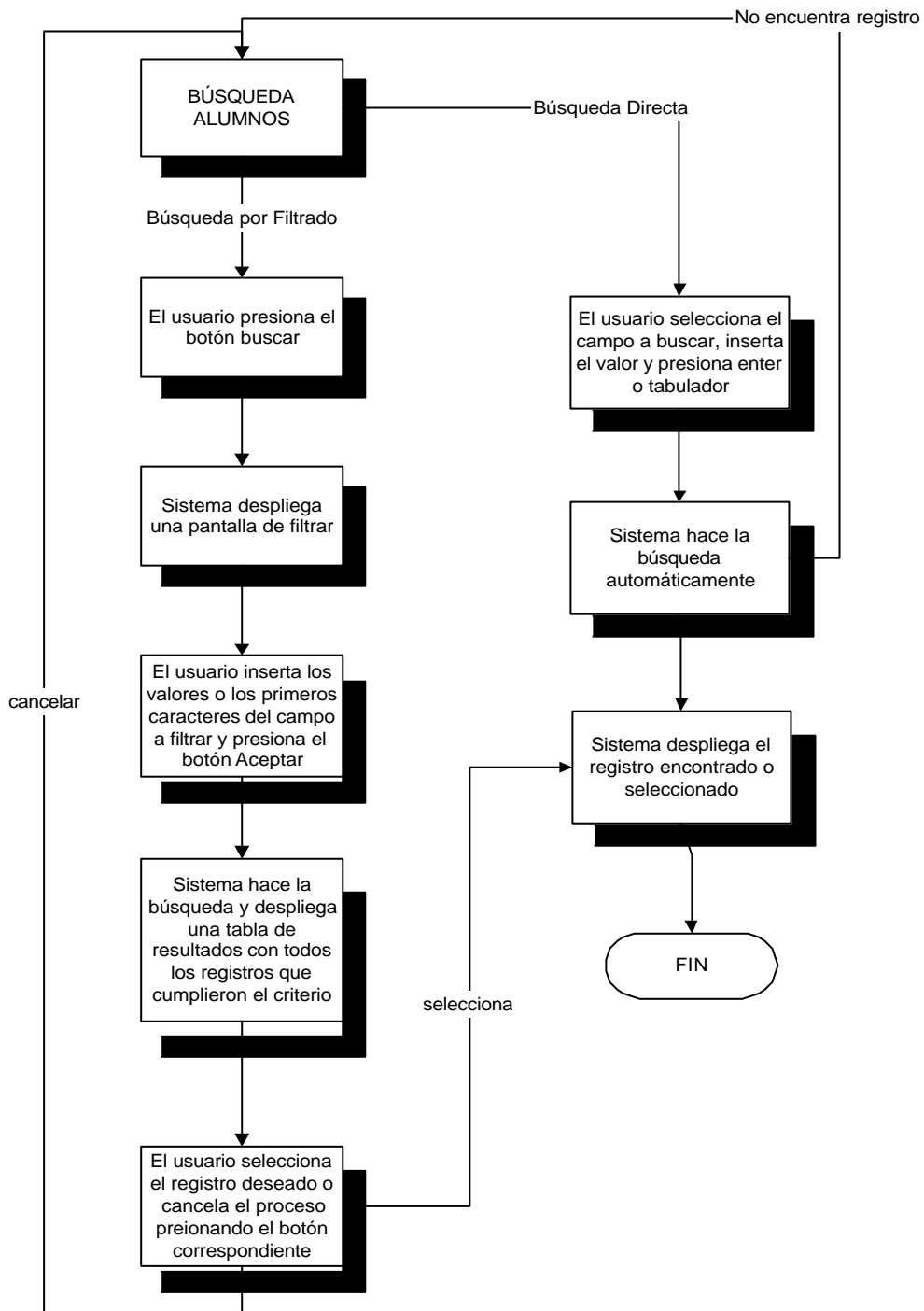


Diagrama 4.7 Flujo para la búsqueda de alumnos.

#### 4.4.1.2.5 Seleccionar alumnos

Permite seleccionar uno o más alumnos para asociarlos a una tesis. También puede borrarse esa selección.

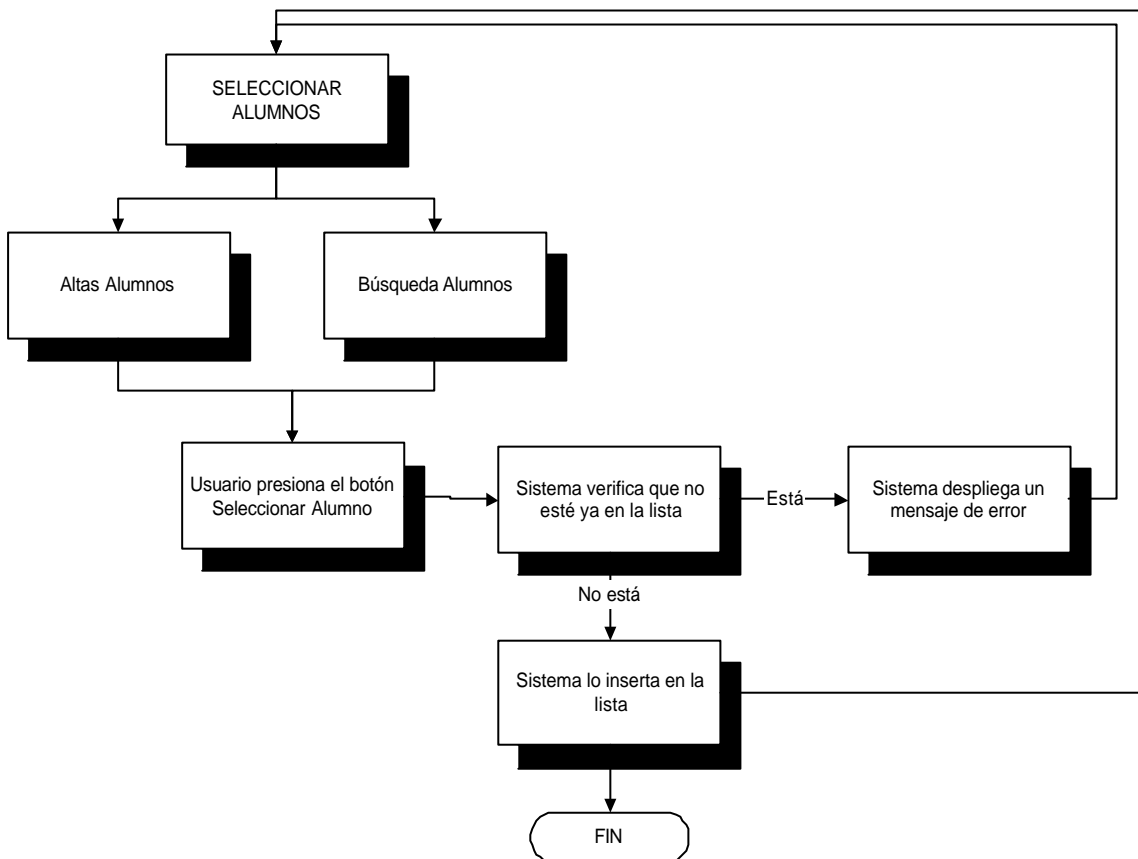


Diagrama 4.8 Flujo de la selección de alumnos.



4.4.1.2.6 Borrar selección alumnos

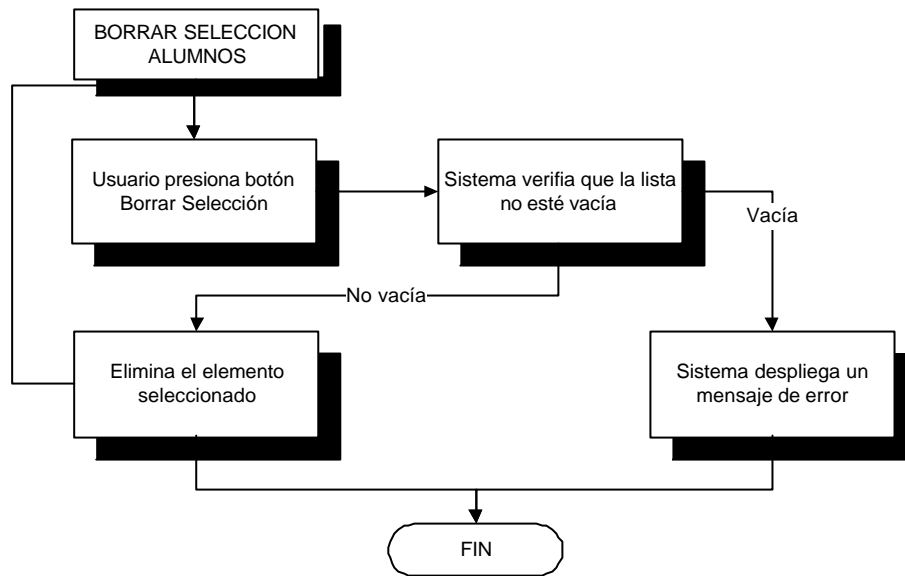
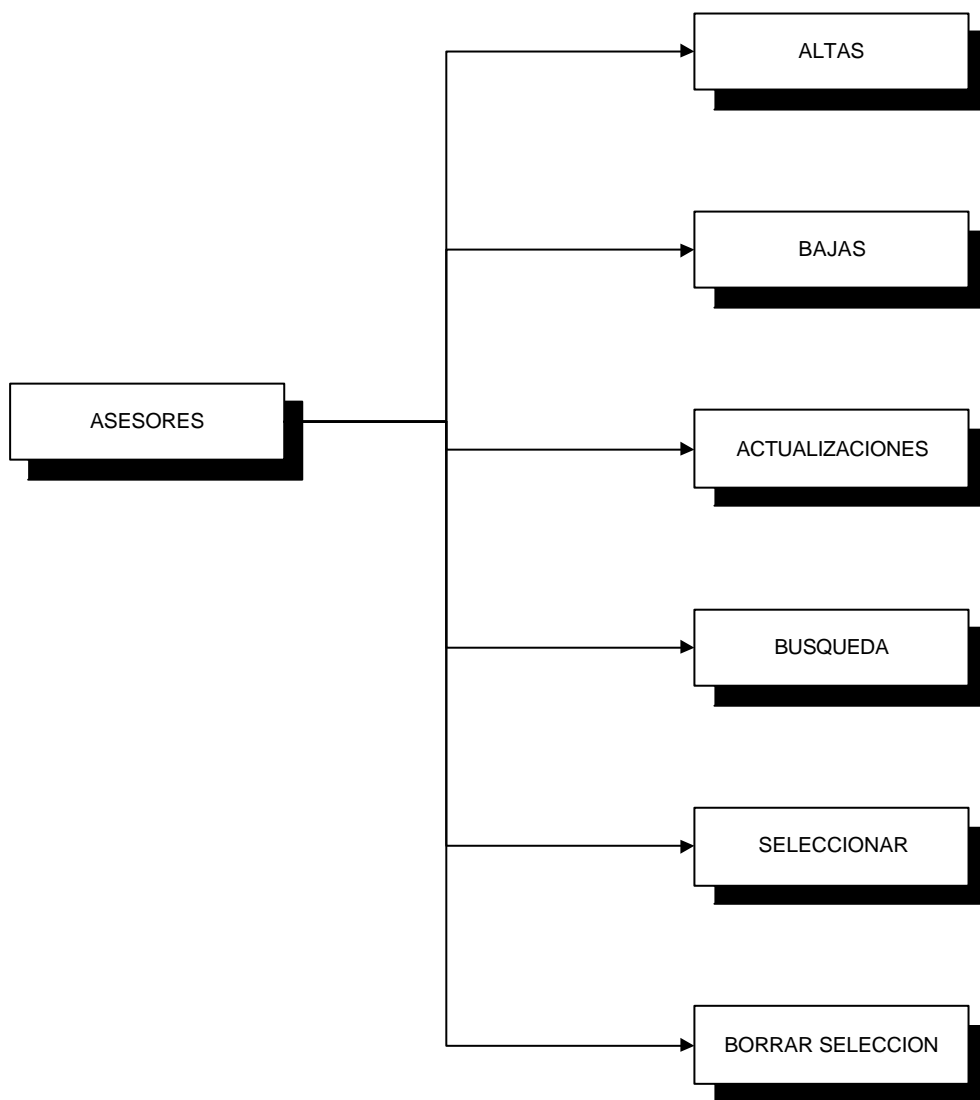


Diagrama 4.9 Flujo para borrar selección de alumnos.

#### 4.4.1.3 Módulo de asesores.

Aquí el usuario puede capturar toda la información referente al asesor y en caso de requerirlo dar de baja o hacer las modificaciones pertinentes. Además puede realizar búsquedas directas y por filtrado.



**Diagrama 4.10** Flujo del módulo de asesores.

4.4.1.3.1 Altas asesores

Ingreso de uno o mas asesores nuevos a la base de datos.

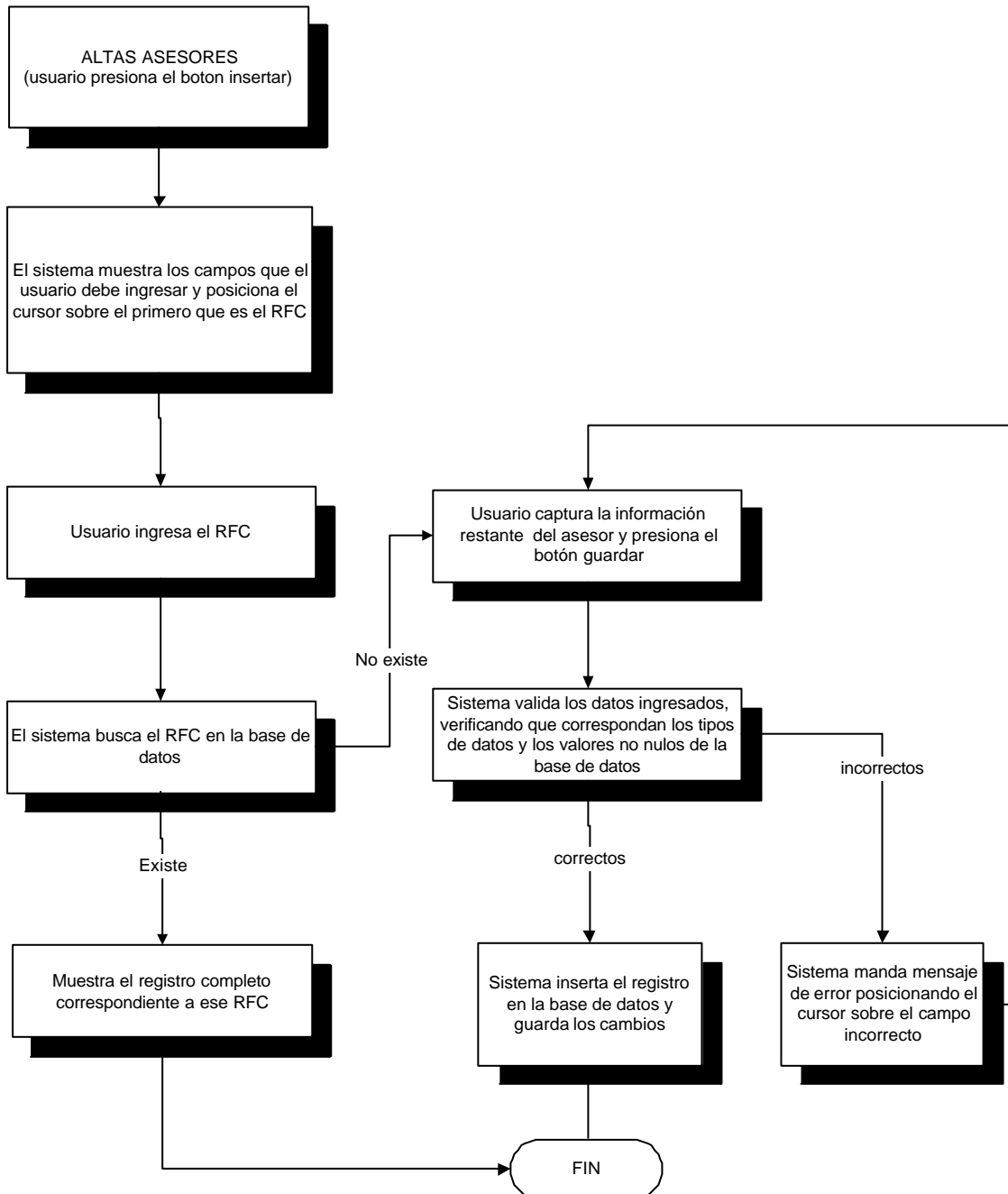
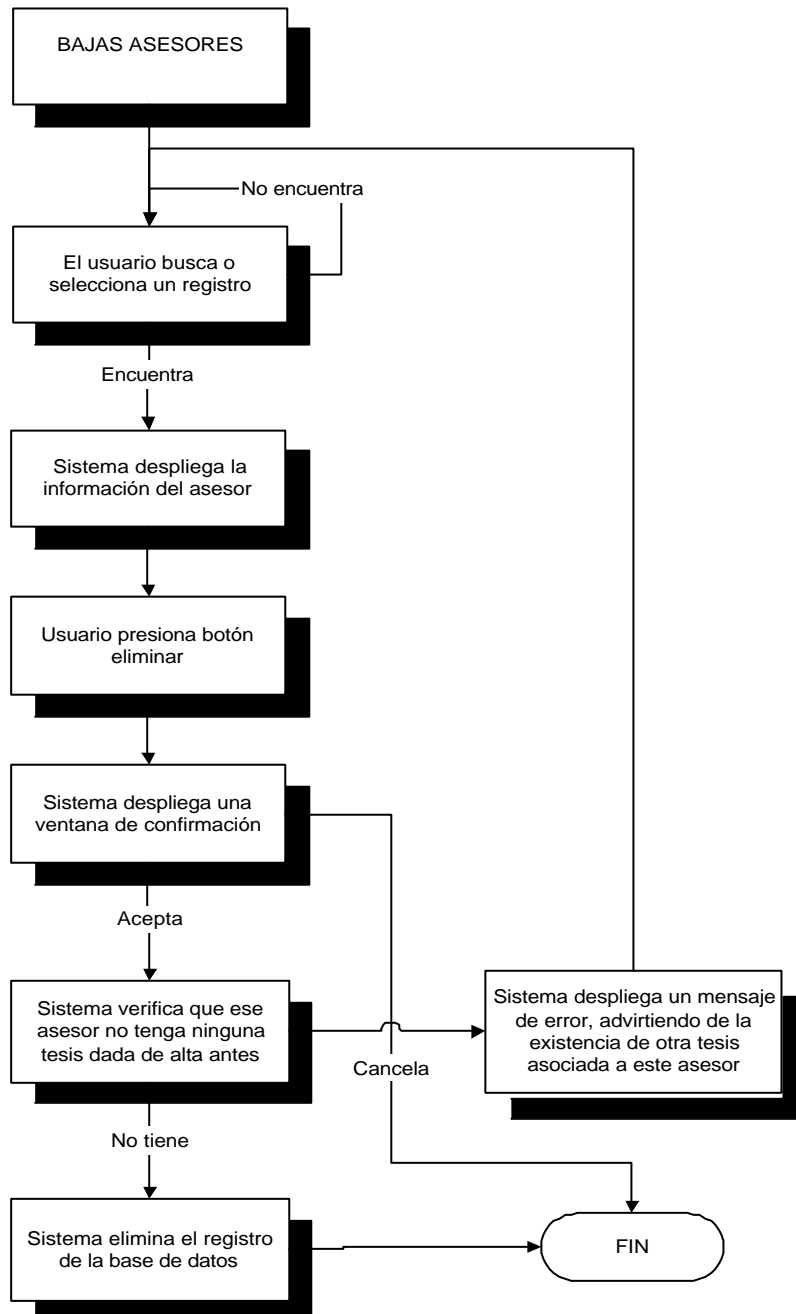


Diagrama 4.11 Flujo para la alta de asesores.

**4.4.1.3.2 Bajas asesores**

Eliminar un o más asesores de la base de datos, respetando la integridad referencial.



**Diagrama 4.12** Flujo para la baja de asesores.

### 3.8.1.3.3 Actualizaciones asesores

Permite realizar cambios en bs datos de algún asesor y verlos reflejados en la base de datos.

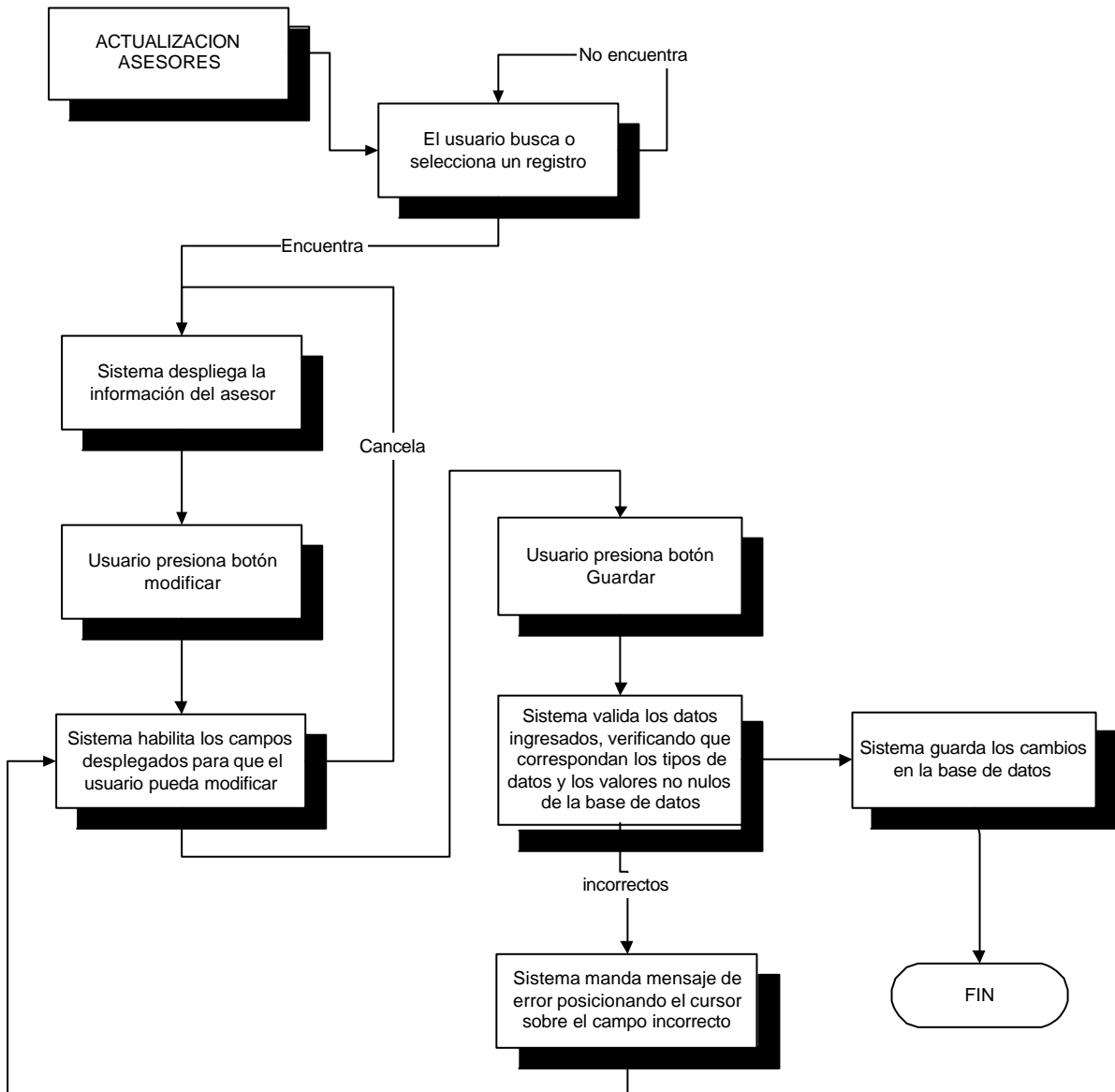


Diagrama 4.13 Flujo para la actualización de asesores.

4.4.1.3.4 Búsquedas asesores

Búsquedas directas y filtrados la tabla de asesores.

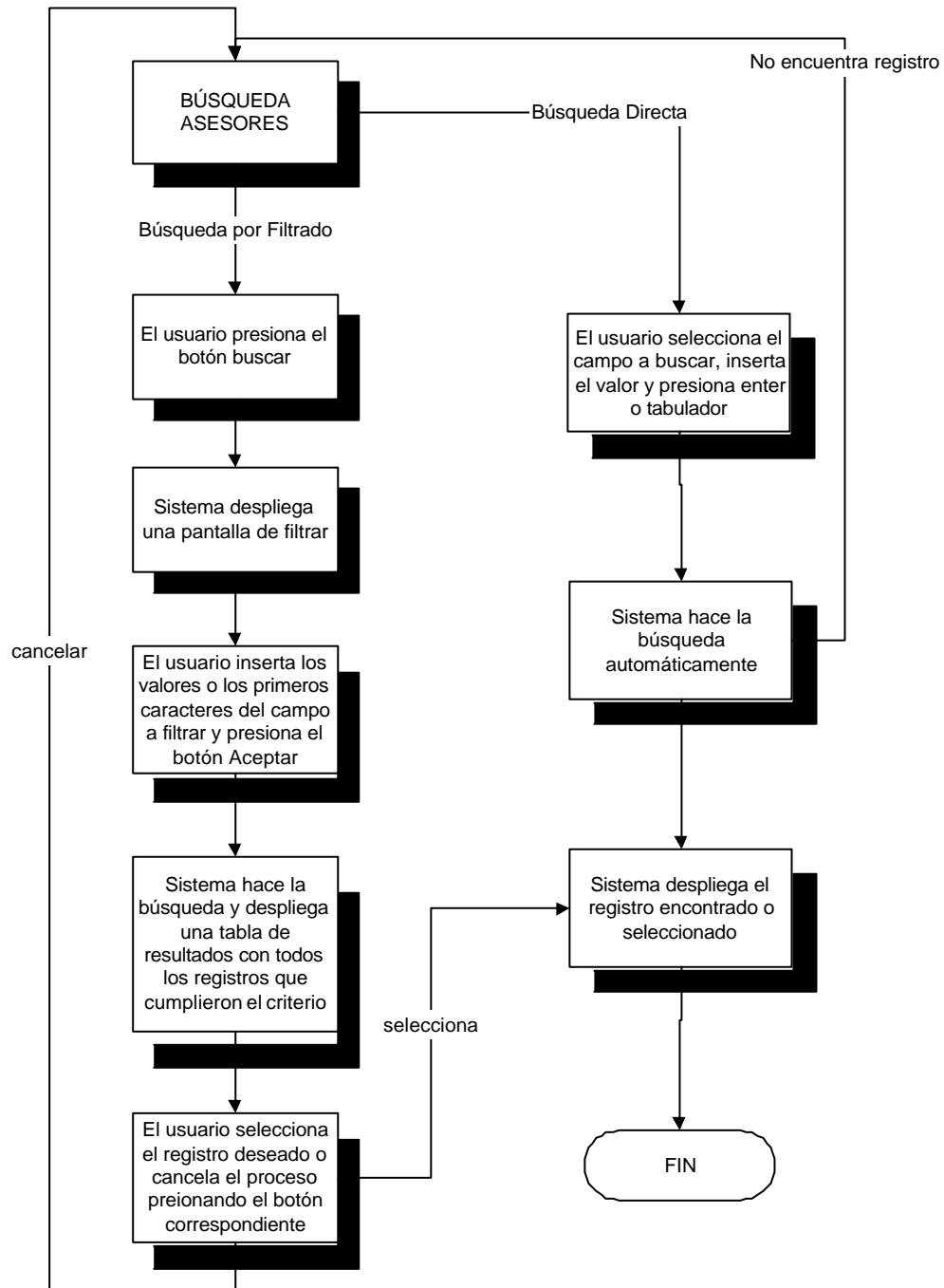


Diagrama 4.14 Flujo para la búsqueda de asesores.

#### 4.4.1.3.5 Seleccionar asesores

Selecciona uno o más asesores para asociarlos a una tesis. También puede borrarse esta selección.

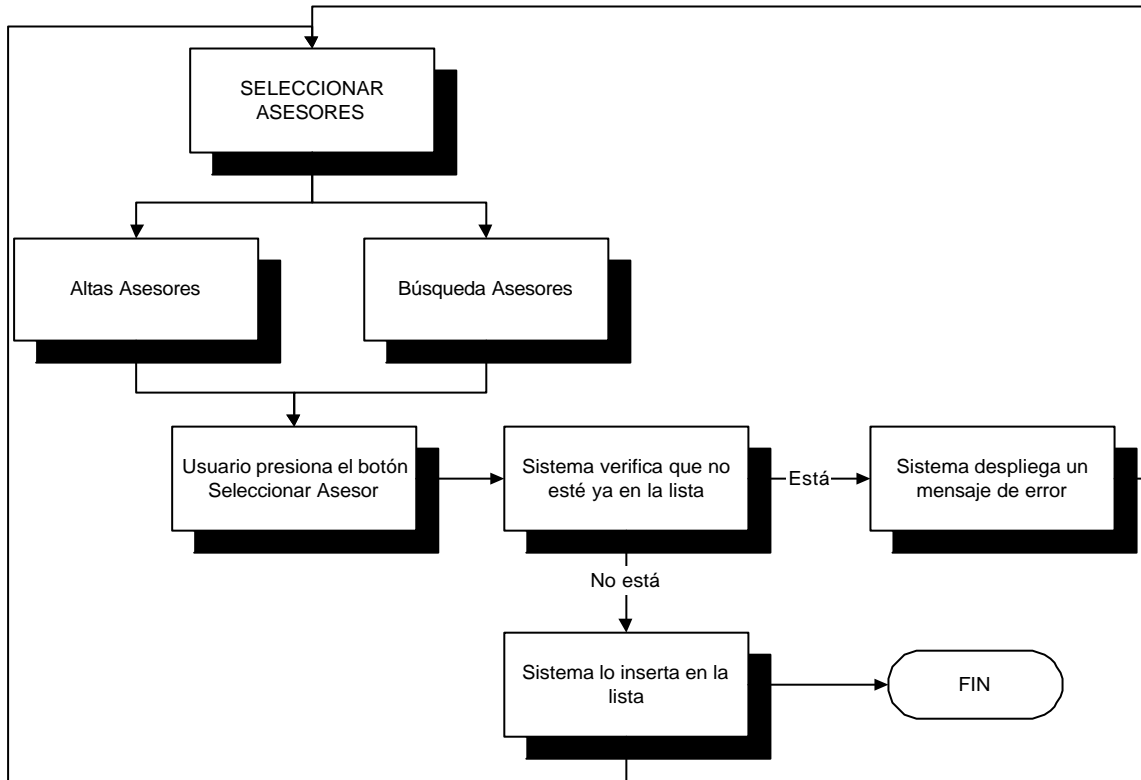


Diagrama 4.15 Flujo para la selección de asesores.

4.4.1.3.6 Borrar selección de asesores

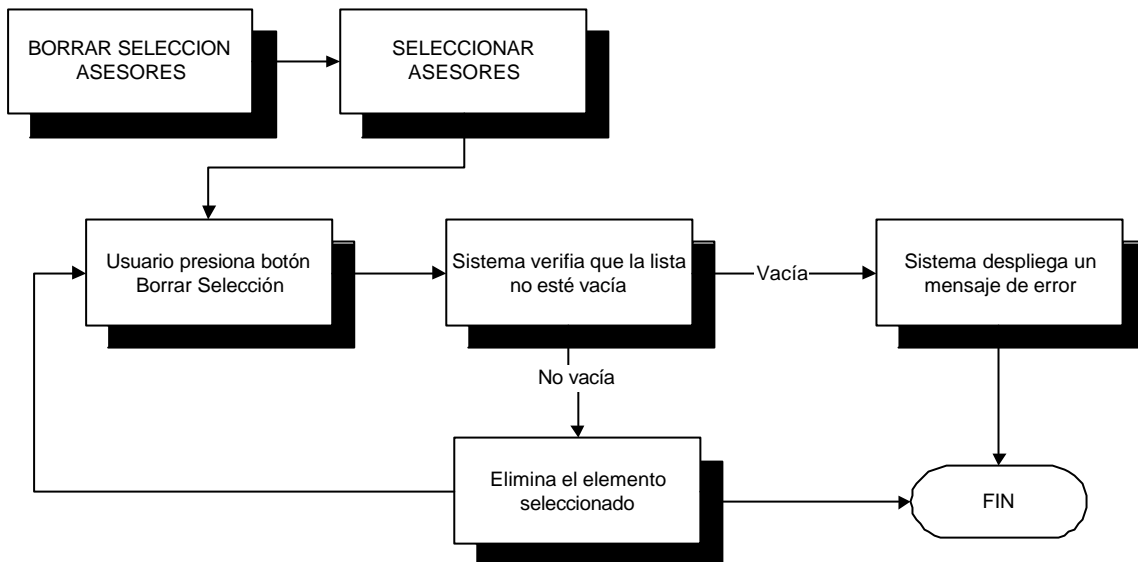
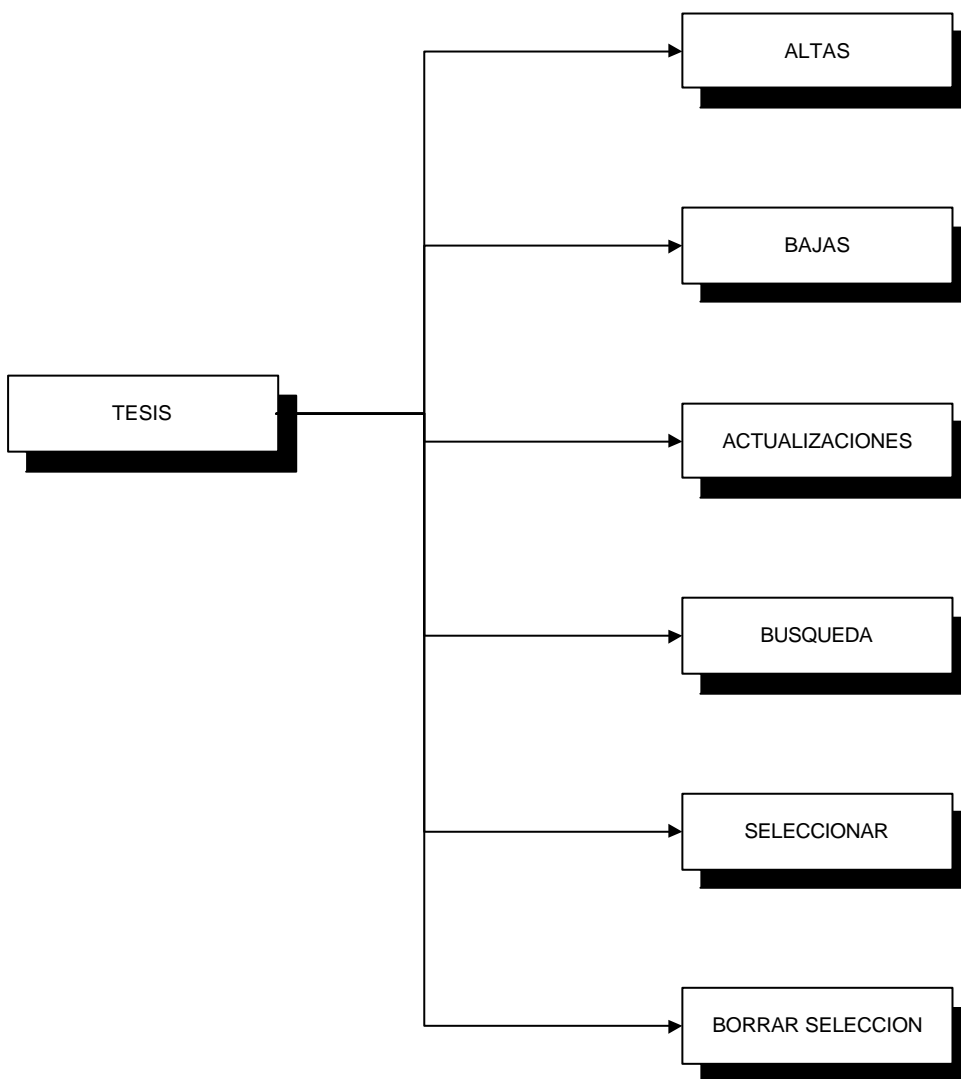


Diagrama 4.16 Flujo para borrar la selección de asesores.



#### 4.4.1.4 Módulo de Tesis

En este módulo se captura toda la información referente a las tesis y sus subtítulos. Aquí termina una primera fase del proceso, ya que establece las relaciones en la base de datos entre alumnos, asesores, tesis y carreras.



**Diagrama 4.17** Flujo del módulo de tesis.

4.4.1.4.1 Altas de tesis

Captura de tesis nuevas y sus subtítulos. Permite al usuario asignar directamente un número de matriz o dejarlo en blanco para que el sistema de el siguiente consecutivo.

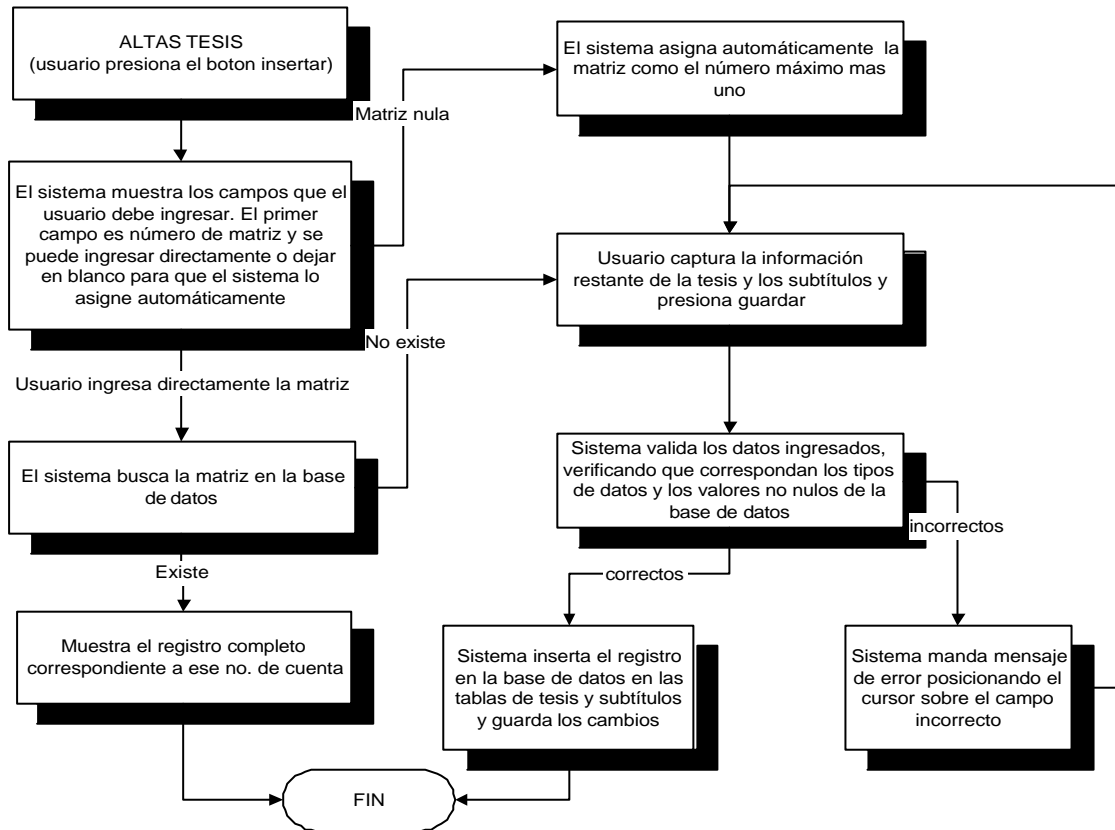


Diagrama 4.18 Flujo para la alta de tesis.

### 3.8.1.4.2 Bajas de tesis

Elimina la tesis y sus subtítulos de la base de datos.

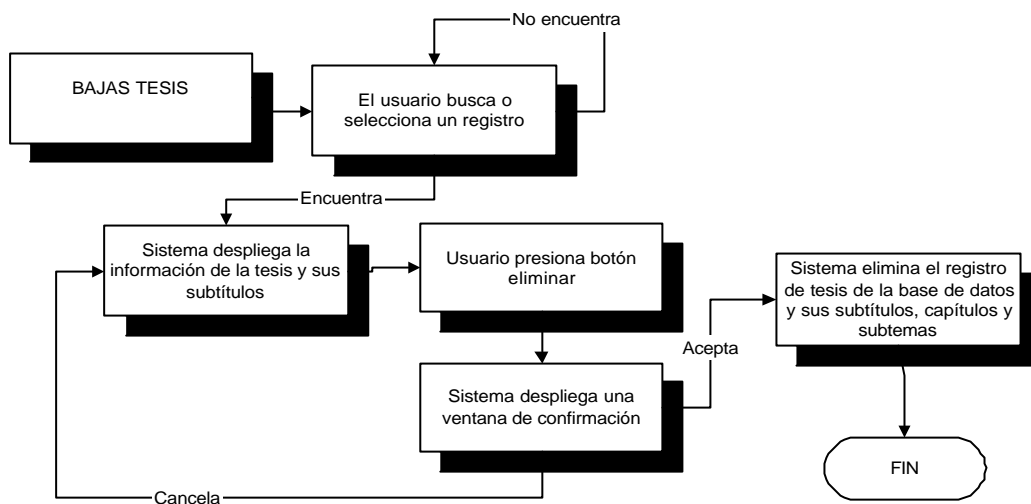


Diagrama 4.19 Flujo para la baja de tesis.

4.4.1.4.3 Actualizaciones de tesis

Permite modificar las tesis capturadas anteriormente pero que todavía no han completado el proceso de registro.

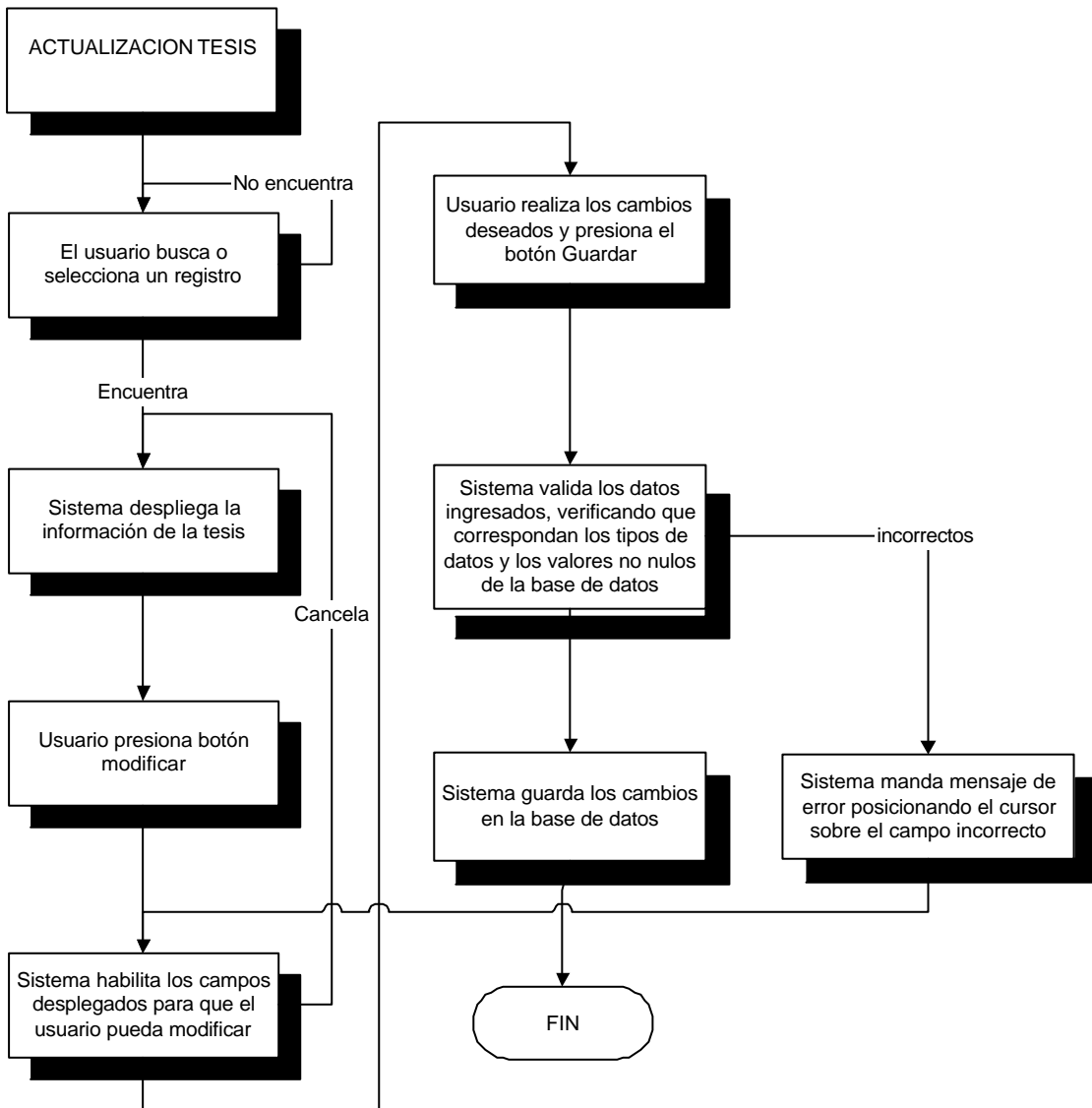


Diagrama 4.20 Flujo para la actualización de tesis.

3.8.1.4.4 Búsqueda de tesis

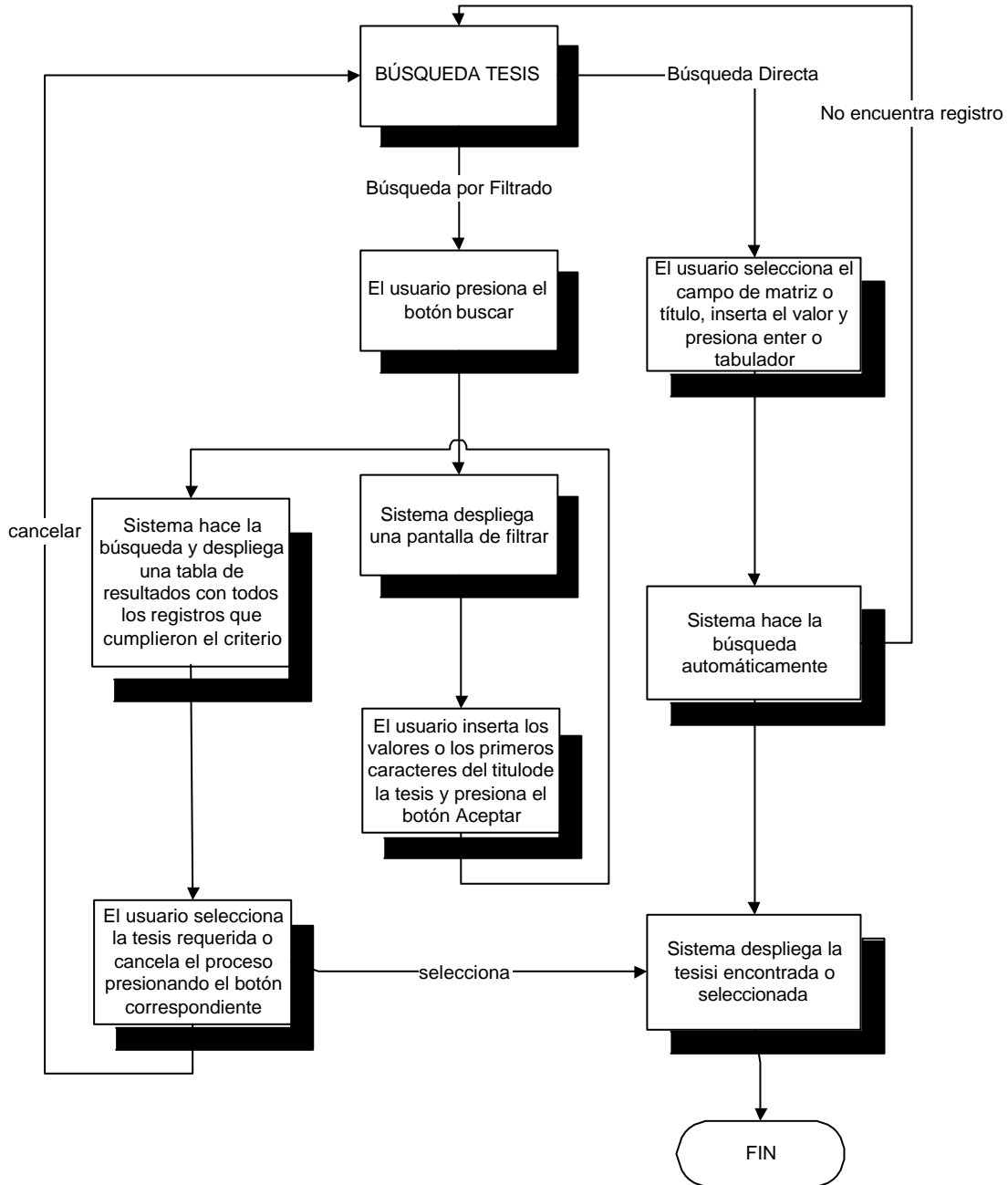


Diagrama 4.21 Flujo para la búsqueda de tesis.

### 3.8.1.4.5 Selección de tesis

Selecciona o borra la selección de la tesis que se va a registrar.

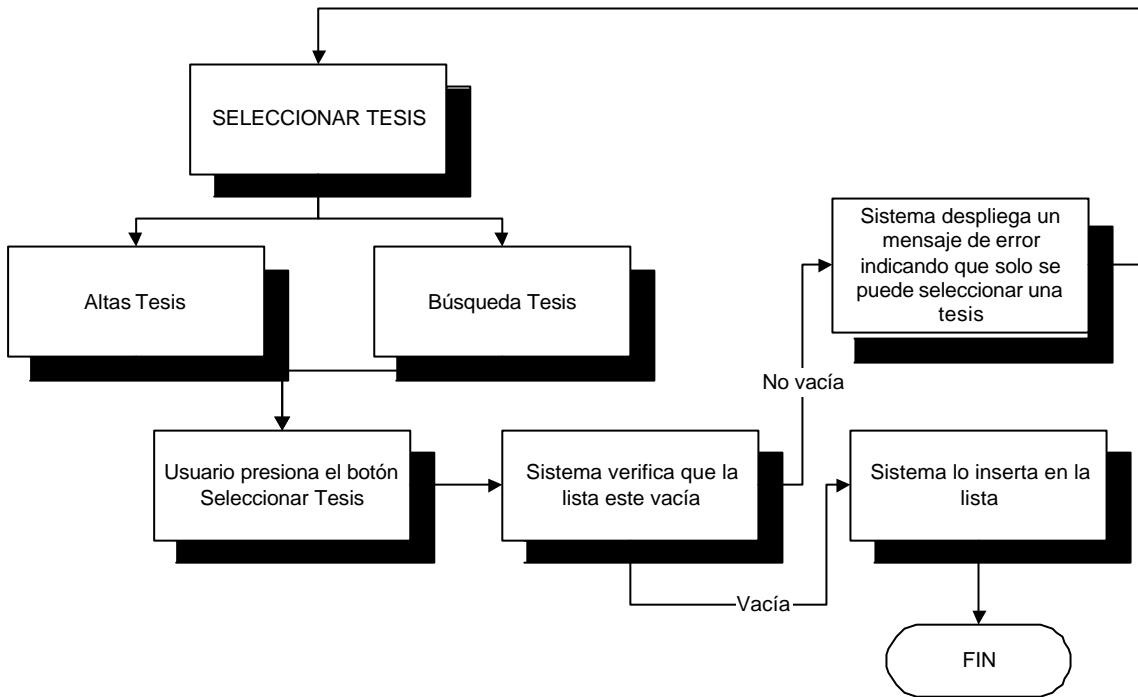


Diagrama 4.22 Flujo para la selección de tesis.

### 3.8.1.4.6 Borrado selección de tesis

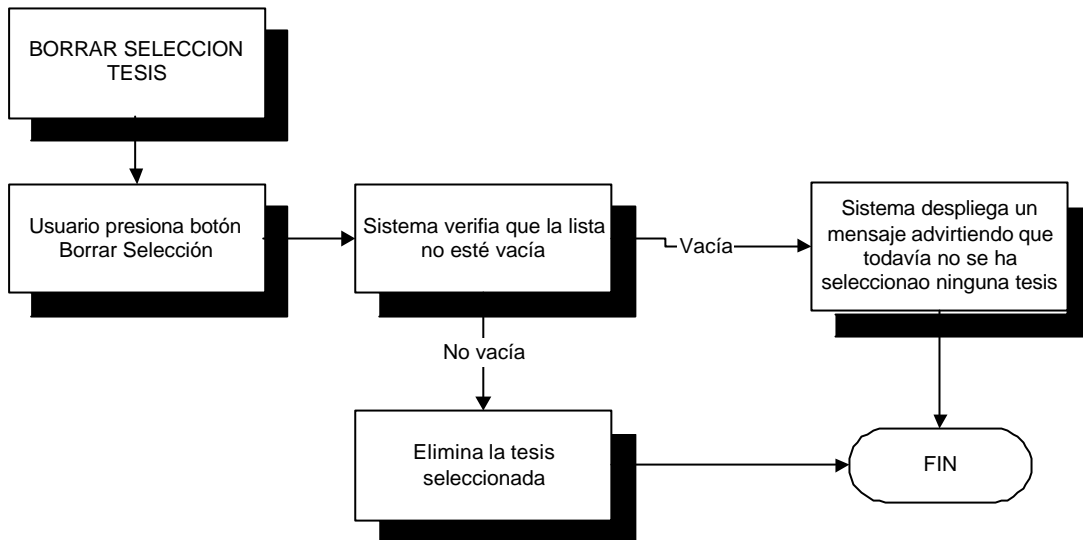
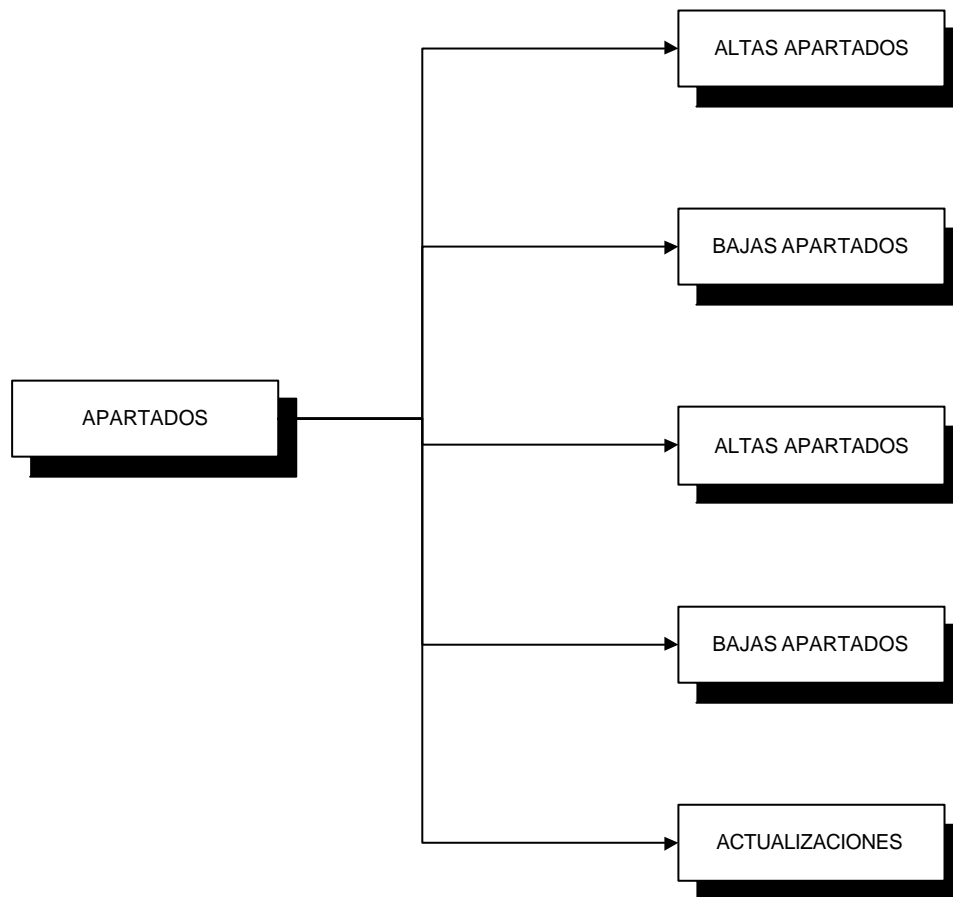


Diagrama 4.23 Flujo para borrar la selección de tesis.

#### 4.4.1.5 Módulo de apartados y subtemas

En este módulo se capturan los apartados asociados a la tesis actual y los subtemas de cada apartado, en el caso de tenerlos. Se puede realizar cualquiera de las operaciones mostradas a continuación.



**Diagrama 4.24** Flujo del módulo de apartados y subtemas.



4.4.1.5.1 Altas de apartados

Ingreso de los apartados correspondientes a una tesis.

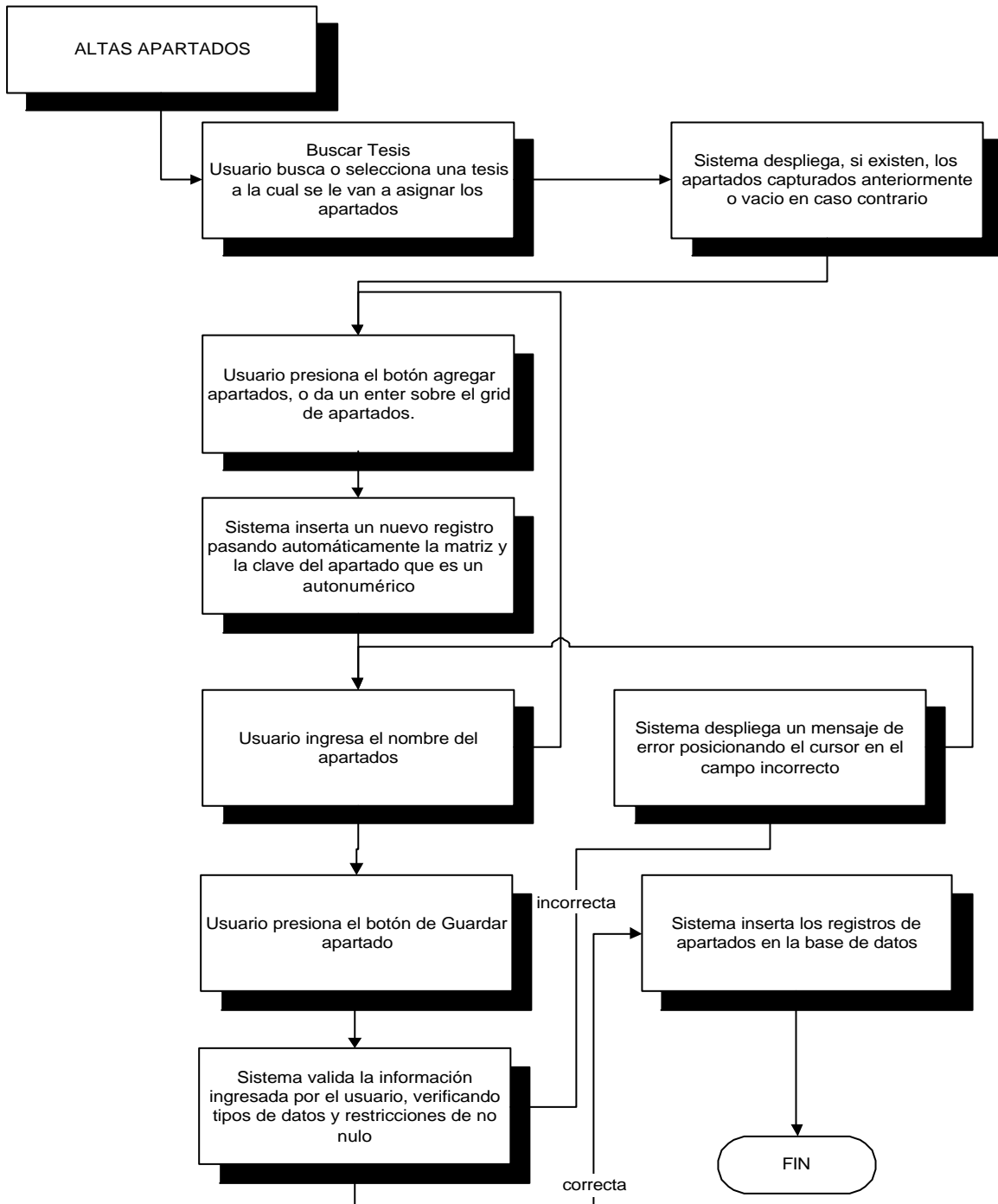


Diagrama 4.25 Flujo para la alta de apartados.

#### 4.4.1.5.2 Bajas apartados

Eliminar apartados de una tesis.

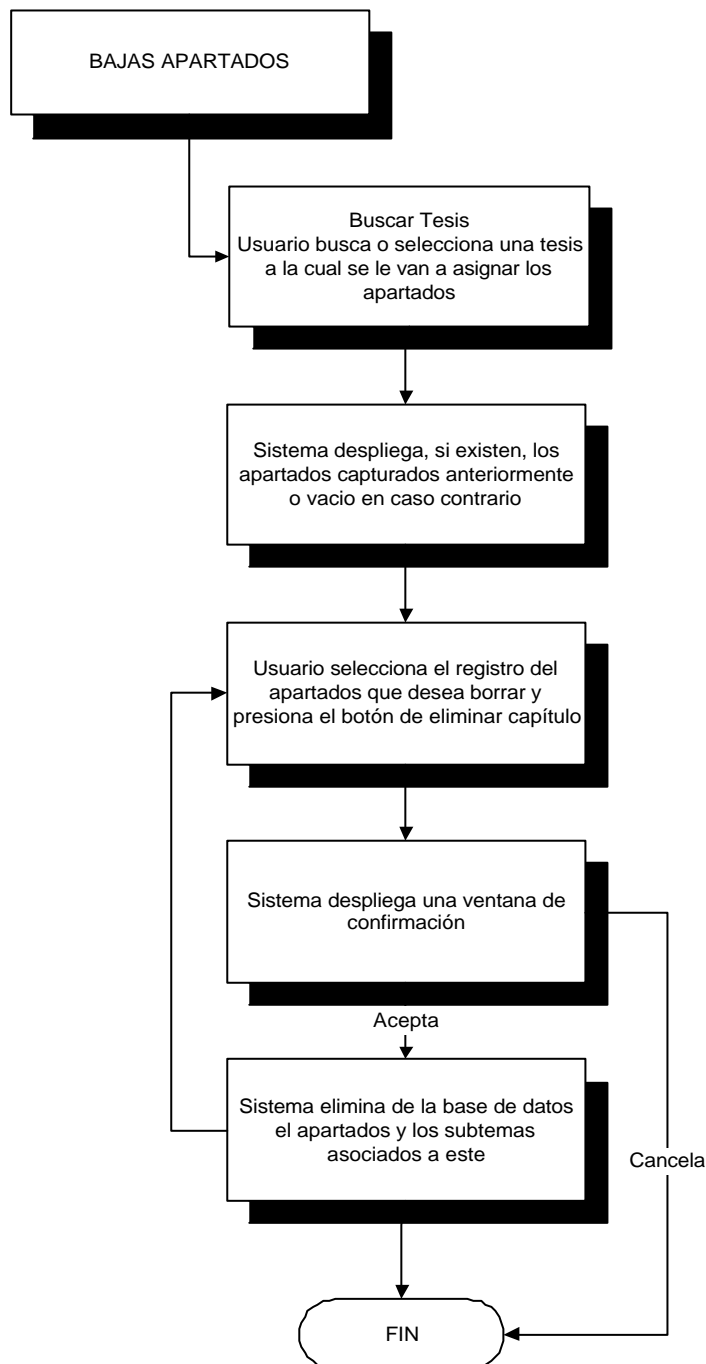


Diagrama 4.26 Flujo para la baja de apartados.

4.4.1.5.3 Altas subtemas

Ingresa un nuevo subtema.

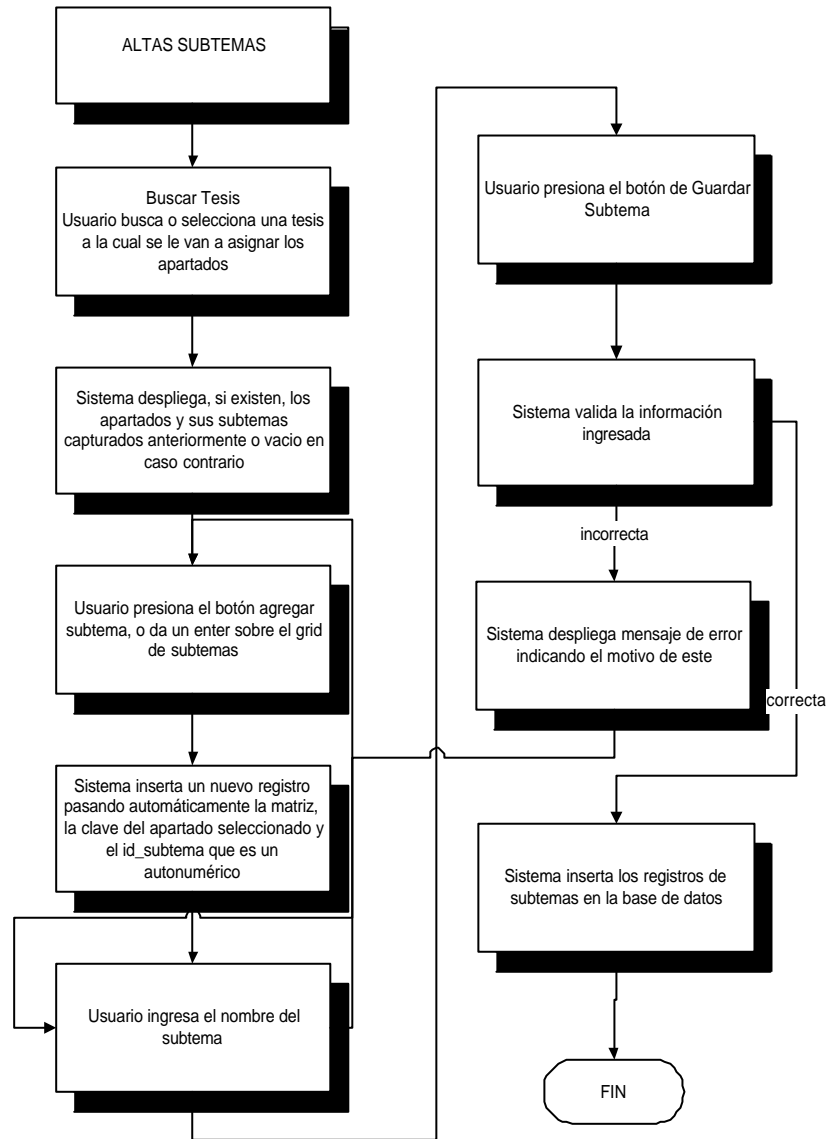


Diagrama 4.27 Flujo para la alta de subtemas.

#### 4.4.1.5.4 Bajas subtemas.

Elimina un subtema existente.

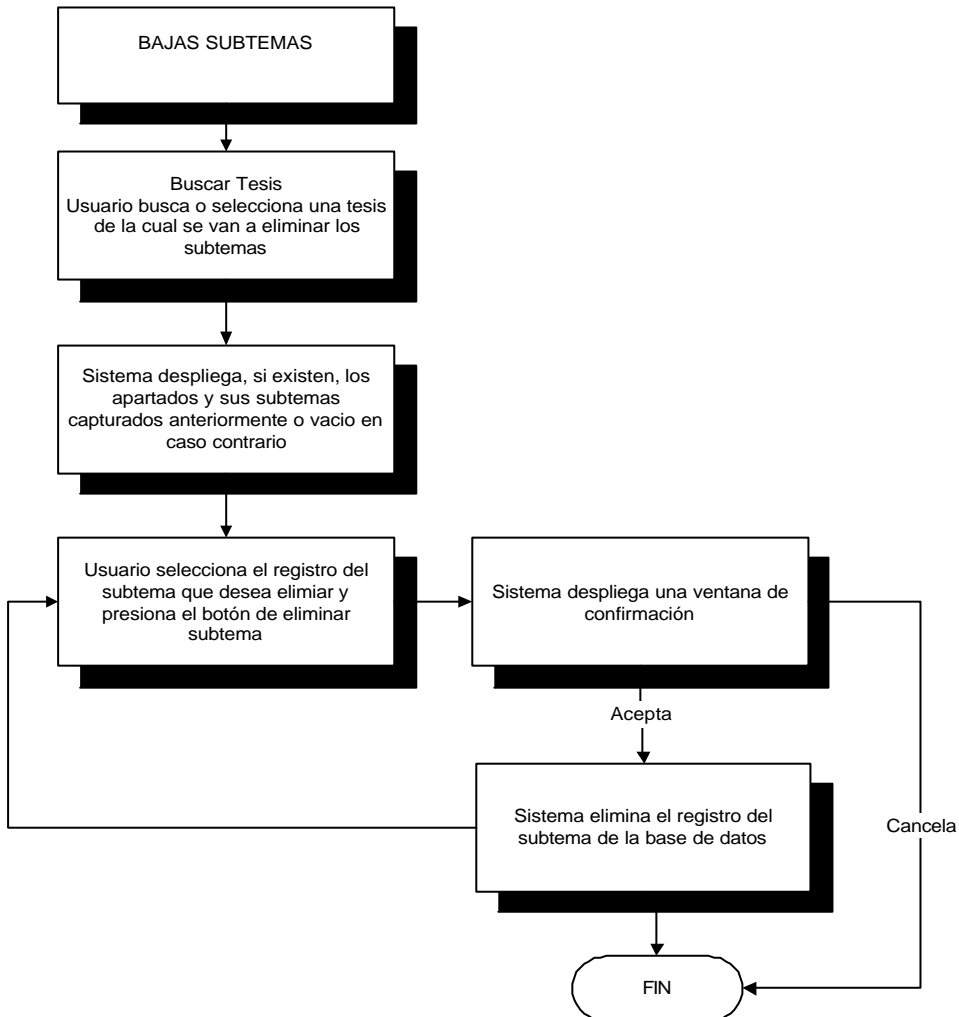
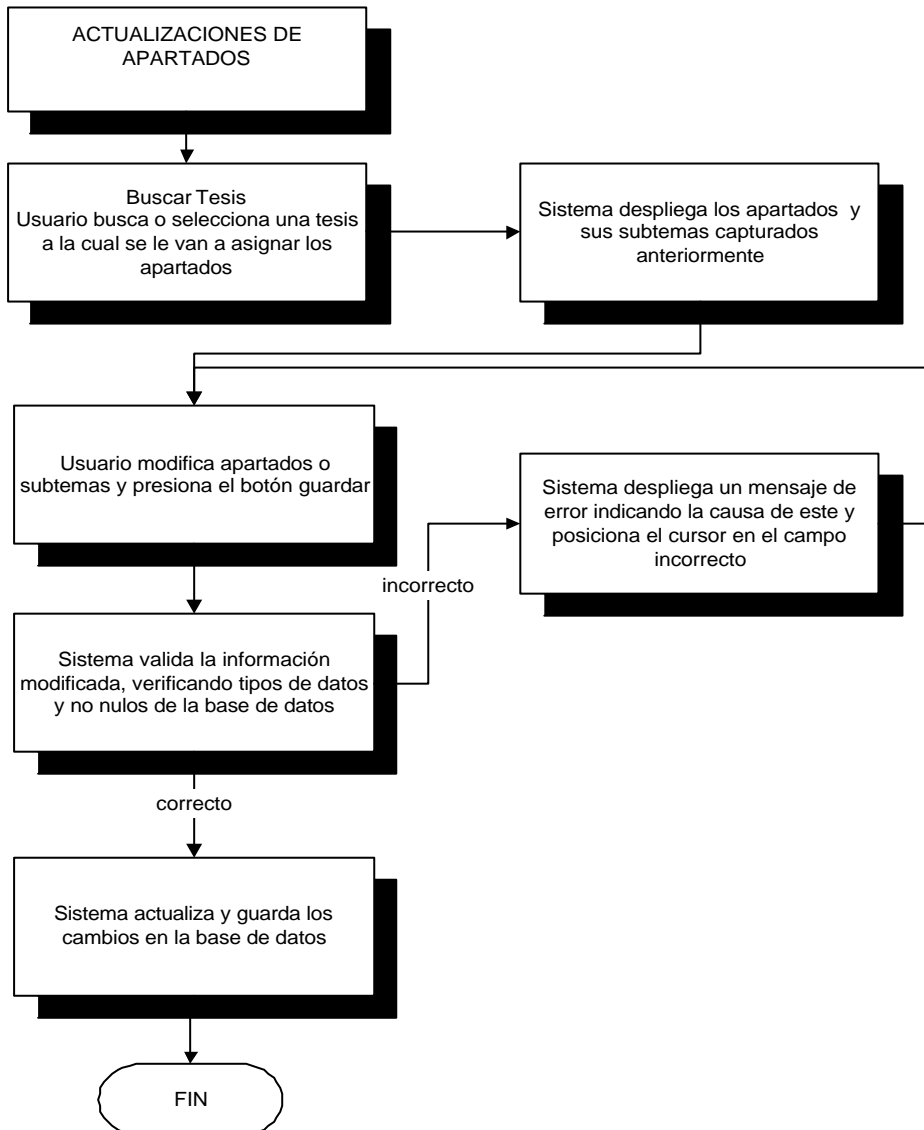


Diagrama 4.28 Flujo para la baja de subtemas.

#### 4.4.1.5.5 Actualizaciones de apartados y subtemas

Modifica apartados o subtemas y guarda los cambios en la base de datos.



**Diagrama 4.29** Flujo para la actualización de apartados y subtemas.

#### 4.4.1.6 Módulo de revisión

Este es uno de los módulos finales del proceso de registro, en el que se genera el índice a partir de la información capturada hasta ese momento. Sirve también para dar una revisión final a los datos completos de la tesis y poder regresarse en caso de ser necesario.

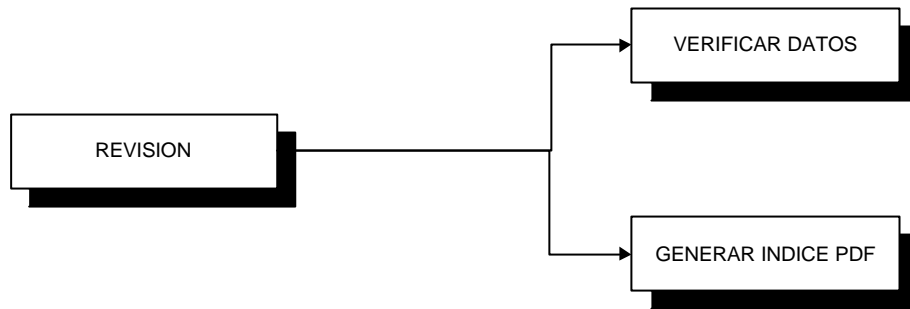


Diagrama 4.30 Flujo del módulo de revisión.

##### 4.4.1.6.1 Verificar datos y generar índice pdf

Se despliega toda la información capturada hasta ese momento. Permite al usuario regresar a los módulos anteriores para corregir algún dato. Una vez revisada la información habilita la opción de generar índice en pdf.



Diagrama 4.31 Flujo para la verificación de datos y generación de Índice PDF.

#### 4.4.1.7 Módulo de archivos

Es utilizado para copiar los apartados en texto completo de la tesis en formato pdf y para encriptarlo.

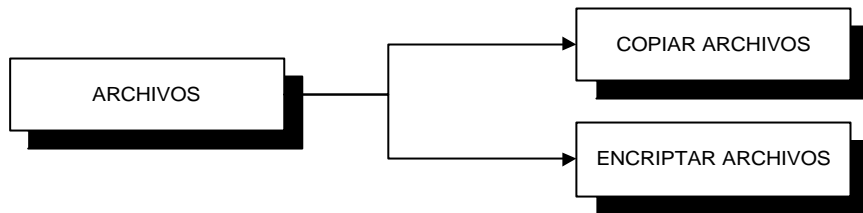


Diagrama 4.32 Flujo del módulo de archivos.

#### 4.4.1.7.1 Copiar y encriptar archivos

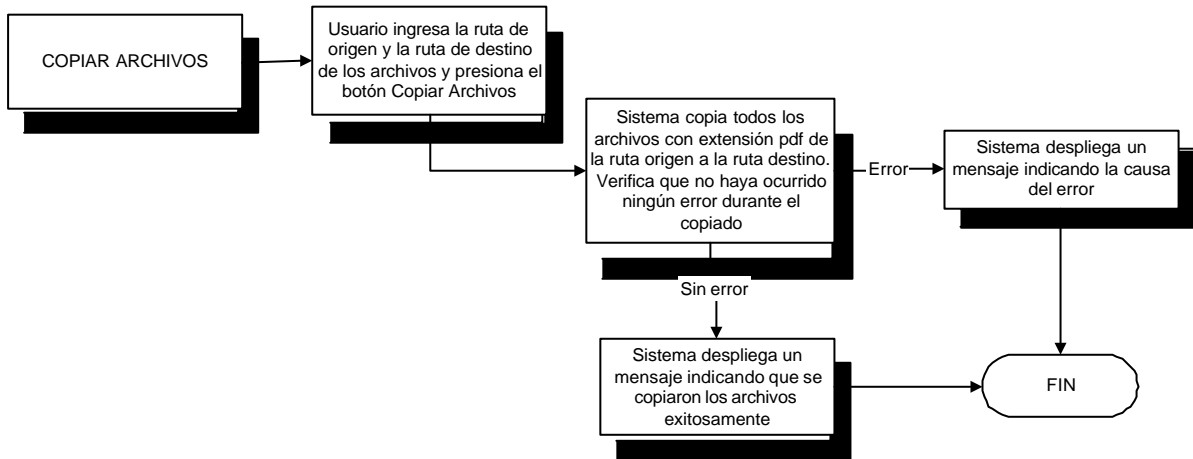


Diagrama 4.33 Flujo para el copiado de archivos.

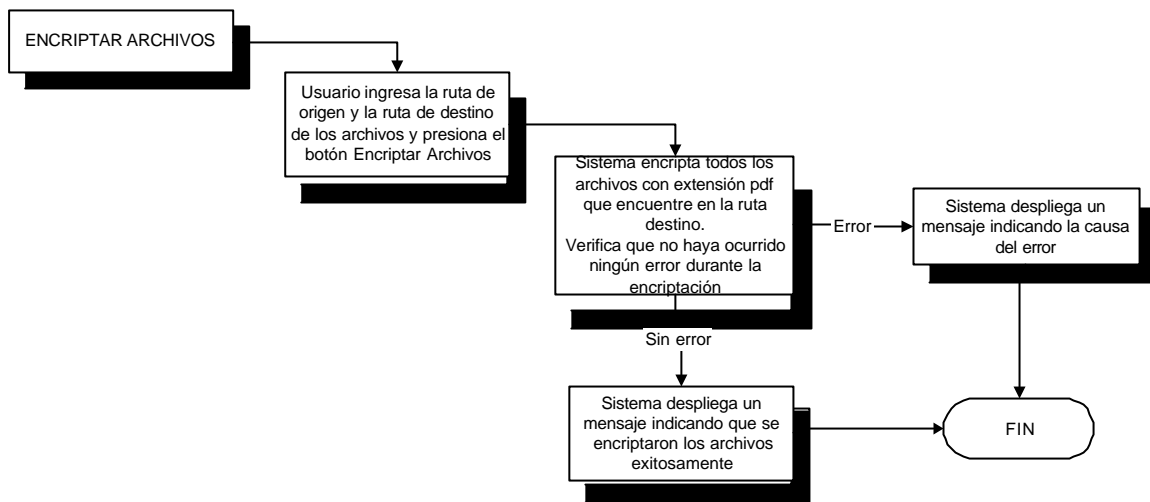


Diagrama 4.34 Flujo para el encriptado de archivos.

#### 4.4.1.8 Utilerías

En utilerías encontramos opciones herramientas disponibles para el administrador del sistema solamente, las cuales ayudan a mantener el funcionamiento de la aplicación cliente de manera óptima y permite hacer ciertas modificaciones de mayor peso dentro del sistema.

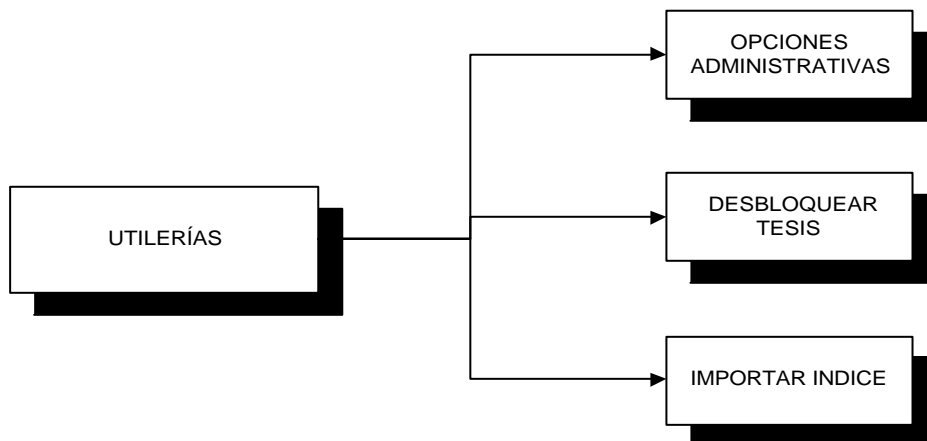


Diagrama 4.35 Flujo del utilerías.



#### 4.4.1.8.1 Importar índice

Genera los capítulos y sus subtemas a partir del índice de la tesis en archivo de texto, y convierte este a pdf.

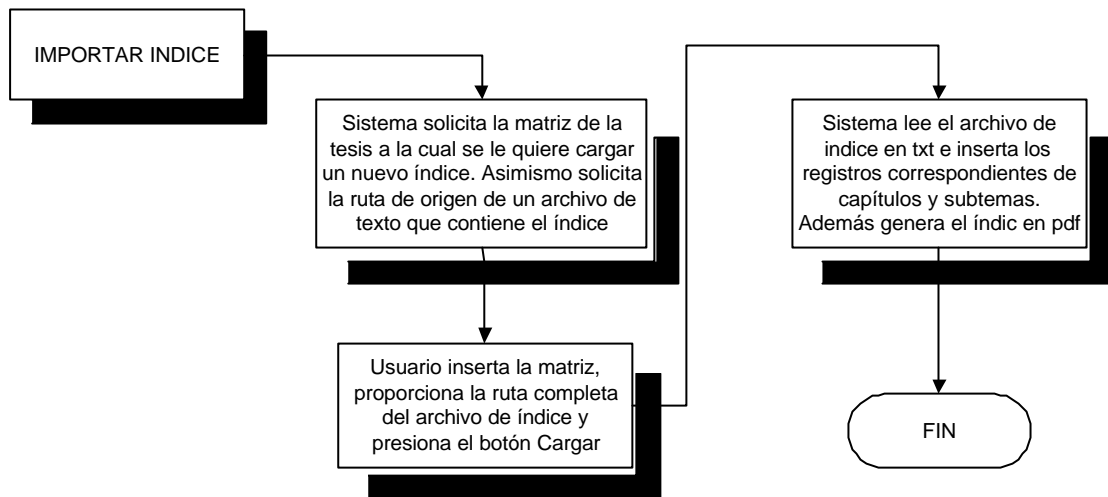


Diagrama 4.36 Flujo para la importación del índice.

#### 4.4.1.8.2 Opciones administrativas

Opciones para configurar la aplicación cliente.

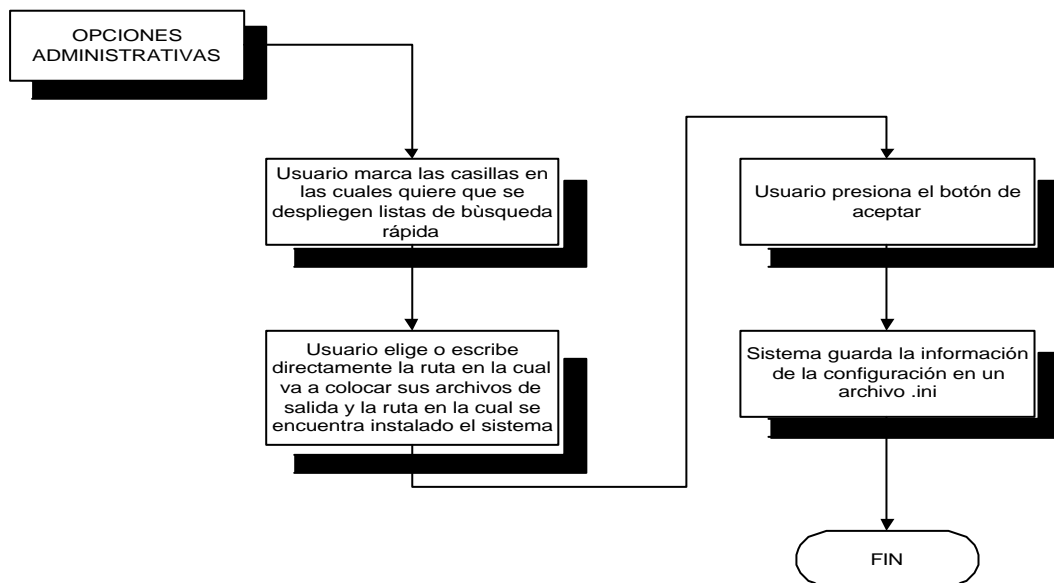


Diagrama 4.37 Flujo de opciones administrativas.

#### 4.4.1.8.3 Desbloquear tesis

Desbloquea una tesis que ya ha completado su registro para poder modificarla.

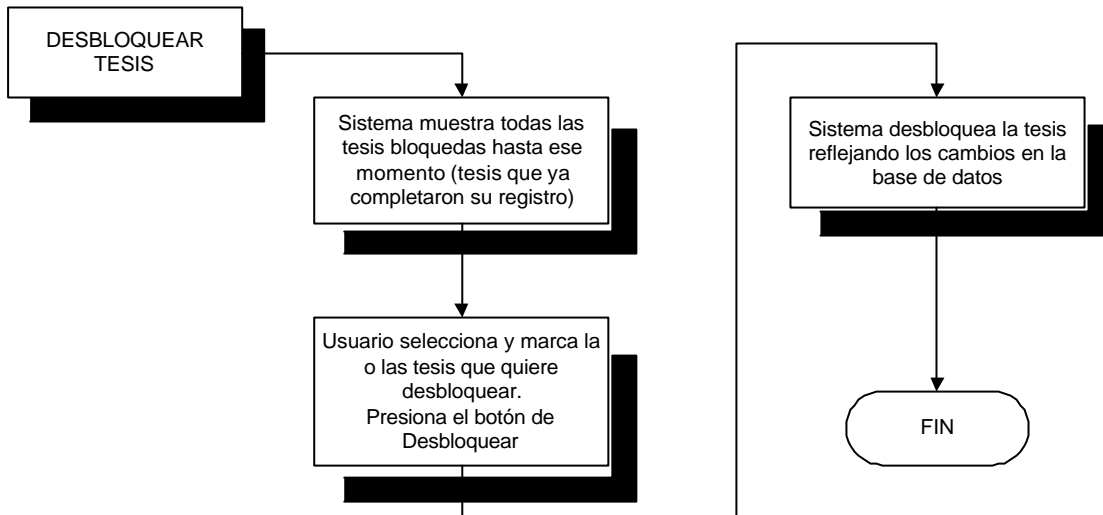
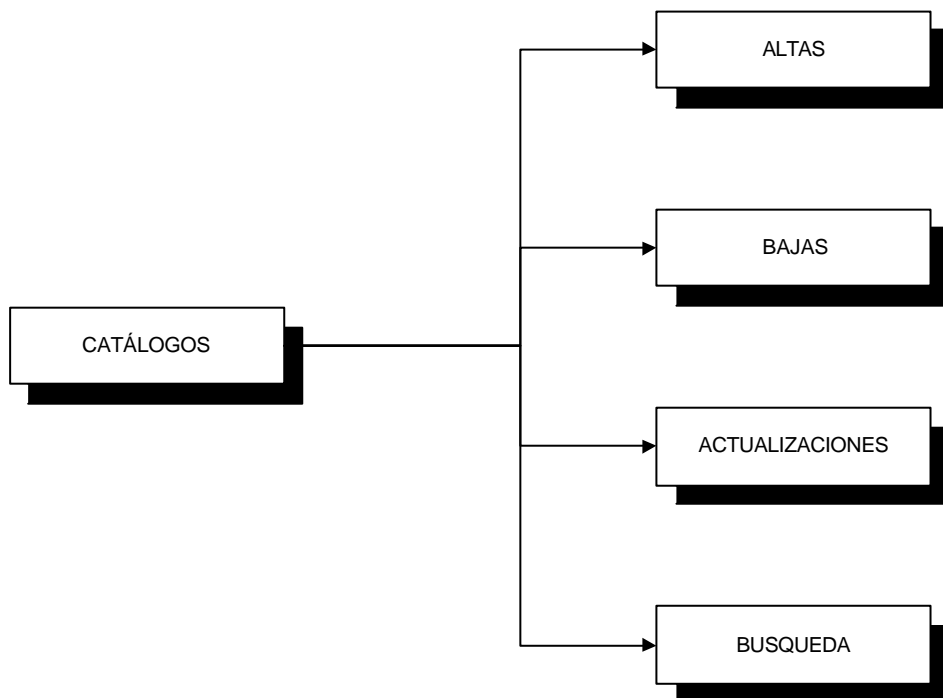


Diagrama 4.38 Flujo para el desbloqueo de tesis.

#### 4.4.1.9 Módulo de catálogos

Mantenimiento, altas, bajas y actualizaciones de los catálogos de la base de datos. Estos son:

- Instituciones.
- Facultades.
- Carreras.
- Grados.
- Usuarios del sistema.
- Perfiles de usuario del sistema.



**Diagrama 4.39** Flujo del módulo de catálogos.

4.4.1.9.1 Altas en catálogos

Ingreso de registros nuevos en cualquier catálogo.

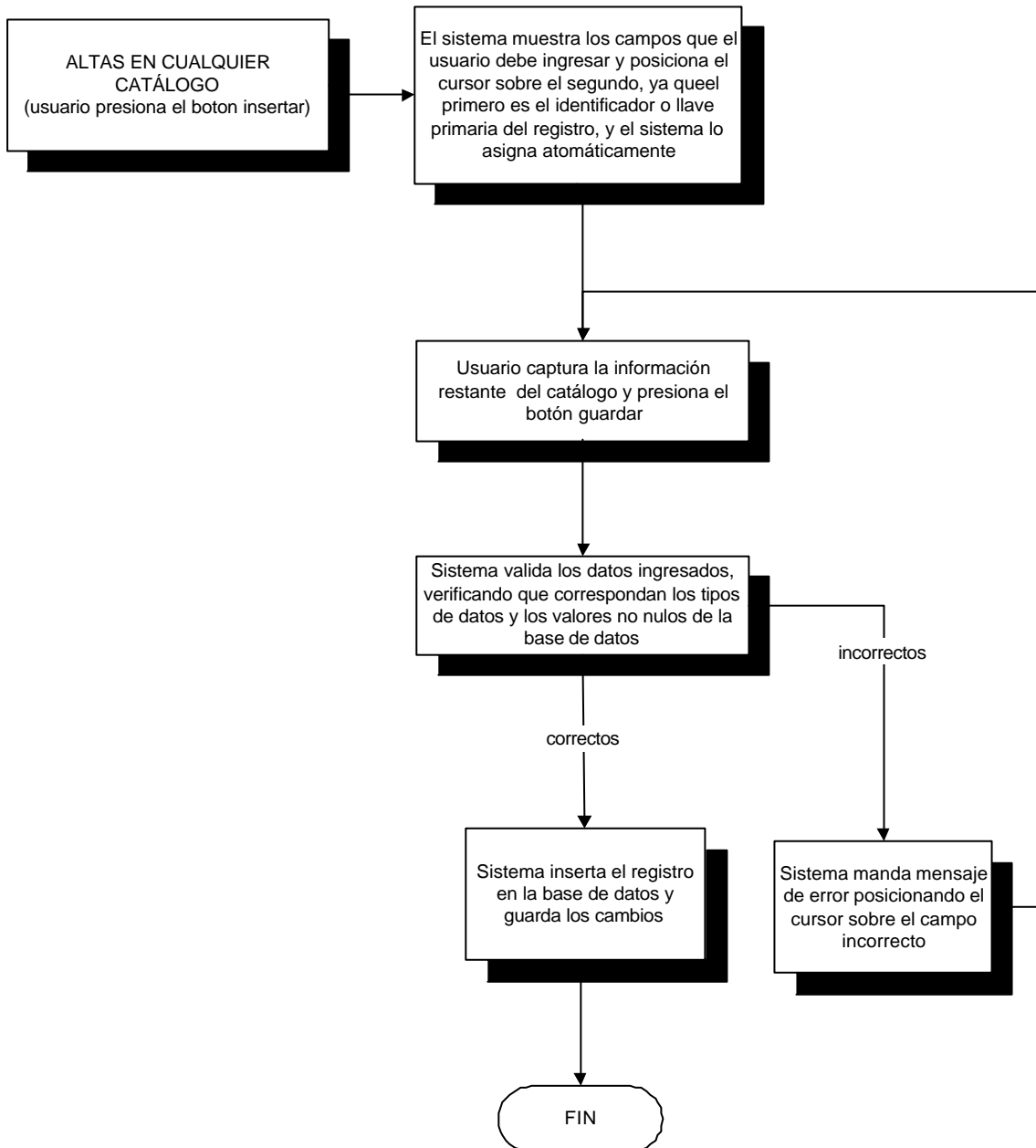


Diagrama 4.40 Flujo para la alta de catálogos.

### 3.8.1.9.2 Bajas en catálogos

Elimina registros de cualquier catálogo.

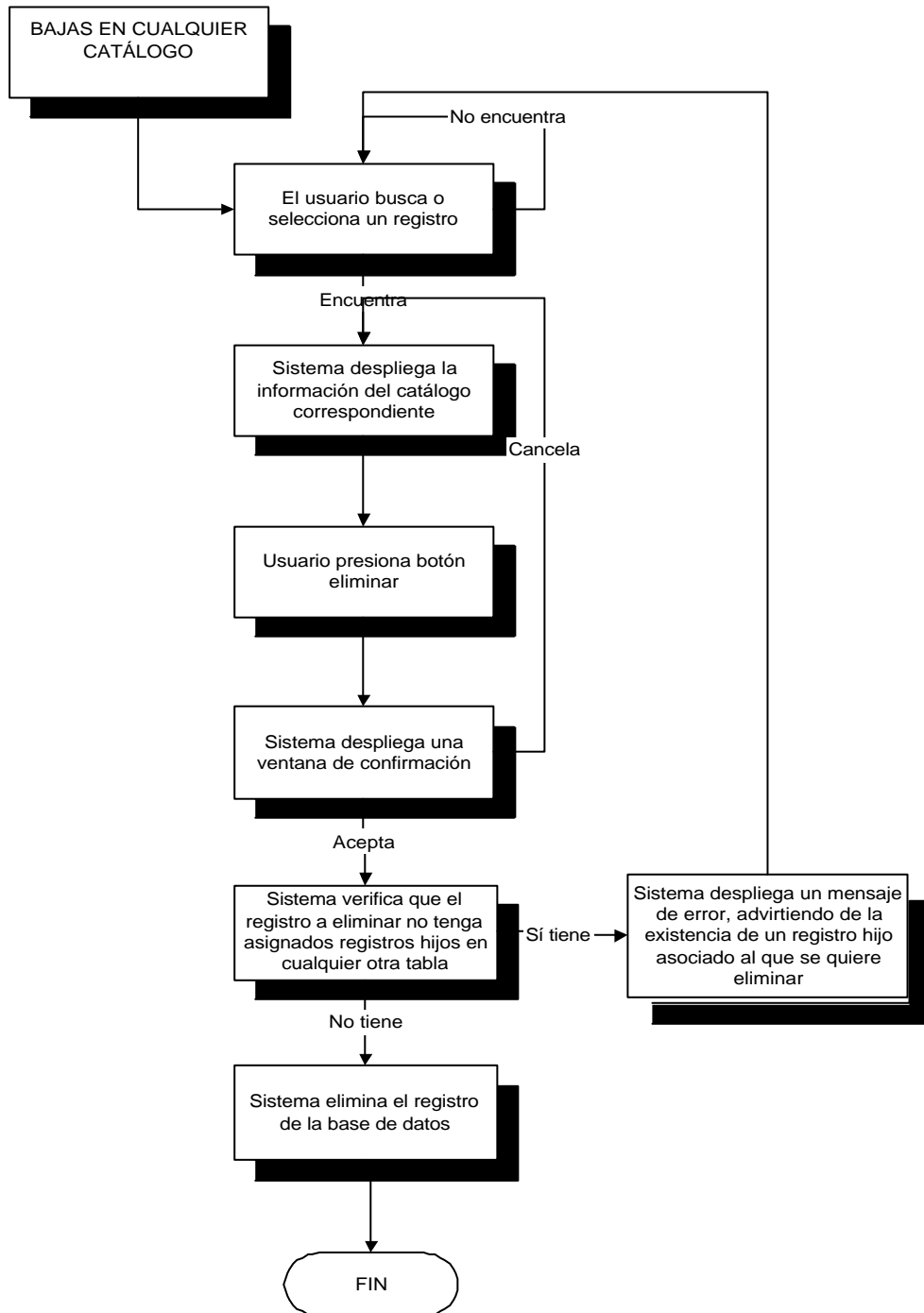


Diagrama 4.41 Flujo para la baja de catálogos.

### 3.8.1.9.3 Búsquedas en catálogos

Realiza búsquedas en cualquier catálogo.

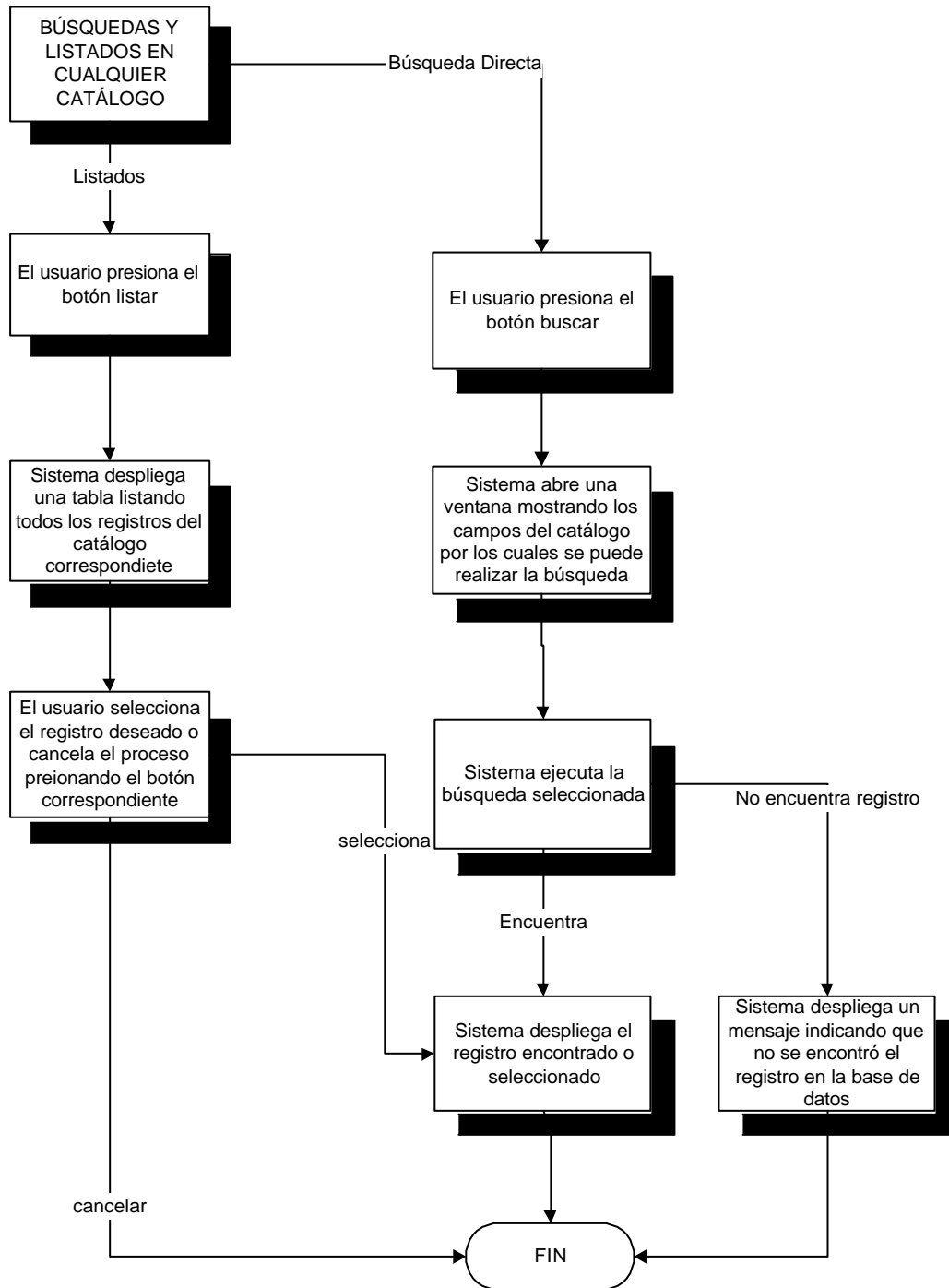


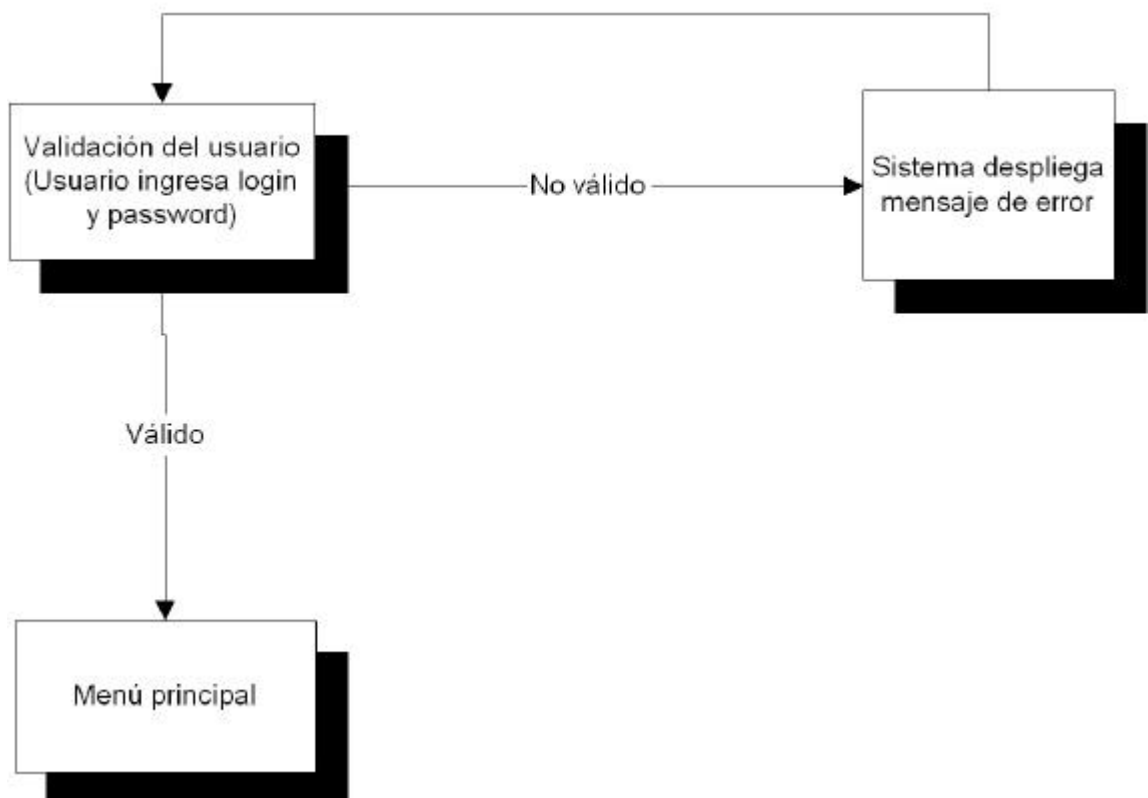
Diagrama 4.42 Flujo para la búsqueda en catálogos.

#### 4.4.2 Diseño de la Aplicación WEB

La estructura de la aplicación WEB esta conformada por tres módulos, que se encuentran en el menú principal y son:

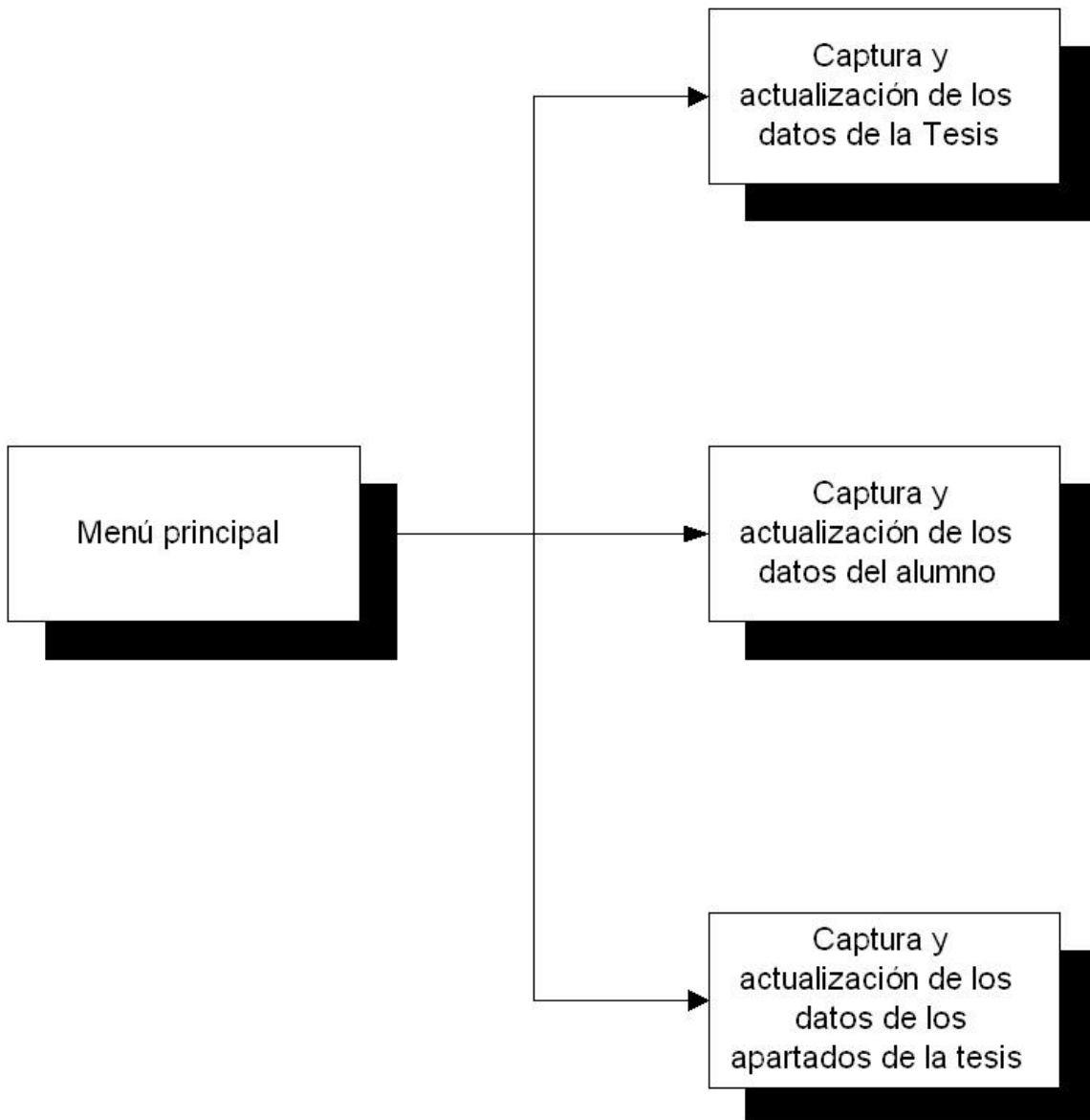
- Tesis
- Alumnos
- Apartados

Primero se mostrara el diagrama para la entrada al sistema:



**Diagrama 4.43** Flujo para la entrada al sistema.

Gráficamente podemos ver los módulos como sigue:



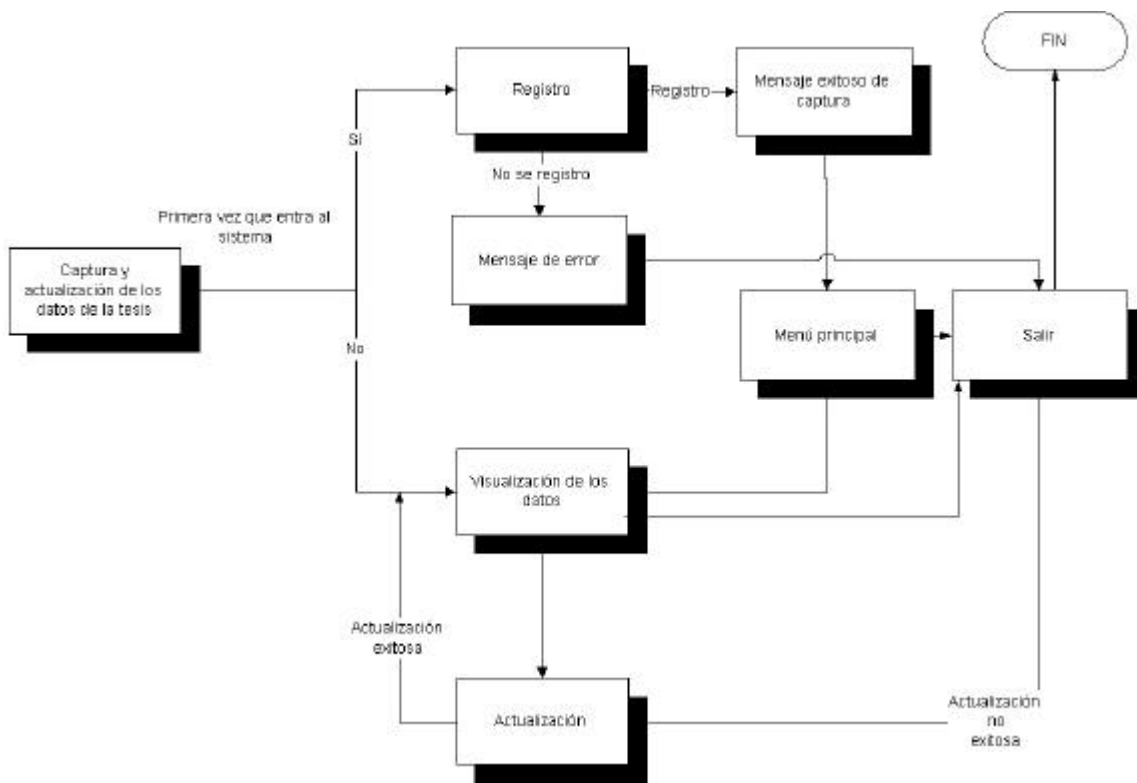
**Diagrama 4.44** Flujo de módulos.



Estos menús secundarios están organizados de manera que se agrupan en ellos diversas opciones, las cuales se describirán en detalle más adelante.

#### 4.4.2.1 Módulo de tesis

En este módulo se capturan los datos de la tesis. Si es la primera vez que se entra a este módulo entonces directamente se entra a la pantalla de captura y desde ahí se introducen los datos necesarios y sino entonces se entra a una pantalla de visualización de los datos pudiendo desde ahí hacer modificaciones.



**Diagrama 4.45** Flujo del módulo de tesis.

#### 4.4.2.2 Módulo de alumno

En este módulo se hace la captura de los datos del alumno. Al igual que en el módulo de tesis si es la primera vez que se entra a esta opción entonces se pasa a la pantalla de captura de los datos del alumno y sino se visualizan los datos teniendo la opción de hacer modificaciones a los datos antes capturados.

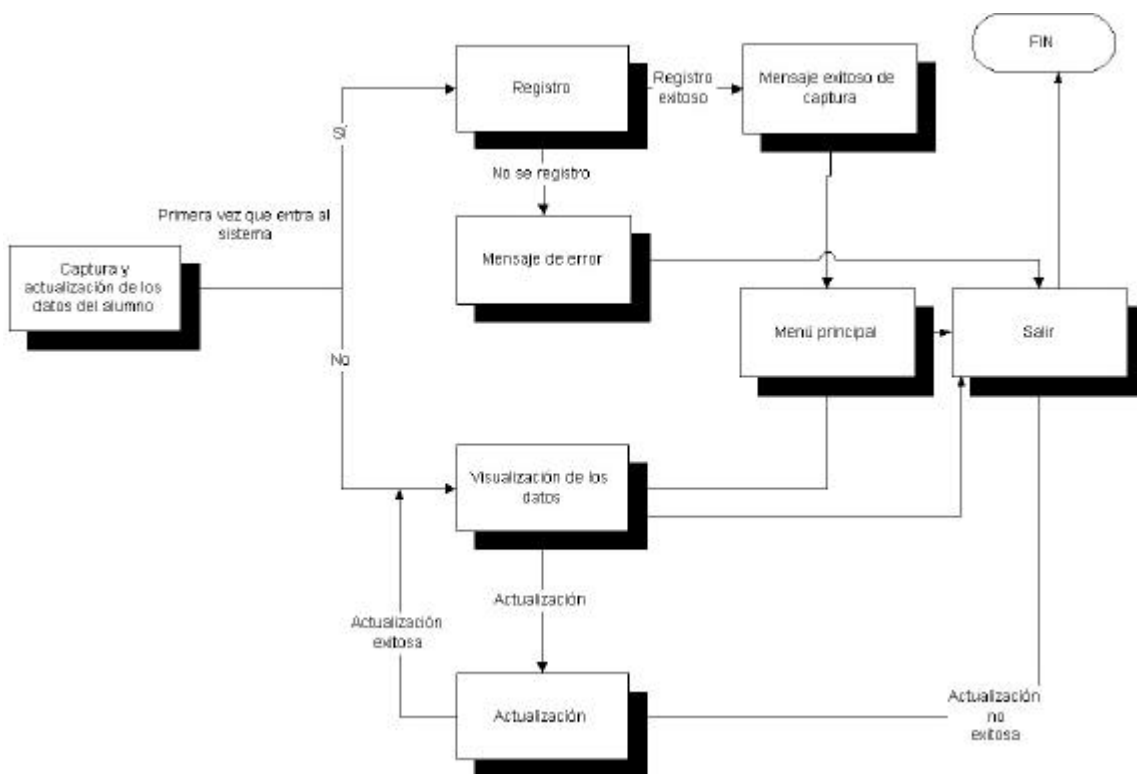
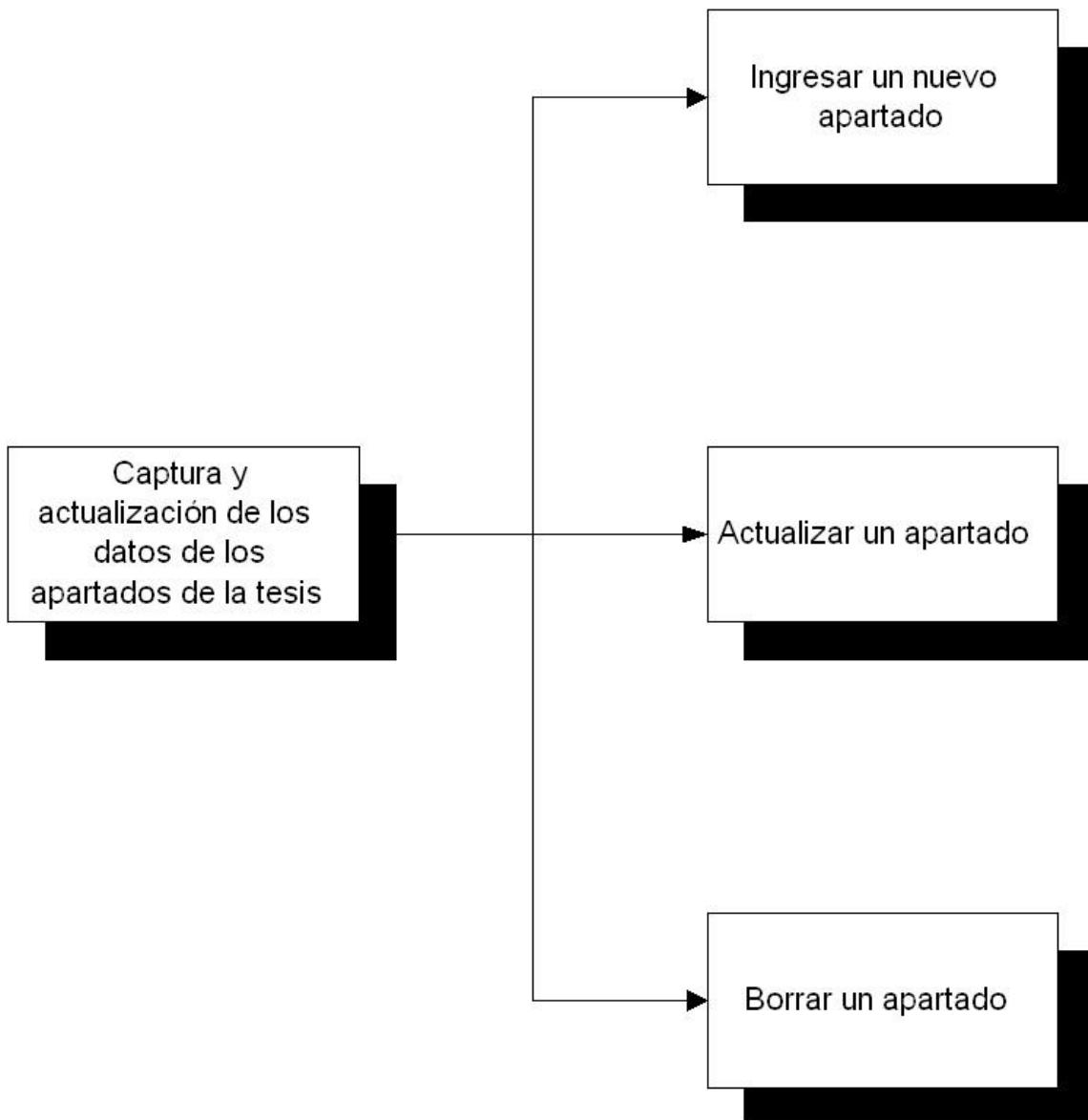


Diagrama 4.46 Flujo del módulo de alumno.

#### 4.4.2.3 Módulo de apartados

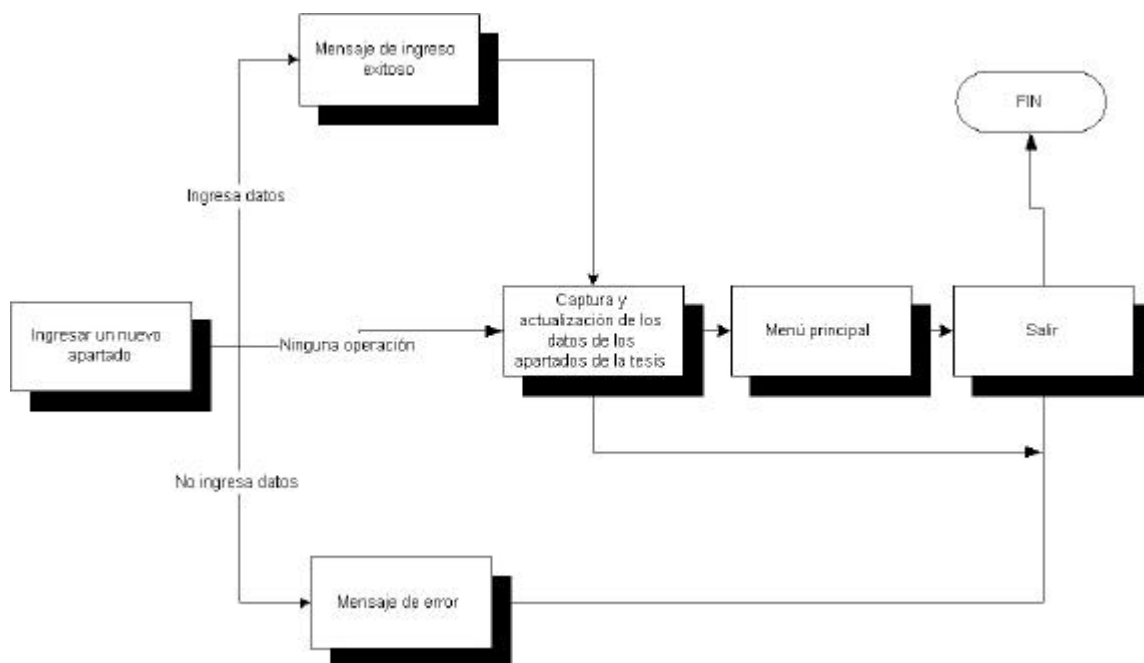
Este módulo presenta tres opciones diferentes que son:



**Diagrama 4.45** Flujo del módulo de apartados.

#### 4.4.2.3.1 Ingreso de un apartado

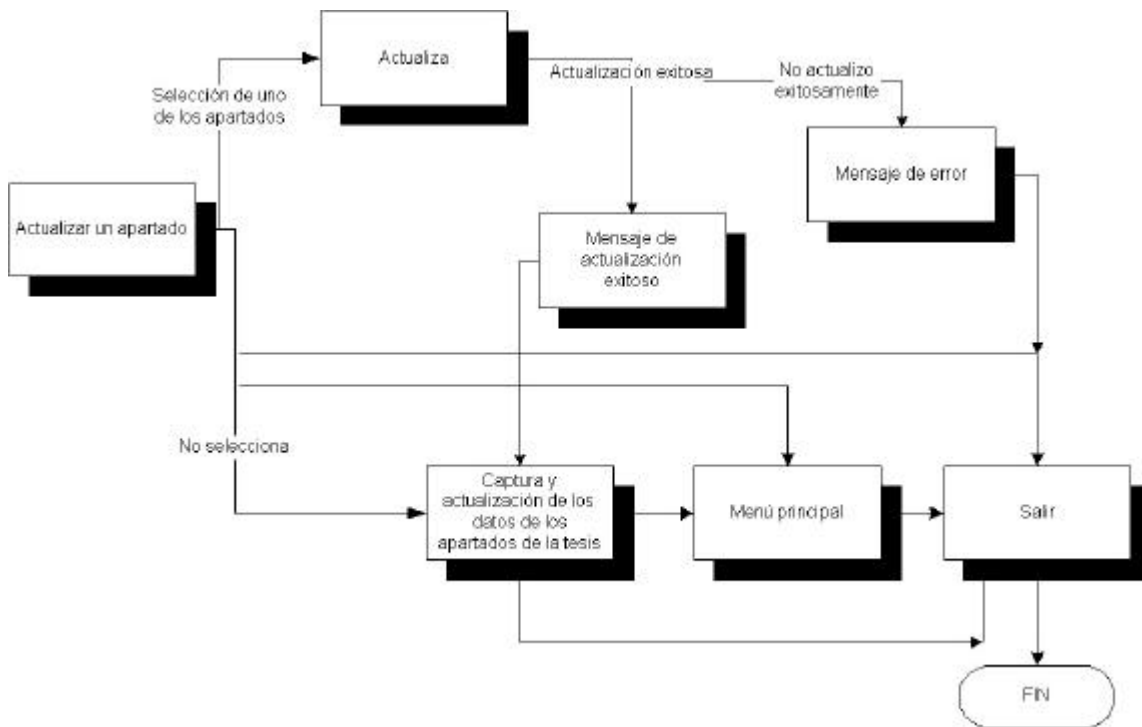
El ingreso de un nuevo apartado se hace como lo muestra el siguiente diagrama:



**Diagrama 4.45** Flujo para el ingreso de un apartado.

#### 4.4.2.3.2 Actualización de un apartado

La actualización de un nuevo apartado lo muestra el siguiente diagrama:



**Diagrama 4.45** Flujo para la actualización de un apartado.

#### 4.4.2.3.3 Borrado de un apartado

El borrado de un apartado es muy parecido a la actualización. El diagrama siguiente muestra la acción de borrado:

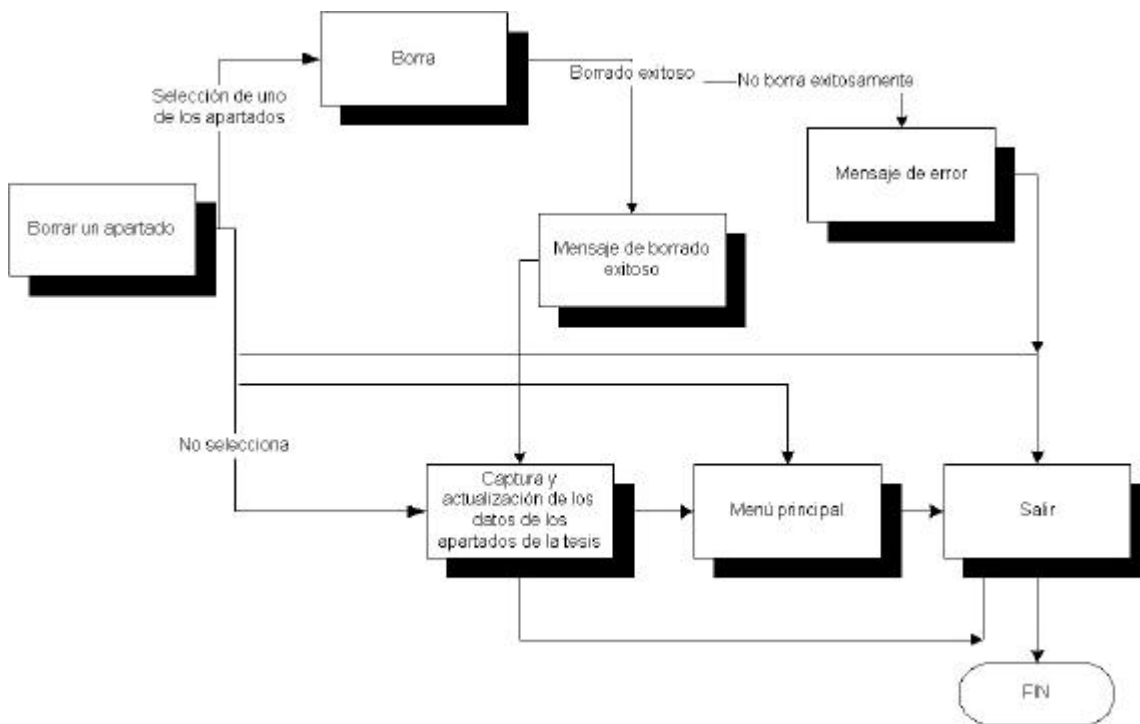


Diagrama 4.45 Flujo para el borrado de un apartado.

### 4.5 Características de las herramientas de software utilizado

#### 4.5.1 Interfaz gráfica

En esta etapa del proceso es necesario elegir el lenguaje o la herramienta de desarrollo de software que se va a utilizar para programar nuestro diseño, por tal motivo, a continuación listamos las características de 3 de los lenguajes más importantes utilizados actualmente.

#### PowerBuilder

La integración de PowerBuilder con ambientes de desarrollo es fácil de hacer. Posee una serie de " painters" , con los que un desarrollador puede construir Ventanas,

Ventanas de Datos, y muchos otros objetos que personalizan una aplicación. Además, PB ofrece utilerías muy poderosas para el diseño y administración de Bases de Datos, de tal forma que permite planearlas, crearlas, manipularlas realizar transferencia de datos entre Bases de Datos por medio del pipeline painter.

Se aprecia la facilidad conque se puede visualmente reconstruir objetos reutilizables que hacen uso de la POO y los conceptos de herencia, polimorfismo, y encapsulamiento. PowerBuilder incorpora la facilidad de obtener y manipular información de las bases de datos mediante el uso del datawindow.

PB nos da la capacidad para que las aplicaciones desarrolladas puedan ser particionadas en un tercer servidor. Esto ayuda a eliminar los tiempos muertos desde el tradicional punto de vista Cliente/Servidor. Con esta distribución, los desarrolladores pueden centralizar los negocios de manera lógica en un solo servidor, lo cual da una gran escalabilidad a las aplicaciones de PB.

Con una gran lista de soporte de terceros, las habilidades orientadas a los objetos se amplían (mecanismos poderosos de bases de datos, soporte a ActiveX y más) lo que hace que PowerBuilder continúe siendo el líder reconocido por grandes empresas y desarrollos visuales del tipo RAD.

PowerBuilder está orientado al 100% para aplicaciones de gestión, con uso intensivo de base de datos. No está pensado, ni diseñado, para aplicaciones de cálculo intensivo, ni para programar drivers, ni rutinas a tiempo real, ni aplicaciones de dibujo... PowerBuilder está pensado para bases de datos, y es ahí donde es fuerte.

### **Visual Basic**

Los desarrolladores pueden crear aplicaciones basadas en los componentes destinados a Internet, Intranet, y los tradicionales ambientes Cliente/Servidor. Visual Basic es muy fácil de aprender y de utilizar, no solamente porque el lenguaje de programación no es OOP (al final viene de BASIC), sino también porque el IDE es simple y cómodo de usar, y los objetos de base de datos que vienen con Visual Basic son más fáciles de utilizar que sus contrapartes de Delphi, aunque claro que no son tan potentes.

Visual Basic hace muchas cosas por el programador. Por ejemplo, los objetos cuentan referencias y esto significa que por ejemplo si creamos un objeto asignándolo a una variable local, el objeto será liberado automáticamente cuando la función o el procedimiento finalice (a menos que lo asignamos a una variable no local). Visual Basic tiene un sistema de administración sofisticado de memoria y utiliza un "colector de basura" (garbage colector) así que es rápido liberando memoria.

En cuanto al acceso a bases de datos, en Visual Basic es más simple. Se usa un solo componente que abre el Recordset, ofrece interfaz visual de navegador y se enlaza con los controles de datos, mientras que en Delphi hay que usar tres componentes para ello. La gran ventaja de los Recordsets de Visual Basic sobre los Datasets de Delphi es que los primeros manejan consultas actualizables automáticamente: se puede tener una consulta de dos tablas y que sea "viva", mientras que en Delphi tenemos que utilizar actualizaciones cacheadas y un componente UpdateSQL con las respectivas consultas SQL para insertar, actualizar o eliminar un registro. Además, cuando uno actúa contra un servidor de base de datos en Delphi tiene que utilizar un componente Transaction, mientras que esto no es necesario en Visual Basic. Visto desde la perspectiva de un programador Visual Basic, el acceso a datos de Delphi resulta ser demasiada molestia cuando se lo compara con los Recordsets y el Control de Datos de Visual Basic que simplifican notoriamente esta cuestión.

Visual Basic es adecuado para simples aplicaciones de interfaz de usuario, típicamente aplicaciones de gestión (facturación, inventarios, nóminas, etc.). Su facilidad de uso lo hace la opción correcta para los programadores principiantes.

## **Delphi**

Delphi es más difícil de aprender que Visual Basic, pero no para quienes están familiarizados por ejemplo con Turbo Pascal, FreePascal, o incluso C/C++. Delphi es más difícil de usar, pero tiene sus ventajas. Por ejemplo, los objetos generalmente no cuentan referencias y esto significa que el programador tiene que encargarse de liberar los objetos no usados creados por un procedimiento o una función cuando ese procedimiento o función termina. La ventaja es que tenemos más libertad en la



manipulación del objeto y podemos liberarlo cuando no lo necesitemos más, sin importar cuántas variables apunten a él. En cuanto a la administración de memoria, Delphi no tiene un colector de basura como Visual Basic, pero tiene su propio sistema de administración de memoria, el que está optimizado para bloques pequeños de datos. El acceso a base de datos puede resultar algo incómodo al principio para quienes están acostumbrados a Visual Basic, pero es muy potente, flexible y extensible.

Delphi es aún mejor para aplicaciones de interfaz de usuario pues la disponibilidad de componentes VCL permite que uno desarrolle interfaces de calidad superior tanto en términos de funcionalidad como de presentación, pero no es tan fácil de utilizar para los principiantes, que son el grupo mayoritario y para quienes prevalece la facilidad sobre la calidad del trabajo, velocidad de ejecución, tamaño del código, uso de los recursos del sistema, tiempo de desarrollo, costos o cualquier otro factor.

Una de las características más importantes es su soporte hacia la orientación a objetos.

Mediante la herencia los programadores de Delphi pueden crear código genérico que evita redundancias. Al no haber redundancia de código se programa mucho más rápido, sobre todo en la etapa de mantenimiento del sistema o cuando se detectan errores. Gracias a la herencia es posible hacer un cambio de código en un solo lugar y que automáticamente se actualicen todas las pantallas y procesos relacionados. También mediante la programación orientada a objetos es posible crear nuevos programas a partir de otros. El polimorfismo permite que se mezcle correctamente el código de un programa ya existente con el de otro nuevo lográndose un comportamiento distinto en cada situación

La programación orientada a objetos llega a su máximo nivel de reutilización mediante la creación de componentes y controles. programarlos en Delphi es muy sencillo y claro. El código se escribe en el mismo `objects pascal` y con las mismas reglas que se utilizan en Delphi para cualquier programa. Una vez que se tiene un componente en Delphi éste se encadena normalmente con el ejecutable, sin embargo, también es posible convertirlos a Activex para ser usados mediante archivos.

Delphi tiene un abanico mucho más amplio de posibilidades, tecnologías, y una filosofía de programación más abierta, lo cual puede ser bueno, o puede no ser necesario si no vamos a aprovecharlo al completo o nuestro departamento sólo se dedica "a hacer selects".

Una vez analizado estos lenguajes decidimos utilizar Power Builder en su Versión 8.0, debido a que este se adapta al tipo de aplicación que estamos desarrollando, la cual está orientada casi en su totalidad a la gestión de la base de datos. Otra cuestión muy importante que influyó en esta decisión, y que nos hizo descartar Delphi, que, cabe mencionar es otra muy buena herramienta, es que en el Departamento de Sistemas de la DGB ya se cuenta con Power Builder 8.0 y con personal capacitado en este lenguaje de programación, por lo que resultó muy conveniente esta elección.

#### **4.5.2 Aplicación WEB**

Al igual que en la interfaz gráfica se tiene que elegir el lenguaje que se va a utilizar para la programación sobre la WEB con bases de datos y el tipo de servidor WEB a utilizar. Por tal motivo se listan algunos de los lenguajes con los cuales podría realizarse esta tarea.

#### **ASP - Active Server Pages**

Active Server Pages es una tecnología que corre del lado del servidor desarrollada por Microsoft.

Con ASP se pueden crear sitios web dinámicos poniendo códigos dentro de las páginas HTML. El código es ejecutado por el servidor antes de que la página sea enviada al navegador. Para esto se puede utilizar tanto Visual Basic como JavaScript.

ASP es un componente estándar en Windows 95,98, 2000 y XP. Puede activarse en cualquier ordenador que corra Windows.

Muchos proveedores de alojamiento web ofrecen ASP, y es una tecnología muy popular, si bien no es la mejor opción, ya que sistemas PHP resultan mucho más fiables y potentes.

### **Java Server Pages (JSP)**

JSP es una tecnología desarrollada por Sun. También funciona del lado del servidor. En cuanto a la forma de programar, resulta muy potente y útil. El tema es que requiere de servidores muy potentes para poder funcionar correctamente. De otro modo resulta insoportablemente lento.

Con JSP se pueden crear páginas web dinámicas, poniendo código dentro de las páginas HTML. El código es ejecutado antes de que la página se envíe al navegador. Como JSP utiliza el sistema Java, esta tecnología no está restringida a una plataforma de servidor determinada. Para migrar de un tipo de servidor a otro, solo hacen falta mínimos ajustes.

### **PHP**

PHP es un revolucionario sistema, que realmente cambió la manera en que los sitios web interactúan con los usuarios. PHP es un sistema que corre en el servidor, permitiendo crear páginas web dinámicas e interactuar con bases de datos de una manera eficiente, sencilla y veloz. Tiene una curva de aprendizaje muy rápida, y al ser un producto de desarrollo por fuente abierta, posee una comunidad de soporte impresionante, que colabora con su desarrollo a diario. Como en todos los sistemas que corren en el servidor, el código es ejecutado por el servidor web antes de servir la página al cliente (navegador).

### **Servidor WEB**

Un servidor web es un programa que responde a las peticiones de páginas de un browser o navegador, y devuelve a su vez la página al mismo navegador a través de la Internet.

Los servidores web permiten ejecutar scripts los cuales son interpretados y regresados como una página en html, esto permite crear páginas web dinámicas. A continuación se presenta una lista de servidores web y que scripts pueden correr.

**Internet Information server 5.0** - ASP, ASP.NET, CGI, Python, PHP

**Apache** - PHP, CGI/Perl, Python

**Tomcat** - CGI/Perl, JSP,Servlets, JavaBeans

**iPlanet** - JSP, Servlets, Enterprise JavaBeans

**Chilisoft** - ASP

**Go webserver** - LotusScript

---

---

## **CAPÍTULO 5**

---

---

## **DESARROLLO DEL SISTEMA**

### **5.1 Desarrollo de la base de datos**

Como ya se mencionó, Oracle es el dbms que elegimos para alojar nuestra base de datos. Para crear físicamente esta base, necesitamos considerar las estructuras internas de Oracle y cómo es que estas se relacionan.

Las bases de datos de Oracle almacenan los datos en archivos. Internamente, existen estructuras en la base de datos que realizan una asignación lógica entre los datos y los archivos, lo que permite almacenar por separado diferentes tipos de datos. Estas divisiones lógicas se denominan espacios de tablas.

En su forma mas elemental, una base de datos de Oracle consta de:

- Uno o más archivos de datos. (Forman los espacios de tablas)
- Uno o más archivos de control. (Conservan la coherencia interna y guían las operaciones de recuperaciones)
- Dos o más registros de rehacer en línea. (Registran todas las transacciones realizadas en la base de datos)

Internamente, la base de datos contiene:

- Múltiples usuarios/esquemas.
- Uno o más segmentos de anulación. (Mantienen la coherencia entre varios usuarios y permiten anular transacciones)
- Uno o más espacios de tablas. (Realizan una asignación lógica entre los datos y los archivos)
- Tablas del diccionario de datos.
- Objetos de usuario (tablas, índices, vistas, etc.)

El servidor que accede a dicha base de datos, como mínimo, consta de:

- Un SGA (Facilita la transferencia de información entre usuarios. Además, almacena la información estructural sobre la base de datos que se consulta con más frecuencia)
  - Caché del búfer de bloques de datos. (Guarda los bloques de la bd que se leen desde los segmentos, tales como tablas, índices y agrupaciones)

- Caché del bufer del registro de rehacer. (Optimiza la escritura de los registro de rehacer en línea)
- Área SQL compartida. (Almacena la caché del diccionario de datos y de biblioteca. Por lo que permite que se comparta información estructural y sobre los datos, así como las instrucciones SQL más ejecutadas )
- El proceso en segundo plano SMON. (System Monitor. Monitor del Sistema)
- El proceso en segundo plano PMON.
- El proceso en segundo plano DBWR. (Database Writer, escritor de la base de datos)
- El proceso en segundo plano LGWR. (Log Writer, escritor del registro)
- Procesos de usuario, con sus áreas PGA asociadas.

Distribución lógica de los segmentos en una base de datos óptima.

<b>Espacio de Tablas</b>	<b>Utilización</b>
SYSTEM	Diccionario de datos
DATA	Tablas de operación normal
DATA_2	Tablas estáticas utilizadas durante la operación normal
INDEXES	Índices de las tablas de operación normal
INDEXES_2	Índices de las tablas estáticas
RBS	Segmentos de anulación para operaciones normales
RBS_2	Segmentos de anulación especiales que se utilizan para la carga de datos
TEMP	Segmentos temporales para operaciones normales
TEMP_USER	Segmentos temporales creados por un usuario concreto
TOOLS	Tablas de herramientas RDBMS
TOOLS_I	Índices de las tablas de herramientas RDBMS
USERS	Objetos de usuario, en las bases de desarrollo

El caso mostrado en la tabla anterior ejemplifica un caso ideal, para sistemas mucho más grandes, ya que para implementar esta configuración se necesitarían 12 discos duros.

Para nuestra base de datos la configuración es la siguiente:

<b>Espacio de Tablas</b>	<b>Utilización</b>
SYSTEM	Diccionario de datos
DATATS	Tablas de operación normal
RBKTS	Segmentos de anulación para operaciones normales
TEMPTS	Segmentos temporales para operaciones normales
TOOLS	Tablas de herramientas RDBMS

El espacio de tablas SYSTEM es el punto de partida, Oracle lo proporciona por default y es donde se guarda el diccionario de datos. Se revocó toda cuota sobre este espacio de tablas prohibiendo la creación de cualquier objeto.

Se creó un table space denominado DATATS, que es el espacio físico donde se almacenan los datos asociados con tablas, agrupaciones, y en nuestro caso con índices.

Asimismo el espacio de tablas RBKTS es de suma importancia, ya que estos segmentos mantienen la concurrencia de los datos.

En el espacio TEMP se almacenan objetos que se generan dinámicamente y también los que almacenan datos durante las operaciones de ordenación de gran tamaño. Y conviene tener el segmento de TOOLS debido a que hay muchas herramientas que almacenan segmentos de datos en SYSTEM ya que este espacio de tablas es el predeterminado para el usuario homónimo.

El hecho de separar estos segmentos de SYSTEM elimina un potencial problema del área del diccionario de datos.

No se creó un espacio de tablas INDEX 1 y 2 o el segundo de DATA y TOOLS debido a que el servidor en cual se aloja la base de datos cuenta solamente con un disco duro, por lo que no ayudaría en nada esta configuración.

Una vez creada la base simplemente ejecutamos el script crebas.sql mostrado en el apéndice.



## 5.2 Desarrollo del sistema

Como ya se había comentado, este sistema esta integrado por dos partes: la interfaz gráfica y la aplicación WEB. Dado que estas dos partes utilizan una base de datos se diseño el sistema de manera que los dos utilizarán la misma base y así tener un sistema integral capaz de cumplir con las necesidades para el cual fue hecho.

A continuación se presenta el desarrollo de cada una de esas partes.

### 5.2.1 Desarrollo de la interfaz gráfica

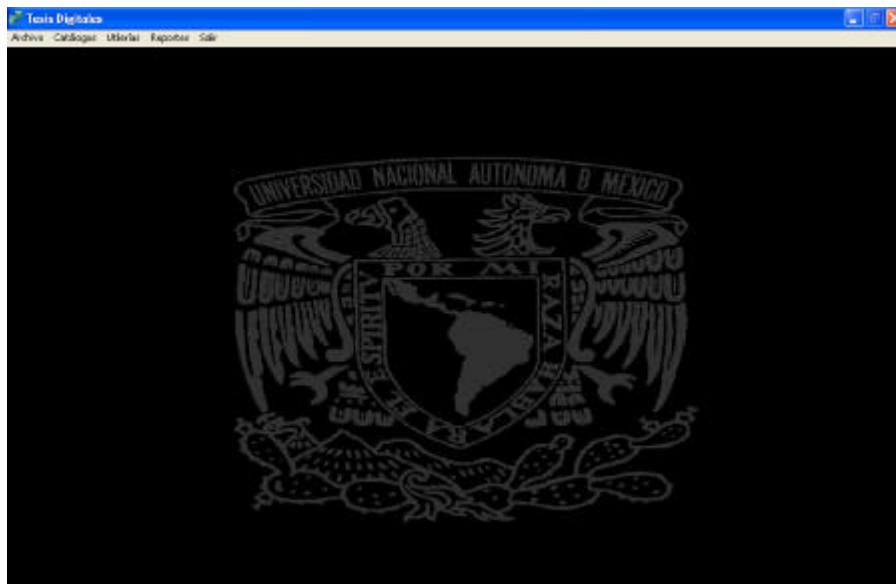
SIRARTE está diseñado para ser utilizado de una forma fácil e intuitiva. Esto se logra mediante el aprovechamiento de una interfaz gráfica. En la medida de lo posible las pantallas poseen atajos de teclado que, para un usuario experto, se verá reflejado en el aumento de su eficiencia en el manejo de las formas de captura.

Como se mencionó en el capítulo anterior, el sistema cuenta con ciertas normas de seguridad, esto se puede observar desde que se ejecuta el sistema, el cual solicita un login y un password para poder ingresar:



En el caso de que el usuario este dado de alta en la base y sus datos sean los correctos después de dar clic en el botón de aceptar se mostrará una pantalla de presentación que desaparece automáticamente después de dos segundos.

Inmediatamente después se abre la pantalla principal en la cual van a estar contenidos todos los módulos del sistema, como se muestra a continuación:



En este momento el sistema ya validó el login y el password y aplica las políticas de seguridad, habilitando o deshabilitando opciones del menú principal como se muestra a continuación:

Perfil Administrador:



Perfil de un Usuario Normal:



Para comenzar el proceso de **registro de una tesis**, se selecciona del menú archivo la opción "Inicia Proceso de Registro de Tesis" como se muestra:



Aparecerá inmediatamente la pantalla de alumnos

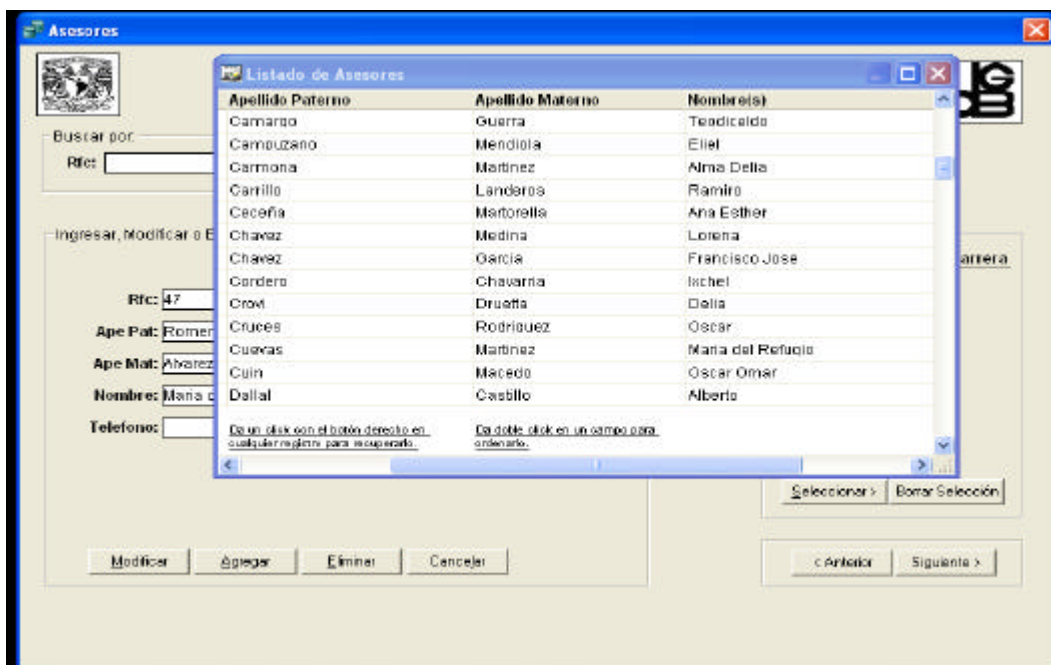
Id	Cve Marc	Nombre Carrera	Nombre Facultad
299	27	Ingenieria Mecanico	Facultad de Ingenieria
300	28	Ingenieria Industrial	Facultad de Ingenieria
301	29	Ingenieria Electrico Electronico	Facultad de Ingenieria
302	30	Ingenieria en Telecomunicaciones	Facultad de Ingenieria
303	31	Ingenieria Geofisico	Facultad de Ingenieria
304	32	Ingenieria en Computacion	Facultad de Ingenieria
305	34	Especialidad en Seguridad en Instalaciones Industriales de Explotacion Petrolera	Facultad de Ingenieria
306	35	Especialidad en Control de Calidad	Facultad de Ingenieria
307	36	Especialidad en Costos de la Construccion	Facultad de Ingenieria
308	37	Especialidad en Geotecnia	Facultad de Ingenieria

Las formas de alumnos, asesores y tesis tienen un funcionamiento similar. Tenemos en la parte superior la sección de "Buscar por:" en la cual tenemos dos o tres opciones de búsqueda. En cada opción de búsqueda tenemos una flecha que nos despliega una lista con los registros que se encuentran en la base de datos. Simplemente tecleando los primeros caracteres del campo de búsqueda el sistema nos acerca al valor buscado, con lo que en algunos casos no es necesario teclear el valor completo para realizar la búsqueda. Una vez que hemos localizado el registro se puede modificar o eliminar pulsando los botones respectivos. También se pueden agregar registros nuevos, dando click al botón de agregar.

En el caso de alumnos, se comienza agregando el número de cuenta que va a ser el identificador único de cada registro, por lo que al momento de ingresarlo y cambiar de casilla el sistema va a verificar que efectivamente el número sea nuevo. En caso de que ya esté en la base de datos, simplemente se va a desplegar la información referente a ese registro. Una vez capturados todos los campos simplemente se presiona el botón de guardar y el sistema lo coloca en la lista de alumnos seleccionados, junto con la clave de su carrera.



Podemos seguir ingresando alumnos e irlos seleccionando en la lista en caso de que la tesis así lo requiera. Una vez ingresados los alumnos se ingresan los asesores dando click en el botón siguiente. De la misma forma se llenan los datos del asesor y se pulsa siguiente:



Aparece la forma de captura de tesis, la cual funciona igual que las anteriores, pero con la diferencia que aparece un campo de matriz que se llena automáticamente con el siguiente consecutivo si se deja vacío.

The screenshot shows a web application window titled 'Tesis' from the 'DIRECCIÓN GENERAL DE BIBLIOTECAS DEPARTAMENTO DE TESIS'. The interface includes a search bar, a form for entering thesis details, and a table for sub-titles.

**Form Fields:**

- Buscar por: Matriz:  Título:
- Ingresar, Modificar o Eliminar Tesis:
  - Matriz:
  - Título:
  - Num Páginas:
  - Año:
  - Adicional:
  - Ilustraciones:  Sí  No

**Table:**

Matriz	Id Sub.	Subtítulo
118	1	subtitulo 1
118	2	

**Buttons:** Modificar, Guardar, Eliminar, Cancelar, Seleccionar >, < Anterior, Siguiente >

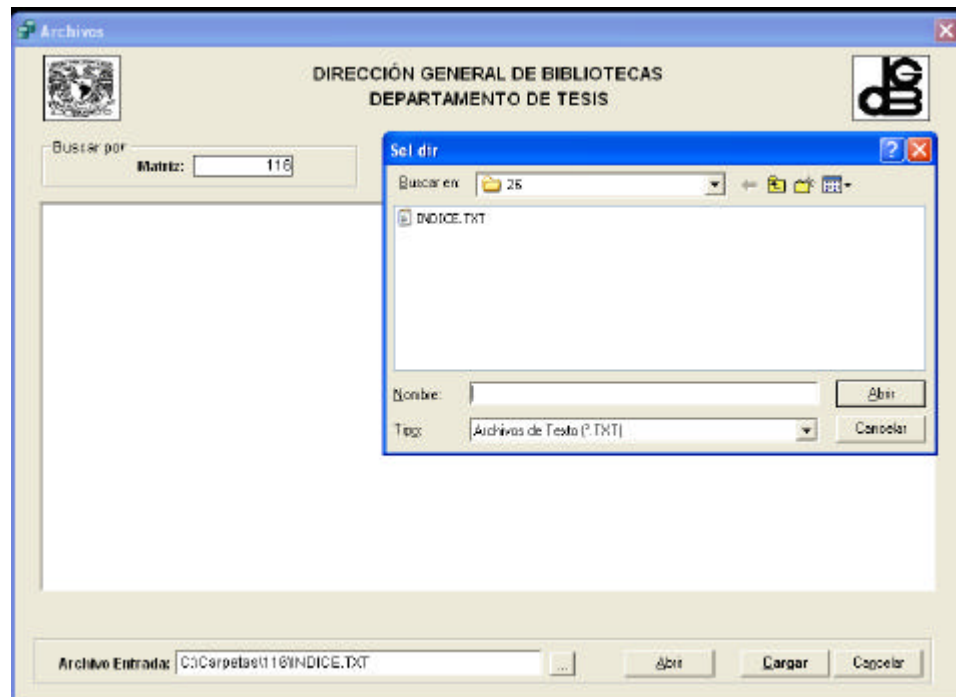
En la parte inferior aparece un cuadro de subtítulos, los cuales vamos a ir ingresando dando click al primer botón colocado en la parte superior derecha del cuadro, o simplemente dando enter en el mismo. Automáticamente pasa los valores de matriz y de id\_subtema, por lo que simplemente se tiene que teclear los subtítulos y presionar guardar al finalizar. Después se presiona el botón de seleccionar y después el de siguiente.

Inmediatamente después aparece la ventana de Capítulos y Subtemas. Aparece una selección de matriz, la cual automáticamente lleva la matriz sobre la cual hemos estado trabajando. Aquí existen dos opciones de captura de datos, la primera es darla de alta manualmente y la segunda es cargarla de un archivo.

En la captura manual el usuario simplemente se coloca sobre el cuadro al que quiera ingresar registros y presiona el botón de enter, lo cual agregará un nuevo registro. Cabe señalar que se pueden ingresar varios subtemas por cada capítulo, por lo que para asociar cada subtema a cada capítulo primero se selecciona el capítulo y después se agregan los subtemas asociados a este.



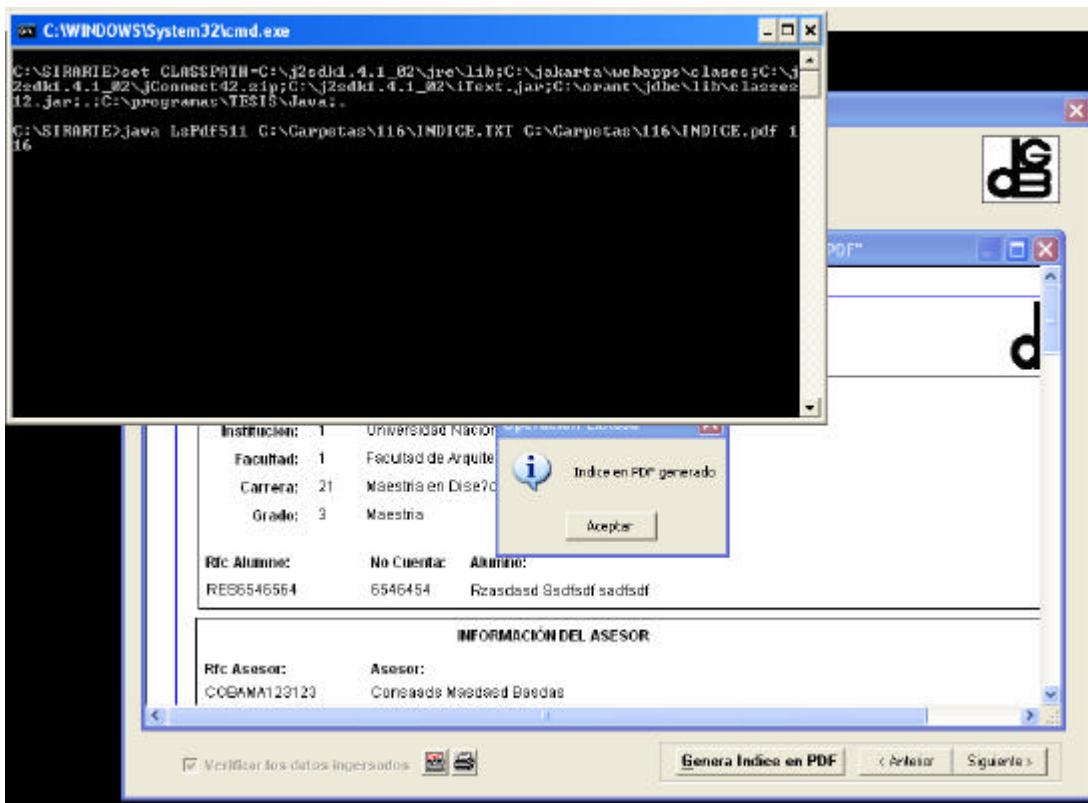
Para realizar la carga desde un archivo se presiona el botón correspondiente (Carga desde archivo) y se despliega la siguiente pantalla:



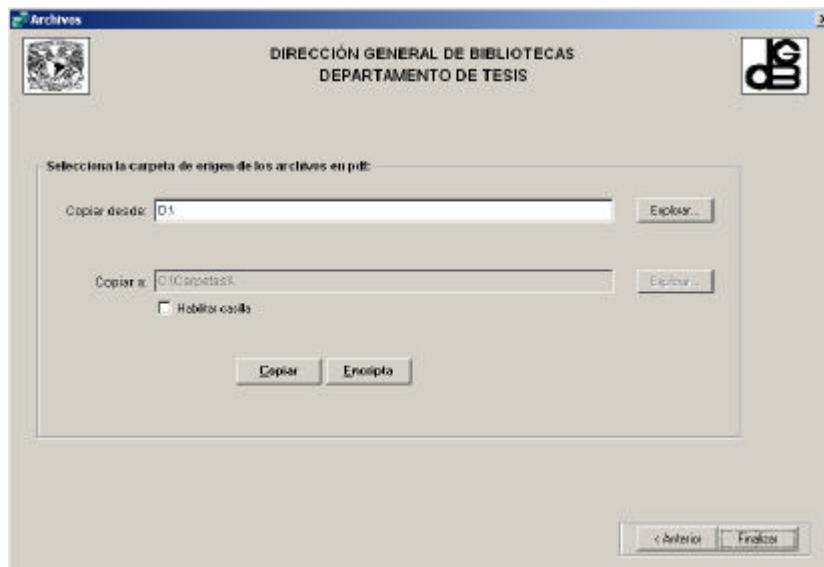
Aquí simplemente seleccionamos el archivo de texto que contiene el índice, respetando el formato dado, y presionamos el botón abrir. Nos muestra el contenido del archivo y entonces presionamos el botón cargar. Automáticamente nos regresa a la pantalla de capítulos con la información ya cargada. Damos click en el botón siguiente, y nos despliega la pantalla de Revisión como se muestra a continuación:



En esta pantalla aparece un reporte de todos los datos relacionados con esa matriz, desde la cual podemos mandar generar el índice en formato pdf.



Por último presionamos el botón de siguiente y lo único que resta es copiar los archivos de cada capítulo de la tesis que entregue el alumno y darles seguridad en caso de ser necesario.





Solamente seleccionamos la carpeta donde están los archivos de origen, la carpeta de destino viene dada por el sistema dependiendo de las configuraciones iniciales. Se puede modificar habilitándola con el check box, pero regularmente esta no cambia.

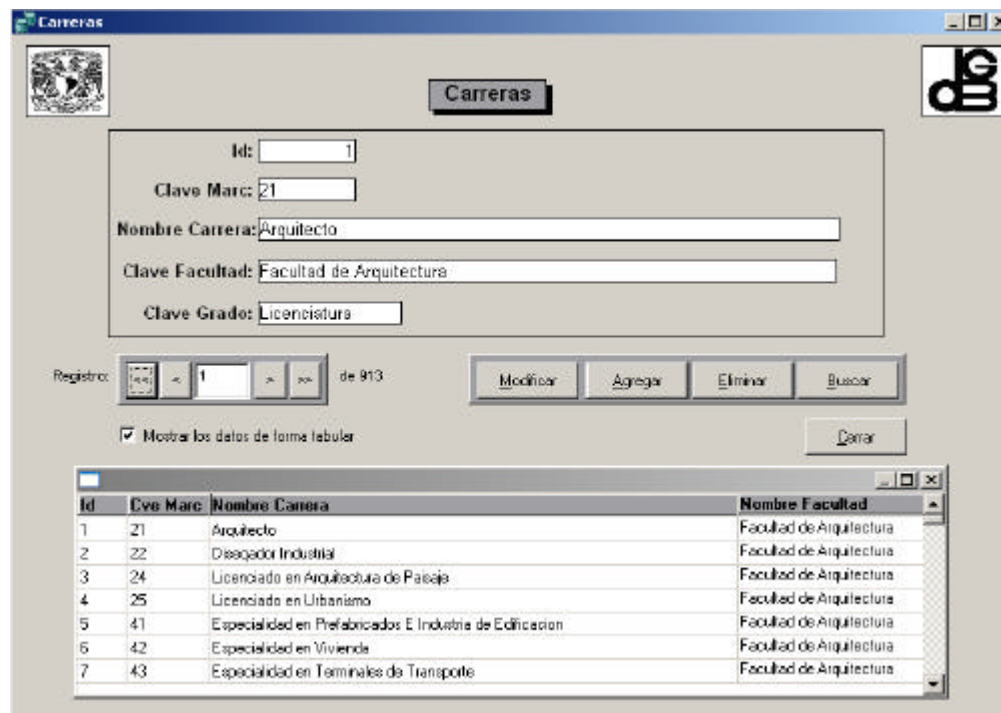
Para darles seguridad a los archivos simplemente se selecciona la carpeta en donde se encuentran los PDF's y se presiona el botón de Encripta. Si el encriptado fue exitoso se muestra el mensaje de confirmación.

Por último damos clic en el botón de Finalizar y nuestra tesis ya quedó debidamente registrada.

En esta parte del sistema se encuentran los **catálogos** del sistema: Instituciones, Facultades, Carreras y grados. Además se podrán manejar a los usuarios (alta, baja, modificación) y hacer modificaciones o altas a los perfiles de usuario existentes.



El siguiente catálogo muestra todas las carreras las cuales, al igual que el catálogo de facultades, instituciones, grados, usuarios y perfiles de usuario podrán hacer modificaciones, Agregar, eliminar y hacer búsquedas como se muestra en la siguiente imagen.



## UTILERÍAS

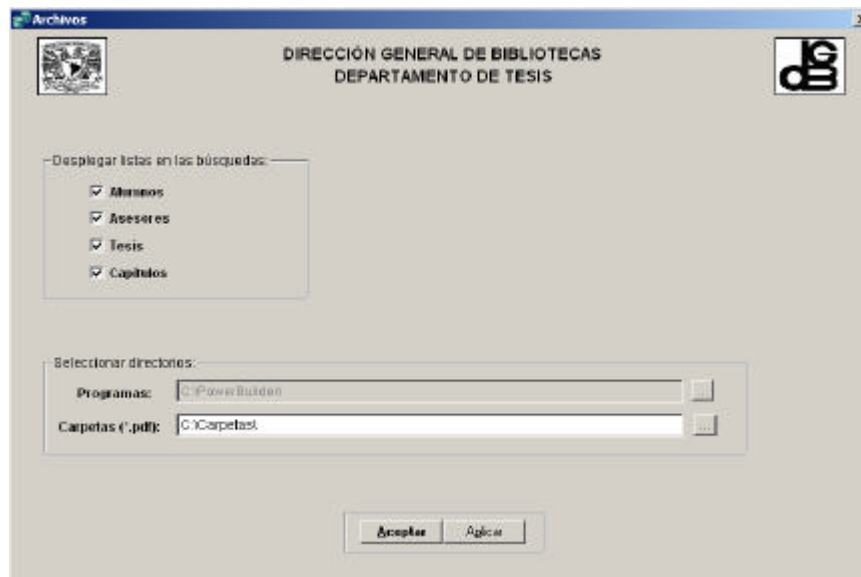
En este módulo se encuentran las opciones administrativas, el desbloqueo de tesis, copiar archivos e importar el índice.



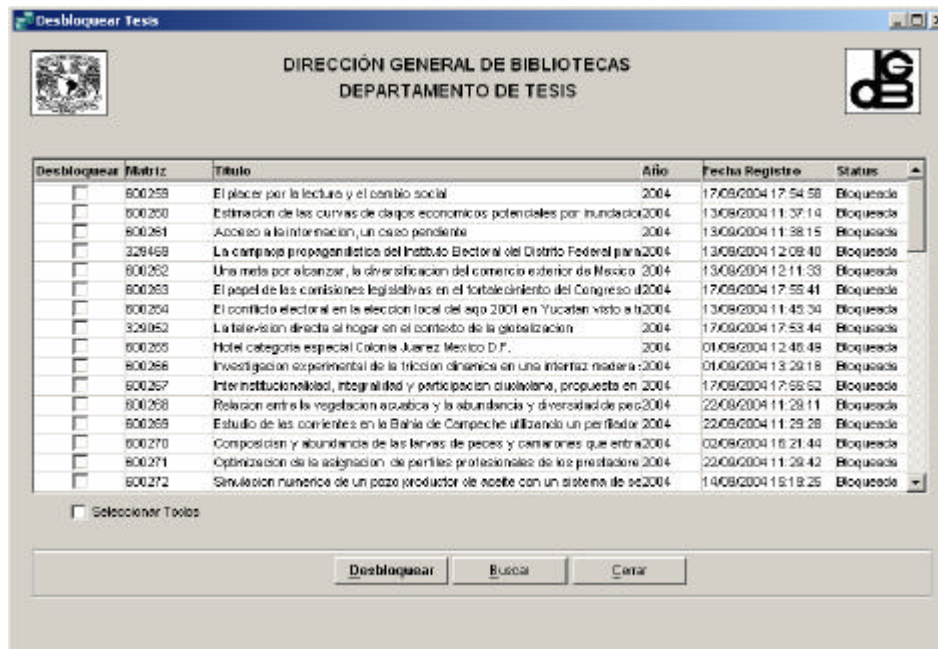
### Opciones administrativas.

En esta pantalla se muestra el control de las opciones administrativas. Se tienen unas opciones para el despliegue de la información a la hora de hacer búsquedas, es decir, si no están seleccionadas entonces no se despliegan. En este caso están seleccionadas por lo que se despliegan en todas las búsquedas. También se tienen las opciones de selección de directorios las cuales nos ayudan a establecer la ubicación de donde será tomada la aplicación y la dirección en donde están colocadas todas las carpetas con sus respectivos PDF's.

Para que los cambios surjan efecto se da un click en el botón de Aplicar.



En esta pantalla se podrán **desbloquear tesis** en el caso de que lo estén. Una tesis es bloqueada cuando el proceso de registro ha sido completado y solo se desbloqueará cuando se le tenga que hacer una corrección. Las tesis se bloquean con el propósito de no tener mucha información cuando se desplieguen todas las tesis.



Las opciones de copiar archivos e importar índices ya fueron explicadas anteriormente. Se podrán generar reportes, tales como el que se muestra en la siguiente pantalla:



El sistema permite configurar la impresión y mandar la misma.

Las opciones para este menú se muestran en la pantalla siguiente:



Cuando se selecciona Salir por completo del sistema se sale por completo del sistema y cuando se selecciona Cambiar de Usuario se muestra la pantalla de inicio para poder entrar como otro usuario diferente.

### 5.2.2 Desarrollo de la aplicación WEB

Esta aplicación WEB se desarrollo con Java Server Pages y sobre el servidor de WEB Jakarta Tomcat.

A continuación se muestra el desarrollo de esta aplicación:

Inicialmente se colocó una pantalla en la que el usuario tiene que identificarse. Para esto el usuario debe de contar con un login y un password que se le asignará por medio de servicios escolares de su respectiva facultad.

La pantalla inicial es la siguiente:



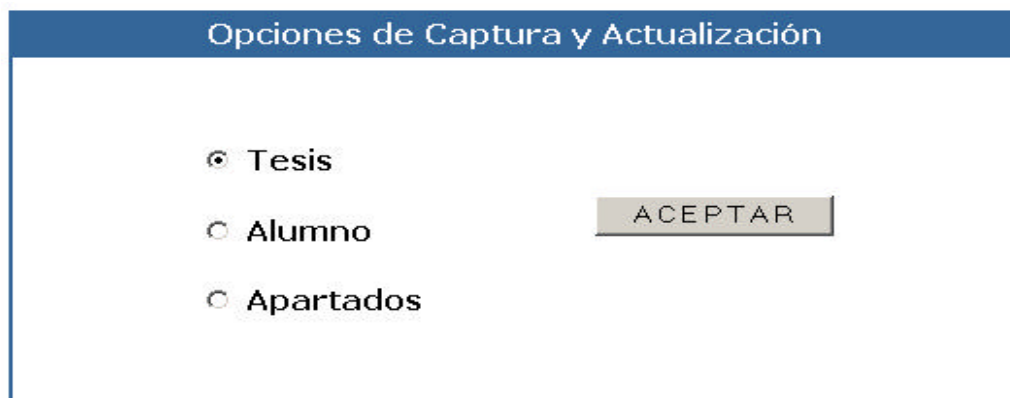
SIRARTE WEB

Login

Contraseña

ACEPTAR

Una vez que el usuario logra entrar al sistema se colocó un menú que nos permitiera hacer las operaciones necesarias para la captura de los datos de la tesis, de los datos del alumno y los datos de los apartados de una tesis.



Opciones de Captura y Actualización

Tesis

Alumno

Apartados

ACEPTAR

[SALIR](#)

En cualquier parte de la aplicación el usuario tiene la oportunidad de salir o de hacer cualquier otra operación en otra parte de la aplicación. Para salir, el usuario solamente hace click en la liga de SALIR.

## TESIS

En esta parte de la aplicación tiene la oportunidad de hacer el registro de los datos de la tesis. Estos datos son los mismos que los de la interfaz gráfica y son: Título, subtítulos (si tiene), número de páginas, año, adicional y si cuenta con ilustraciones o no.

DATOS DE LA TESIS	
Título	<input type="text"/>
Subtítulo(s)	<input type="text"/> <small>Cada vez que ingrese un subtítulo presiona ENTER</small>
Número de páginas	<input type="text"/> (4 números)
Año	<input type="text" value="0"/> (4 números)
Adicional	<input type="text"/> (opcional)
Ilustraciones	<input type="radio"/> Sí <input type="radio"/> No

ACTUALIZAR

[VER DATOS](#)

[OPCIONES](#)

[SALIR](#)

Además, si ya capturo algunos datos también tiene la oportunidad de visualizarlos y de hacer la actualización de estos cuando el lo crea necesario.

Bienvenido													
<p><b>Actualizar datos</b></p> <p>Si desea actualizar sus datos haga click <a href="#">aquí</a></p>	<table border="1"> <thead> <tr> <th colspan="2">Datos de la Tesis</th> </tr> </thead> <tbody> <tr> <td>Título</td> <td>El material ecologico en la zona de xochimilco</td> </tr> <tr> <td>Número de Páginas</td> <td>120</td> </tr> <tr> <td>Año</td> <td>2004</td> </tr> <tr> <td>Adicionales</td> <td>Video</td> </tr> <tr> <td>Ilustraciones</td> <td>s</td> </tr> </tbody> </table> <p style="text-align: center;"><a href="#">OPCIONES</a> <a href="#">SALIR</a></p>	Datos de la Tesis		Título	El material ecologico en la zona de xochimilco	Número de Páginas	120	Año	2004	Adicionales	Video	Ilustraciones	s
Datos de la Tesis													
Título	El material ecologico en la zona de xochimilco												
Número de Páginas	120												
Año	2004												
Adicionales	Video												
Ilustraciones	s												

En el desarrollo de la aplicación se establecieron algunos mensajes para que al usuario le sea fácil el manejo y tenga claro en que momento se realizó alguna operación exitosa.

Algunos de estos mensajes se muestran a continuación:

**Introduzca su login y contraseña por favor**

**INICIO**

**El alumno no existe, favor de verificar sus datos !!!**

**INICIO**

Resultado del registro

**Felicidades!!**

**La actualizacion ha sido realizada satisfactoriamente.**

**Nombre de la Tesis:** El material ecologico en la zona de xochimilco

**CONTINUAR**

Resultado del registro

**Felicidades!!**

**La insercion ha sido realizada satisfactoriamente.**

**CONTINUAR**

Resultado del registro

**Felicidades!!**

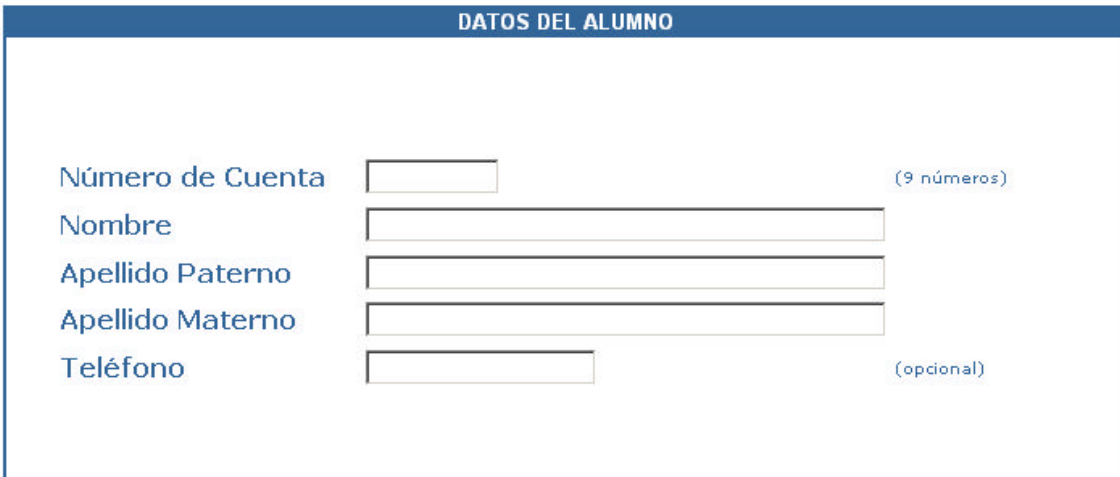
**El borrado ha sido realizado satisfactoriamente.**

**CONTINUAR**

## ALUMNO

En la parte de alumno también se hace el registro de los principales datos de un alumno que son: número de cuenta, nombre, apellido paterno, apellido materno y teléfono (no obligatorio). Se tiene la oportunidad de hacer el registro por primera vez o de hacer la visualización y si se requiere, hacer la actualización de los datos.

Las pantalla por medio de la cual se hace el registro es la siguiente:



El formulario, titulado "DATOS DEL ALUMNO", contiene los siguientes campos de entrada:

Número de Cuenta	<input type="text"/>	(9 números)
Nombre	<input type="text"/>	
Apellido Paterno	<input type="text"/>	
Apellido Materno	<input type="text"/>	
Teléfono	<input type="text"/>	(opcional)

ACTUALIZAR

[VER DATOS](#)

[OPCIONES](#)

[SALIR](#)

Como se ve, también se cuenta con otras opciones tales como: Ver datos (permite visualizar los datos y hacer actualizaciones de los mismos), opciones (permite regresar al menú principal. Tesis, alumno y apartados).

La pantalla de visualización de datos se desarrollo pensando en que el alumno quisiera cambiar algunos de sus datos pero primero tendría que revisar. La pantalla se muestra a continuación:



Bienvenido

Actualizar datos

Si desea actualizar sus datos haga click [aquí](#)

Datos del Alumno

Número de Cuenta	094295025
Nombre completo	Israel Durán Martínez
Teléfono	54879621

[OPCIONES](#)

[SALIR](#)

## APARTADOS

Un apartado es una parte de una tesis. Como se menciona en el capítulo 3 de este trabajo, las tesis pueden dividirse en varias partes y a cada una de esas partes se le llama apartado.

En la Dirección General de Bibliotecas ya se estableció cual va a ser la forma en que se dividirán las tesis y conforme a eso el alumno tendrá que capturar los apartados de su tesis.

Para tal propósito se tiene un menú que nos permitirá manipular a cada uno de los apartados, ya sea que se quiera ingresar un nuevo apartado, hacer la actualización o el borrado de alguno de ellos. A continuación se muestra el menú para el manejo de apartados:

Modificación de Apartados

Ingresar

Actualizar

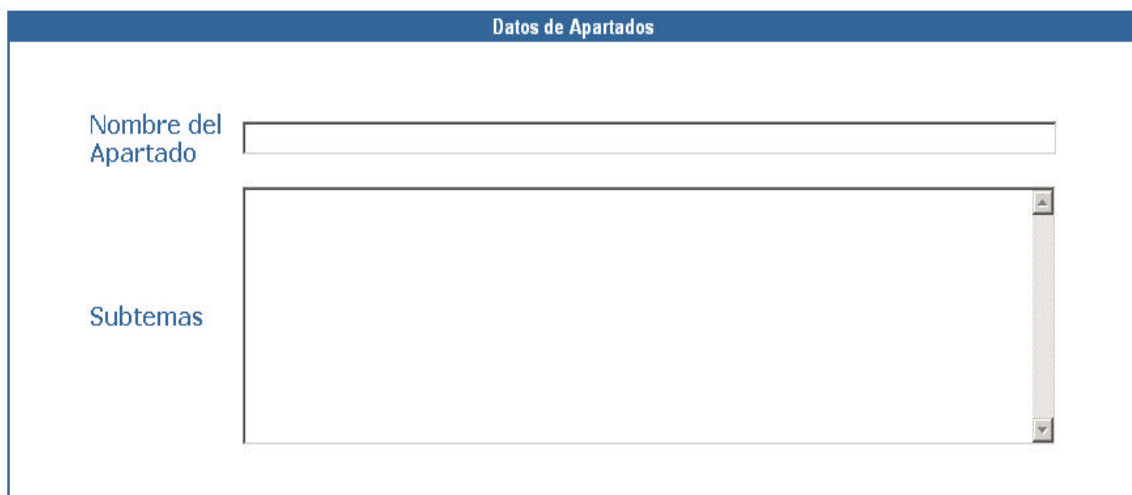
Borrar

[SALIR](#)

## Ingresar

En esta parte se hace el ingreso de un nuevo apartado. Cada vez que se ingrese un apartado este se colocará al final de los demás, es decir, es como si se tuviera una lista y cada vez que se ingrese uno se colocará al final de la lista.

Los datos a ingresar son: el nombre del apartado y en caso de que ese apartado sea un capítulo podría o no tener subtemas. La pantalla de captura para los apartados es:



El formulario tiene un encabezado azul con el título "Datos de Apartados". A la izquierda, hay dos etiquetas: "Nombre del Apartado" y "Subtemas". A la derecha de "Nombre del Apartado" hay un campo de texto rectangular. A la derecha de "Subtemas" hay un área de lista rectangular con una barra de desplazamiento vertical a su lado derecho.

MANDAR DATOS

[OPCIONES PARA CAPITULO](#)

[OPCIONES](#)

[SALIR](#)

## Actualizar

En esta parte se pueden hacer actualizaciones y visualizar los datos que ya fueron capturados. En realidad es muy fácil hacer una actualización, solamente se selecciona de una lista al apartado que se quiera actualizar, se muestra la pantalla con los datos, se hacen las modificaciones pertinentes y se hace click en un botón para que la actualización tenga efecto. La pantalla en donde se escoge al apartado es la siguiente:



ACTUALIZAR

[OPCIONES PARA APARTADOS](#)

[OPCIONES](#)

[SALIR](#)

Y la pantalla de actualización es:



MANDAR DATOS

[OPCIONES PARA CAPITULO](#)

[OPCIONES](#)

[SALIR](#)

### Borrar

Finalmente se tiene la opción de borrar un apartado. Cuando se borra un apartado se borra de la lista completa. El método para borrar un apartado es muy parecido al de actualización. La pantalla en donde se escoge que apartado borrar es:

Selecciona el apartado que deseas borrar

Apartados

[OPCIONES PARA APARTADOS](#)

[OPCIONES](#)

[SALIR](#)

Una vez que se selecciona que apartado borrar se da click en el botón de borrar y aparecerá un mensaje diciendo que el apartado ha sido borrado.

## **CAPÍTULO 6**

---

---

## **IMPLEMENTACIÓN, PRUEBAS Y MANTENIMIENTO**

Hasta este punto ya se realizó el análisis, el diseño y el desarrollo del sistema, ahora es preciso implementarlo, hacer pruebas y tener los recursos para lograr mantenerlo.

### **6.1 Instalación del DBMS.**

Para tal efecto necesitamos configurar el servidor y el sistema manejador de base de datos que en este caso serían Oracle 8.0.5 sobre un servidor Sunblade con Solaris 8. Para instalar este dbms el servidor requiere como mínimo:

<b>Requerimiento</b>	<b>Valor Mínimo</b>
Espacio en disco	600 MB
Memoria	32 MB
Versión de Sistema Operativo	Solaris 2.5.1 o superior

Una vez instalado el sistema se crea la instancia de la base de datos y se corre el script que crea las tablas y los objetos de la misma.

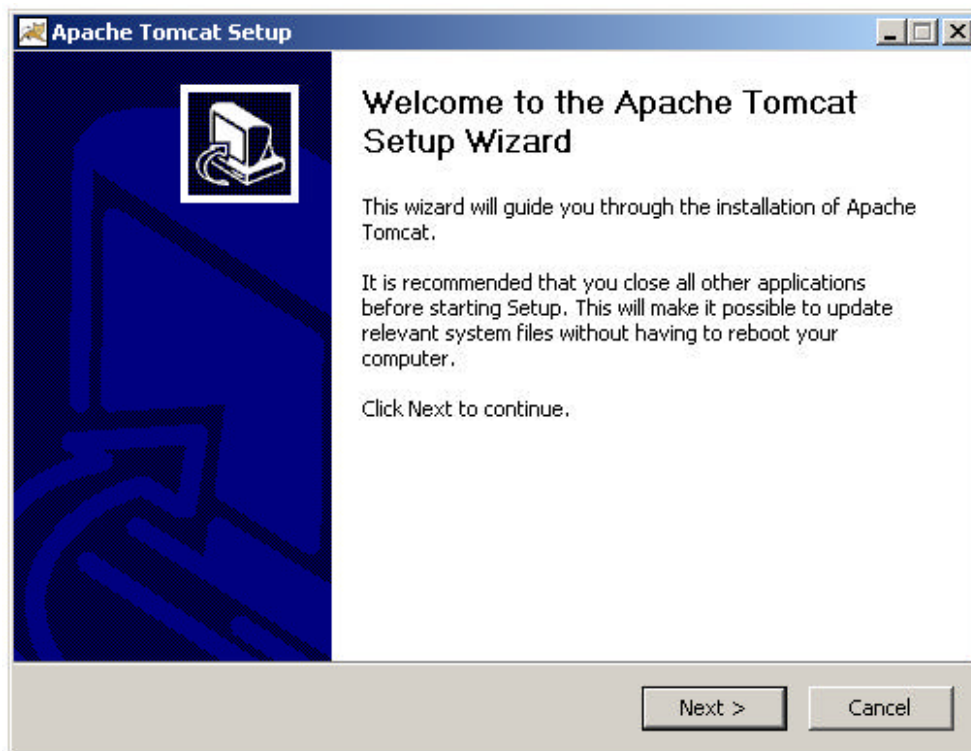
Asimismo es necesario configurar el listener de la base de datos a través del puerto 1521 para que aplicaciones externas a esta se puedan conectar remotamente.

### **6.2 Implementación del servidor WEB**

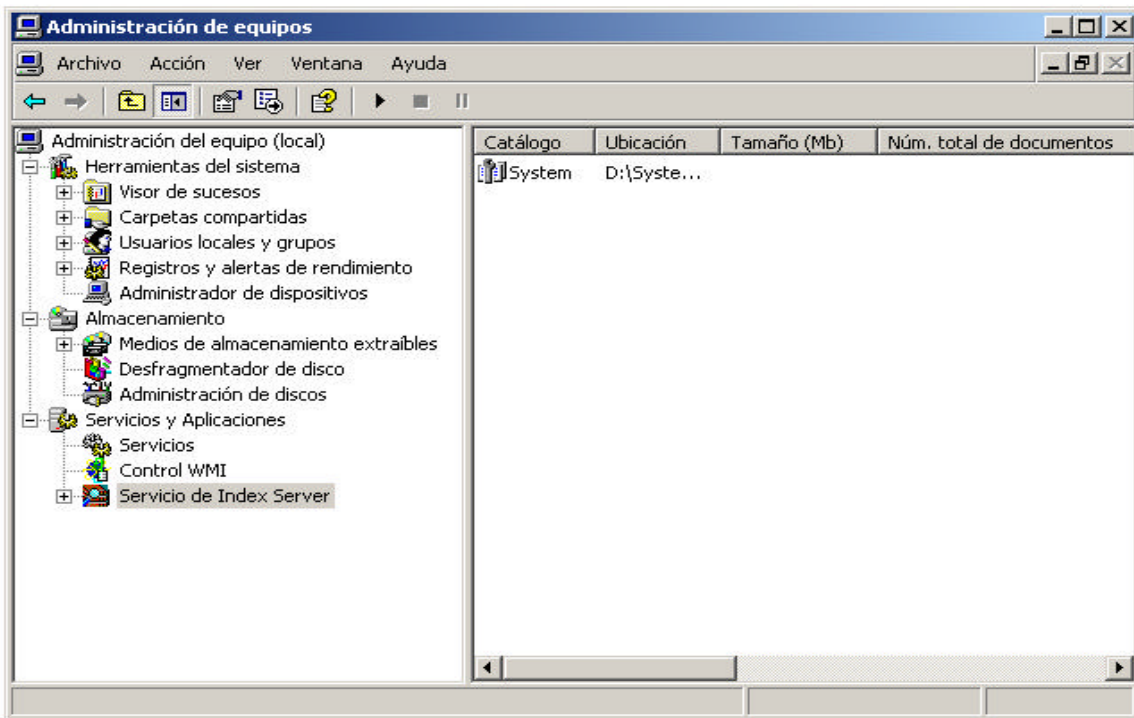
El servidor WEB que se utilizó para la aplicación WEB es Jakarta-Tomcat-5.0.19. Se analizó en su debido tiempo y se decidió que este fuera el servidor a utilizar sin perder de vista que en el futuro la aplicación podría montarse sobre otro servidor diferente.

Lo primero que se hizo fue hacer la instalación del Tomcat sobre un servidor con Windows 2000 server, el cual es administrado por el Departamento de Mantenimiento, Análisis y Desarrollo de sistemas.

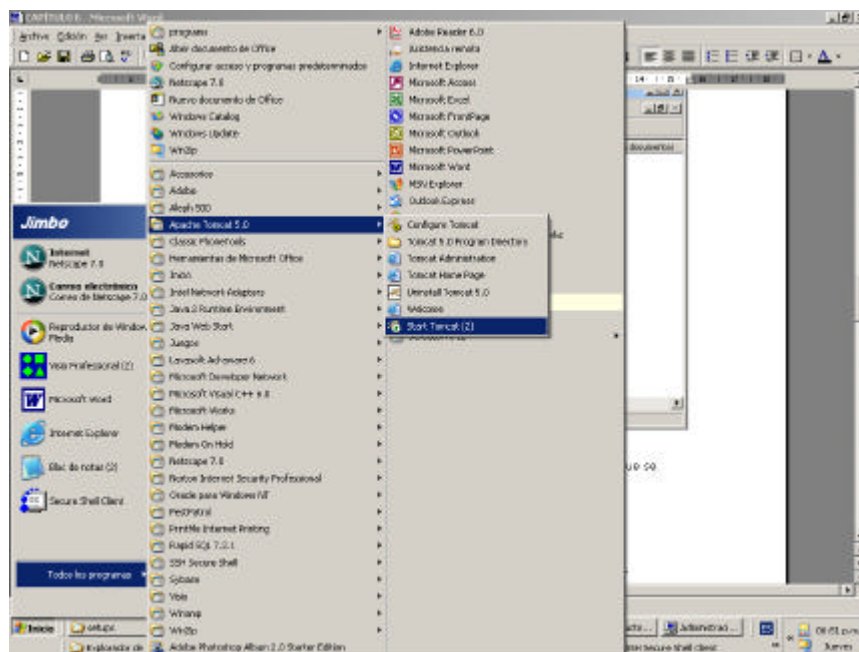
En la siguiente pantalla se muestra una de las pantallas de instalación:



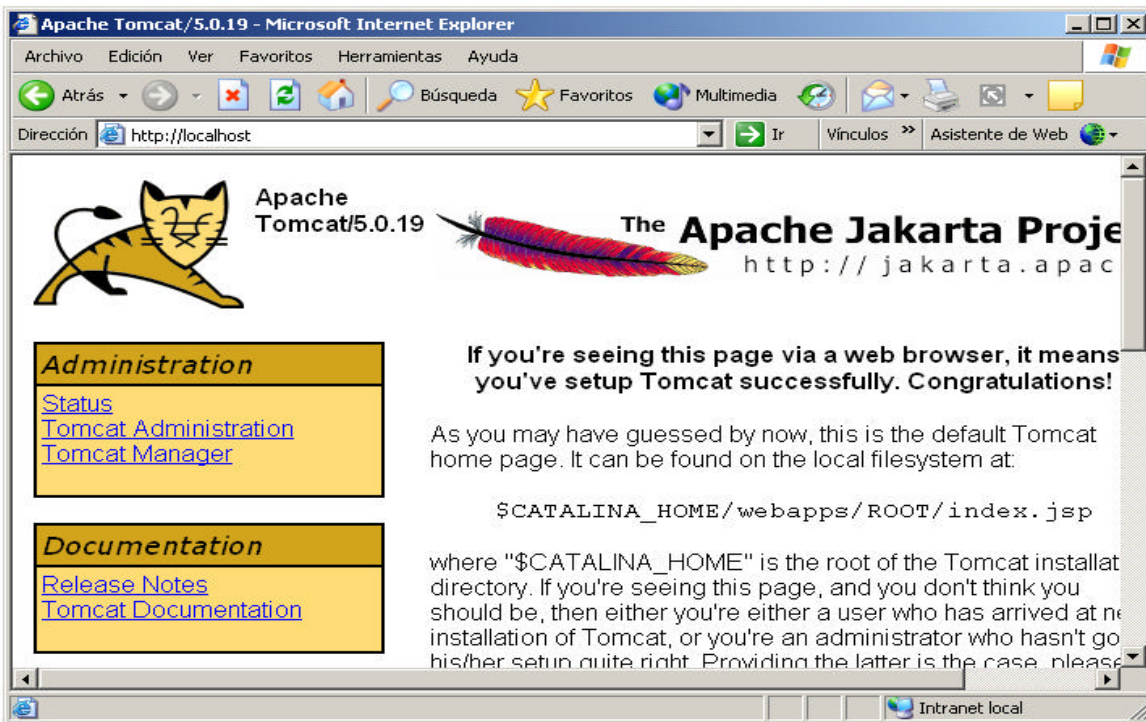
Una vez que el servidor se instaló se procede a hacer la configuración de la máquina para que tome al Tomcat como el servidor por default. Esto se hace en la parte de administración de equipos.



Hecho lo anterior b que prosigue es levantar el servidor. Esto se hace cuando se click en **Inicio – Todos los programas – Apache Tomcat 5.0 - Start Tomcat** como se muestra a continuación:



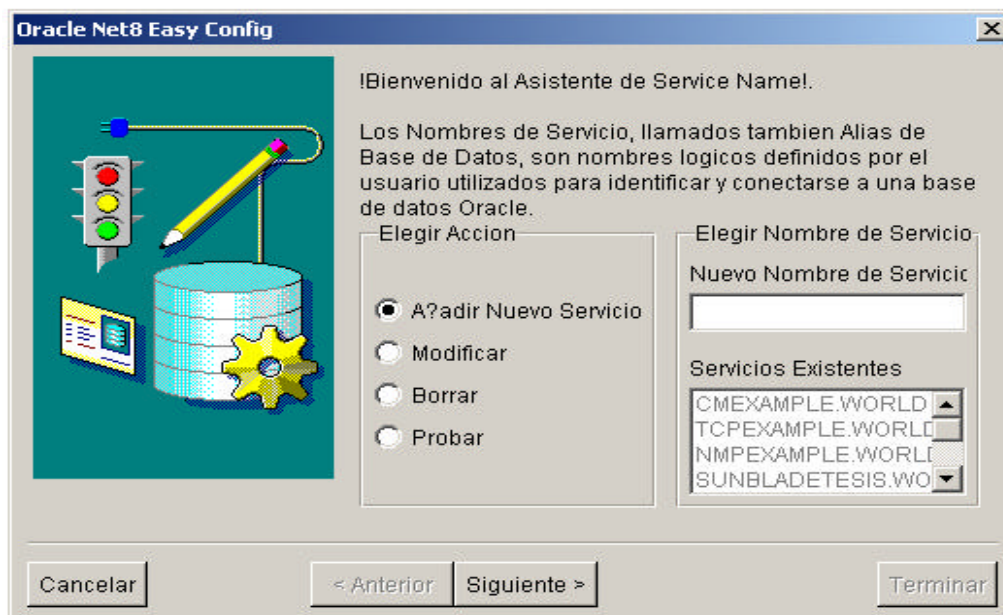




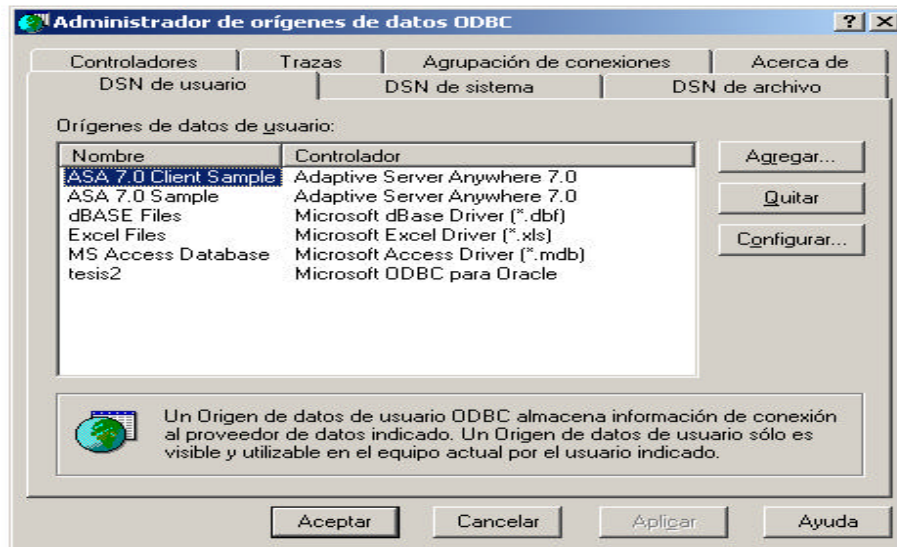
### 6.3 Implementación de la Interfaz gráfica

Para instalar la aplicación cliente se deben seguir los siguientes pasos:

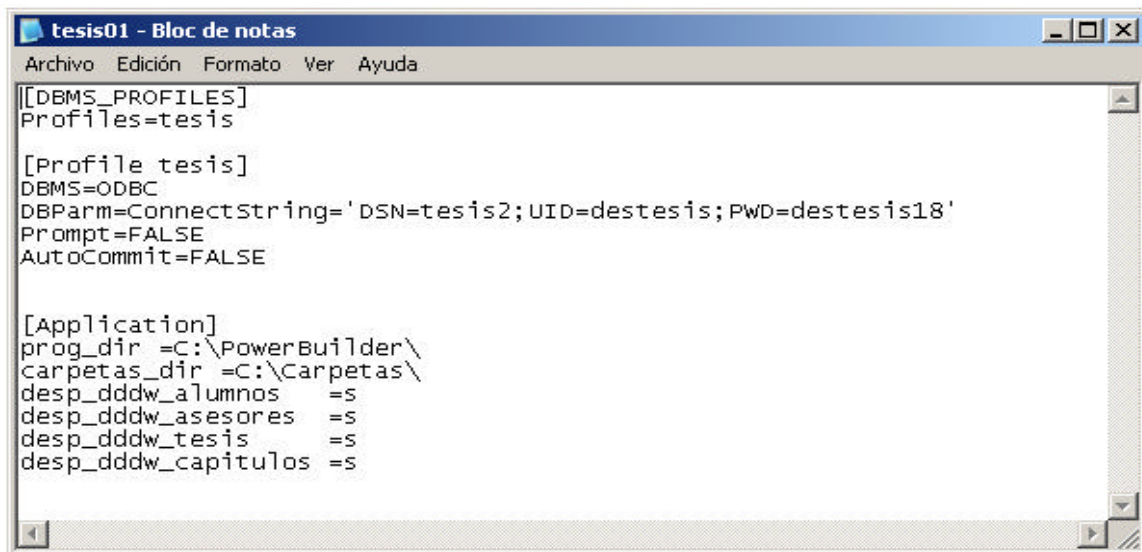
- Instalar el Net 8 Client de Oracle, necesario para conectarnos a la base.



- Dar de alta el servicio en la herramienta anterior, configurándolo a través del puerto 1521 y con el Oracle SID denominado TESIS.
- Crear el ODBC para este servicio.



- Copiar el ejecutable y sus librerías en una carpeta del sistema.
- Configurar el archivo .ini indicándole dónde debe guardar los archivos de salida y cuál es la ruta del ejecutable.



- Se instalan y configuran las variables de ambiente de Java. Además de la instalación de un paquete de Java llamado iText.jar. Todo lo anterior es necesario debido a que una parte de la aplicación utiliza un programa en Java. Además, se configuran las variables de ambiente PATH y CLASSPATH.

Por default el sistema genera una cuenta de usuario con permisos de administrador, que debe ser utilizada para establecer la información de los catálogos de carreras, facultades, instituciones y grados la cual será requerida durante el registro de la tesis. Además es necesario que el administrador del sistema cree las cuentas para los demás usuarios del sistema y les asigne el perfil correspondiente.

#### **6.4 Funcionalidad del sistema**

Una vez finalizada la instalación tanto del software como de la base de datos se podrá utilizar la aplicación GUI y la aplicación WEB. Las dos hacen referencia a la misma base de datos permitiendo hacer la captura o el ingreso de la información de los datos de la tesis, datos del alumno(s), datos del asesor(es) y datos de los apartados de la tesis. Sin embargo, existen diferencias considerables entre una y otra, que justifican el desarrollo de ambas.

La aplicación WEB además de servir para el preregistro de datos, que se hace de manera remota, tiene la ventaja de que el alumno es quien la realiza y no el personal del Departamento de Tesis, con lo cual, el proceso de registro de la tesis se agiliza considerablemente.

En la aplicación GUI se completa y se revisan los datos que se capturaron por medio de Internet pero además se cuenta con una serie de utilerías que nos permiten, entre otras cosas, el manejo y control de toda la información, la transformación del índice de txt a PDF y la creación de una carpeta que haga referencia a la tesis en la cual se generará, también por medio del sistema, una copia con seguridad de los archivos PDF, restringiendo las propiedades de impresión, copiado y pegado, entre otras.

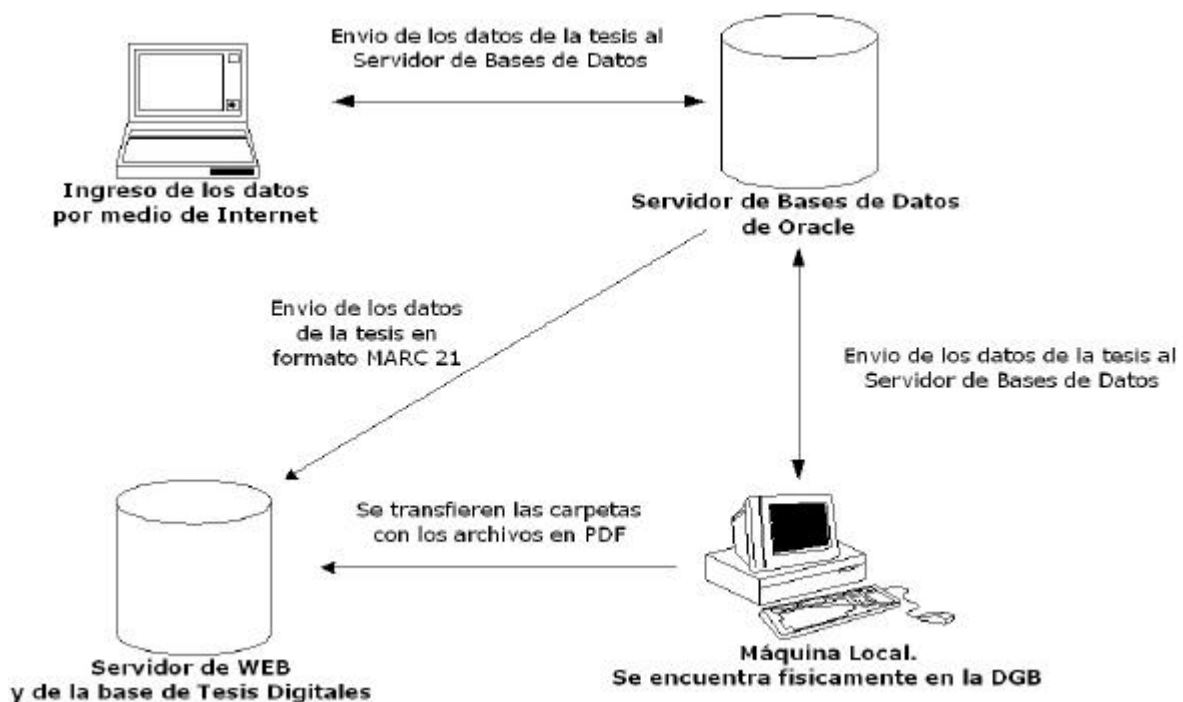
Se tiene que hacer hincapié que la base a donde llegan los datos de la aplicación WEB y de la interfaz GUI es intermedia. Esto se debe a que actualmente la

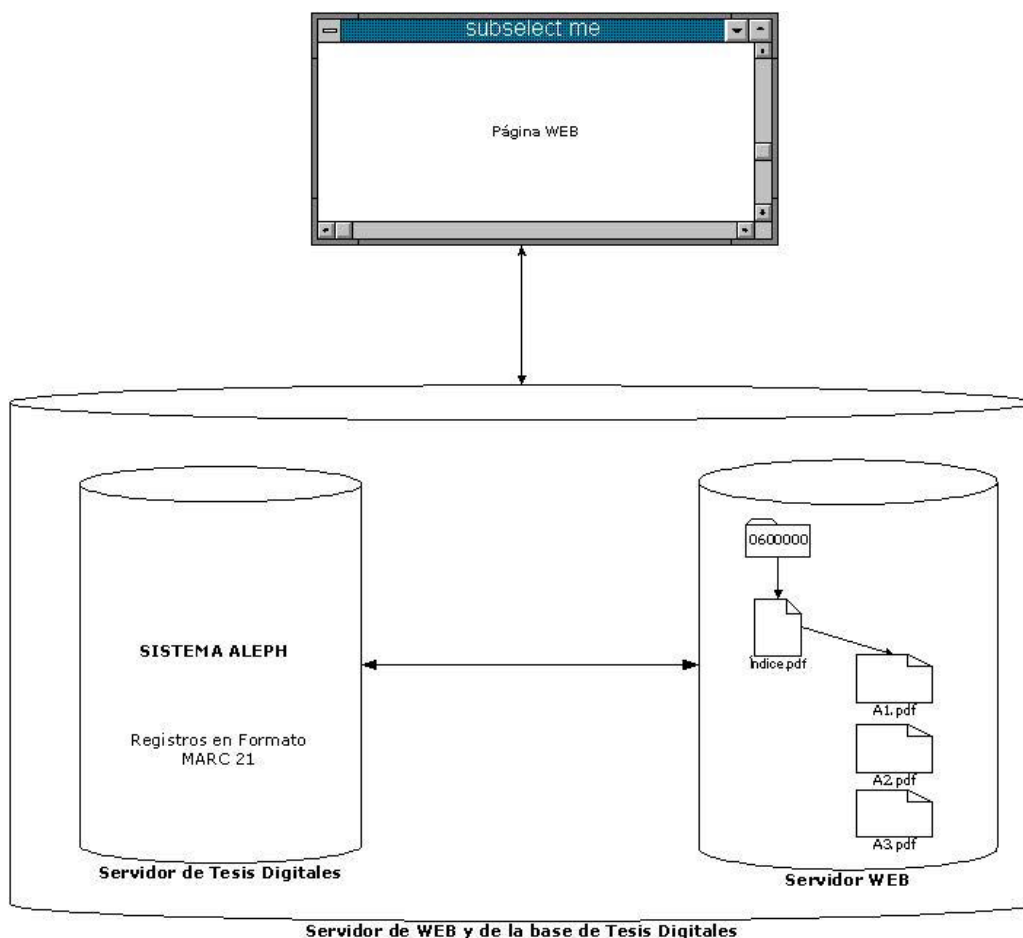
base de tesis digitales trabaja con el sistema denominado ALEPH 300 el cual funciona sobre un manejador de archivos y solo maneja la información en formato MARC 21. Sin embargo, cabe señalar que es la que nos servirá para posteriormente hacer las búsquedas por título, autor, año, grado, etc. y hacer la recuperación del texto completo.

La base intermedia tiene las ventajas de una relacional, es decir, evita la redundancia y la inconsistencia de la información. Además, se podrá manejar la información de una manera eficiente, rápida y nos permitirá generar automáticamente los archivos con formato MARC 21 ejecutando un procedimiento almacenado.

### 6.5 Implementación para la búsqueda y visualización de las tesis digitales.

Primeramente se mostrarán dos diagramas que muestran el proceso completo para lograr hacer búsquedas y poder visualizar las tesis digitales.





Como se muestra en el diagrama, tienen que hacerse varias transferencias al servidor de WEB. Una de ellas es la transferencia de las carpetas que ya fueron generadas por medio de la aplicación Cliente – Servidor. Estas carpetas contienen todos los apartados de la tesis en formato PDF. Esta transferencia se hace por medio del Secure File Transfer.

Esas carpetas se guardan en un directorio. El nombre de cada carpeta esta asignado por un número de matriz, que es un identificador único para cada tesis.

Además se hace la transferencia de los archivos con formato MARC 21 al servidor en donde se encuentra la base de Tesis Digitales y una vez ahí, se carga el archivo a la base por medio de un proceso que utiliza ALEPH que es un sistema integral de bibliotecas, global, comprensible y totalmente integrado, desde la adquisición de materiales hasta la consulta en Web, basado en tablas de parámetros

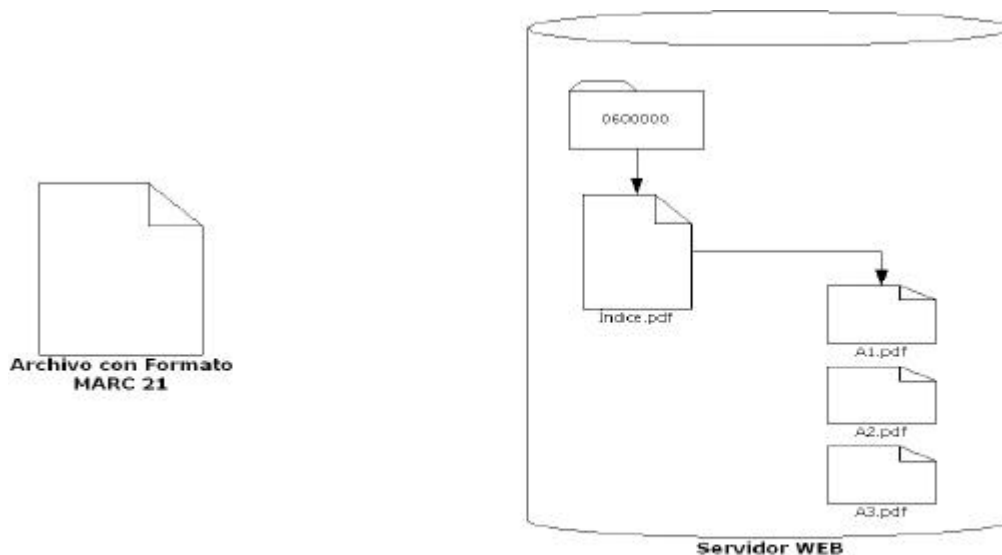
definidas por las propias instituciones usuarias para adecuar el sistema a sus necesidades y aplicaciones específicas.

El formato de estos archivos se muestra a continuación:

```

TDF 0600000 BASE L 01
TDF 0600000 008 L S2003 DGB A ESP
TDF 0600000 090 L $$a001-01039-E2-2003
TDF 0600000 1002 L $$aEstrada Curiel, Aura Adriana
TDF 0600000 24510 L $$aBorges y la tradicion del ensayo ingles
TDF 0600000 260 L $$aMexico : $$bEl autor,$$c2003
TDF 0600000 300 L $$a134 p. : $$bil.
TDF 0600000 502 L $$aTesis Maestria (Maestria en Letras)-UNAM, Facultad de Filosofia y Letras
TDF 0600000 70021 L $$aLara Zavala, Hernan,$$easesor
TDF 0600000 71021 L $$aUniversidad Nacional Autonoma de Mexico.$$bFacultad de Filosofia y Letras
TDF 0600000 8567 L $$xhttp://132.248.9.25:8080/tesdig/ips/entrada.jsp?entrada=Procesados_2004/0600000/Indice.pdf$$nIndice
TDF 0600000 900 L $$aBC$$b000$$x0000600000
TDF 0600000 LN L SPA
    
```

La forma en que se hace la relación entre las carpetas y los archivos MARC 21 es que el archivo tiene la liga que hace referencia a la tesis en texto completo. El siguiente esquema muestra esa relación.



**Esta liga esta contenida en el archivo con formato MARC 21**

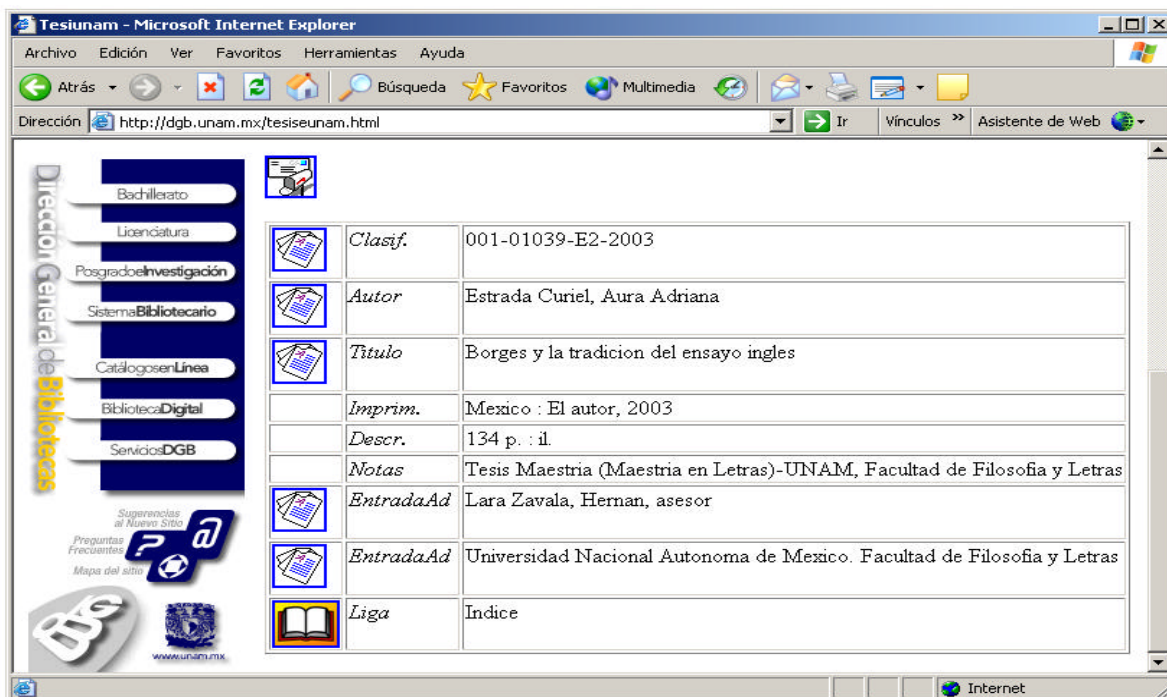
```
TDF 0600000 8567 L $$xhttp://132.248.9.25:8080/tesdig/ips/entrada.jsp?entrada=Procesados_2004/0600000/Indice.pdf$$nIndice
```

Cuando se tenga el registro en la base de datos y la carpeta en el servidor de WEB, entonces se podrá hacer la búsqueda de la tesis en texto completo en la página de la Dirección General de Bibliotecas (<http://dgb.unam.mx>) o en la página de la Biblioteca Central (<http://bc.unam.mx>).



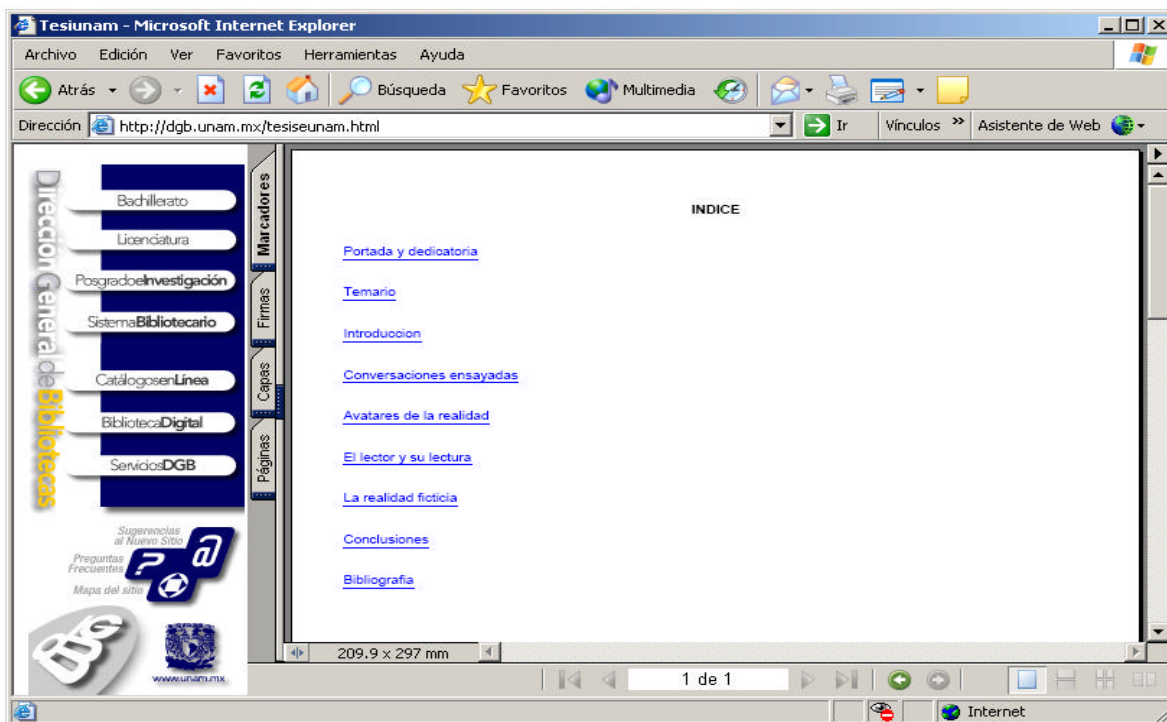
Página de la Dirección General de Bibliotecas

Desde esas dos páginas se puede hacer la consulta y por lo tanto la recuperación del texto completo. La siguiente imagen nos muestra el registro de una tesis.



Recuperación del registro

Al dar click en la imagen *Liga* (Índice) se abrirá el índice en formato PDF.



Índice.pdf



Como se ve, el índice en PDF tiene las ligas a cada uno de los apartados de la tesis.

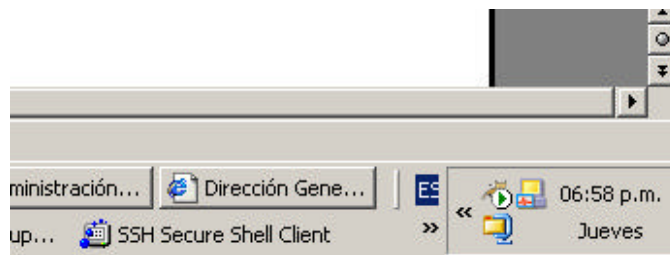
## 6.6 Pruebas.

Realizamos una serie de pruebas del sistema para estar seguros de que cada uno de sus módulos cumple con los requerimientos de forma independiente y se integran correctamente respetando las restricciones de hardware y software.

La primer prueba realizada fue la de módulo, la cual, como se mencionó en el capítulo 2, verifica cada componente o módulo de manera aislada. A continuación se detallan los módulos probados y los resultados obtenidos.

**Prueba del DBMS.** Se realizaron algunas de pruebas para verificar el funcionamiento de la base de datos exclusivamente. Se conectó remotamente a través del cliente de SQL Plus en distintas máquinas y en los sistemas operativos Sun Solaris 8 y Windows (98, XP y 2000) exitosamente.

**Prueba del Servidor Web.** Se verificó que el servidor estuviera activo. El icono que aparece en la barra de herramientas es señal de que el servidor esta activo. Otra forma es abrir un browser y verificar que esta activo.



### **Pruebas de la Interfase Gráfica.**

**Módulo de Alumnos.** Se ingresaron, modificaron y eliminaron varios grupos de alumnos y carreras los cuales fueron validados exitosamente evitando meter rfc's repetidos y validando las condiciones de no nulo. La información se almacenó correctamente en la base de datos sin que esta presentara ningún retardo. Se formaron grupos de alumnos y sus respectivas carreras que corresponden a varias tesis, y el sistema evitó de manera adecuada las repeticiones de estos en un mismo

grupo. Además se realizaron las búsquedas y filtrados de la tabla de alumnos sin ningún problema.

**Módulo de Asesores.** En este módulo también se manipuló exitosamente toda la información relacionada con los asesores, logrando hacer inserciones, actualizaciones, borrados y búsquedas, y cumpliendo con las restricciones.

**Módulo de Tesis.** Se comprobó correctamente todas las funciones de este módulo. Se probó agregar, modificar y eliminar registros de la tesis y sus subtítulos, los cuales fueron almacenados de forma adecuada en la base de datos. También insertó automáticamente los registros en las tablas de intersección, creadas para romper las relaciones de muchos a muchos entre alumnos, asesores, tesis y carreras, de la forma esperada.

**Módulo de Capítulos y Subtemas.** Aquí se comprobó el buen funcionamiento de las dos formas posibles de ingreso de la información, las cuales fueron la de captura directa y la de importación desde archivo. En la segunda forma probamos varios casos no válidos, los cuales fue capaz de detectar el sistema y reaccionar en consecuencia, que la mayoría de las veces resultó en mensajes de información para el usuario.

**Módulo de Revisión.** Las funciones principales de este módulo son mostrar un reporte global de toda la información de la tesis capturada y generar el índice en formato pdf. La primer función se ejecutó sin mayor problema, ya que es simplemente un reporte, y la segunda generó los índices en formato pdf tal como se esperaba.

**Módulo de Archivos.** El sistema logró copiar y encriptar satisfactoriamente los archivos de la tesis en texto completo en formato pdf.

También se verificó que los procesos almacenados funcionaran correctamente y que generaran entre otras cosas el formato MARC 21 verificando que su formato fuera el correcto. Para lo anterior la gente del Departamento de Tesis se encargó de revisar los archivos que el sistema genera con formato MARC 21. Se hicieron todas las correcciones que se pidieron en cuanto a formato y se volvió a probar. Al final se llegó a la conclusión de que el formato MARC 21 que genera la interfaz gráfica es el correcto.

Las pruebas de los módulos de alumnos, tesis y capítulos se realizaron tanto para la aplicación cliente de Windos como para la aplicación web.

Una vez probados todos los módulos individualmente realizamos la prueba de integración, en la cual verificamos que el flujo de la información entre los módulos descritos anteriormente fuera la adecuada. Comprobamos el proceso completo de registro de una tesis, ingresando información de alumnos, asesores, tesis, subtítulos, capítulos y subtemas con buenos resultados.

La siguiente prueba realizada fue la de funcionamiento, en la cual evaluamos que el sistema, ya integrado, estuviera cumpliendo con los objetivos y las funciones especificadas. Obtuvimos buenos resultados al poder capturar un buen número tesis de una manera mucho mas fácil y eficiente y poderlas consultar con su texto completo a través de internet.

También efectuamos la prueba de desempeño, en la cual valoramos la eficiencia del sistema al trabajar con ciertas restricciones tanto de hardware como de software. Esta se ejecutó en el sistema ya integrado con ayuda del personal del departamento de tesis, que, cabe mencionar, serán los usuarios finales del mismo, desde 15 PC's simultáneamente las cuales contaron con distintas especificaciones a partir de las mencionadas en el capítulo 3.

Concluimos con las pruebas de aceptación e instalación. En la primera, los clientes, que en este caso es el personal del departamento de tesis, dieron su aprobación al resultado obtenido, quedando satisfechos con su desempeño. Y por último se montaron los servidores web y de base de datos, así como la aplicación de las máquinas cliente sin ningún contratiempo, lo que da por concluida la prueba de instalación.

### **6.7 Mantenimiento**

El mantenimiento del sistema tuvo principio desde la realización de las primeras pruebas de los módulos. Se presentaron algunos comentarios de los usuarios en la interfaz gráfica, pero en su mayoría referentes a la apariencia o el formato de las pantallas o a la creación de algunos reportes en particular, los cuales, con la

información contenida en la base de datos no representan ninguna dificultad para su desarrollo.

Igualmente se presentaron algunos comentarios en cuanto a formato de la aplicación WEB, los cuales, se resolvieron lo antes posible. Además se planea que los usuarios manden sus sugerencias para el sitio WEB por medio de correo electrónico y así tener el sistema actualizado al máximo.

Se tienen contemplado hacer respaldos de la base de datos para mantener segura la información, los cuales, se harían semanalmente con ayuda del personal de la Dirección General de Bibliotecas. Además de mantener a personal que sea capaz de resolver cualquier problema que se presente en la base.

El contacto con el Departamento de Tesis es muy importante debido a que ellos son los que nos mantienen informados de los cambios que se les tendría que hacer al sistema en caso de que se necesite. Al igual que se debe de asesorar en el caso de que exista algún error y tenga que corregirse a la brevedad.

**CONCLUSIONES**

---

---

## **CONCLUSIONES**

En la actualidad el uso de la tecnología y, específicamente del uso de las computadoras se ha extendido a todos los campos de la actividad laboral. Las bibliotecas no han sido la excepción y mucho menos tratándose de las bibliotecas de la máxima de estudios.

Por lo tanto, gracias al apoyo de la Dirección General de Bibliotecas se ha concluido este proyecto, dejando abierta la posibilidad de continuar mejorando lo que hasta la fecha esta funcionando.

Finalmente, las conclusiones a las que hemos llegado son las siguientes:

- La implementación de Bibliotecas Digitales ha venido en incremento. La Dirección General de Bibliotecas ha formado varias bibliotecas digitales, entre las cuales se encuentra la colección de tesis digitales. Esta colección esta formada por alrededor de 40,000 ejemplares agregándose a estos cerca de 400 surgidos del SIRARTE y que han sido puestos a disposición del público. Además día a día la colección crece y la comunidad cada vez más esta adaptándose ha esta presentación de la información.
- Los tiempos se reducen en varios ámbitos. Primeramente en el Departamento de tesis de la Dirección General de Bibliotecas (DGB) se esta ahorrando el tiempo que se lleva en capturar los datos de cada tesis que se recibe en este recinto. Con el tiempo la gente que se dedica a catalogar podría dedicarse a otro tipo de actividades ya que a futuro se tiene pensado que se entreguen solamente ejemplares en formato digital. Además el contar con la aplicación WEB nos ayudará mucho a los tesisistas a hacer el registro de los datos que se necesitan para la titulación por medio de Internet, por su puesto, sin necesidad de transportarnos hasta la Biblioteca Central de la UNAM haciendo que el trámite de titulación se agilice y cada vez sea más rápido y sencillo.
- Es sabido que el espacio físico que se tiene en los Departamentos de la DGB es reducido por lo que se ven en la necesidad de ahorrar espacio. En el caso

particular del Departamento de Tesis, se esta dando el caso de que el espacio físico para almacenar las cerca de 12,000 tesis anuales que se reciben anualmente se esta acabando a pasos agigantados. Lo que nos ofrece este sistema además de ahorro monetario para ambas partes (DGB y tesis) es ahorro del espacio físico. Las tesis que se reciban en formato digital solo ocuparán espacio en algún servidor lo que a futuro se verá reflejado en un gran ahorro de espacio.

- La recuperación de los datos de cualquier tesis y del texto completo de la misma se hará con rapidez y eficacia. En este punto podemos hablar de ahorro monetario y ahorro de tiempo.
- La difusión de los trabajos recepcionales que se generan en la UNAM se hará con mayor intensidad debido a que la publicación de las tesis se hará casi de manera automática y a la brevedad posible. Estamos hablando de que una tesis puede estar dispuesta al público al siguiente día de que el tesista concluye los trámites en BC.
- El ahorro monetario que ofrece este sistema es realmente significativo y también se da en dos partes: BC y el tesista. En BC este tipo de ahorro puede darse en ahorro de personal o por que no, hasta en ahorro de estantería que es necesaria para colocar las tesis. El tesista tiene la oportunidad de ahorrarse unos pesos ya que podría por lo menos ahorrarse la impresión de dos ejemplares, sin olvidar los costos de transportación si es que requiriera consultar, o hacer algún trámite en BC.
- Finalmente se coloca a la UNAM como una institución que esta a la vanguardia en cualquier ámbito académico. Es bien sabido que la UNAM tiene una de las mejores colecciones digitales hasta el momento no solo en el país sino en América Latina.

## **TRABAJO A FUTURO**

Como trabajo a futuro se planea una aplicación WEB que sea utilizada por servicios escolares de cada facultad, instituto, institución, etc. que tenga dependencia

con la UNAM y que lo único que haga es generar un login y un password para cada tesista. Estos datos estarían en la misma base de datos que se está utilizando para las dos partes del sistema (interfaz gráfica y aplicación WEB) y servirían para acceder a la aplicación WEB en donde se capturan datos de la tesis y del tesista.

El login sería el RFC de cada tesista y el password sería una clave que tendría que ser generada para cada tesista. El RFC se localiza en la base de datos con el nombre de RFC y el password con el nombre de PASSWORD, los dos en la tabla de PERSONAS.

La aplicación va a ser necesaria si se quiere tener un sistema integral y casi totalmente automatizado.



**BIBLIOGRAFÍA**

---

---

## **BIBLIOGRAFÍA**

- Edward Yourton, Analisis Estructurado, 1989.
- Ian Sommerville, Software Engineering, 6th Edición, 2001.
- James, Martín, Organización de las Bases de Datos
- Pressman, Roger S., Software Engineering.
- William B. Heys, Power Builder, Prentice Hall.

### **TESIS**

- Mendieta Rebollo Gabriela, Desarrollo e implementación de un sistema para la automatización del préstamo de libros, tesis y revistas del Sistema Bibliotecario de la UNAM
- Ballesteros Estrada Silvia S., Sistema Integral para el control y registro de facturas para el departamento de adquisiciones de las Dirección General de Bibliotecas de la UNAM, 2003.

### **REVISTAS**

- Revista de la Dirección General de Bibliotecas de la UNAM. "Biblioteca Universitaria". Vol. 4 Número 1, Vol. 5 Número 1 y Vol. 6 Número 2.

### **PONENCIAS**

- Departamento de Mantenimiento, Administración y Desarrollo de Sistemas. UNAM. "Desarrollo de colecciones en las Bibliotecas de la UNAM". Bibliotecas Digitales.

### **DIRECCIONES DE INTERNET**

<http://www.oracle.com>  
<http://www.powerbuilder.org>  
<http://www.support.microsoft.com>  
<http://www.sun.com>  
<http://java.sun.com>