



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"**

**SISTEMA DE SEGUIMIENTO Y CONTROL DE ESTUDIANTES
DE POSGRADO DEL INSTITUTO MEXICANO DEL
PETROLEO**

P R O Y E C T O D E T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:

**RODRIGO RAMÍREZ ORTÍZ
MARCO ANTONIO CERVANTES BLANCAS**

DIRECTOR DE TESIS

ING. RODOLFO VAZQUEZ MORALES.



MÉXICO

2007



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A nuestros padres

*Por todo el apoyo, amor y cariño que nos han brindado
Por sus consejos que nos dieron fuerza para seguir adelante en tiempos difíciles.*

Al Instituto Mexicano del petróleo

Gracias por la oportunidad que nos dio para nuestro desarrollo profesional.

A la Universidad Nacional Autónoma de México

*Y a todos nuestros profesores gracias por todo
lo que nos enseñaron, su paciencia, su apoyo y consejos.*

A:

*Lili Santiago Monzón y Esperanza Sánchez Martínez
por la confianza, el apoyo, el tiempo dedicado para la realización
de este trabajo, por su amistad y sus conocimientos.*

*Al Área de la Coordinación Académica
y
Al área de Becas del IMP*

*Lic. Gerardo, Lic. Reza, Lic. Ángel , Contadora Ma Luisa,
Lorena, Pepe, Luís, Cecy , Imelda, Licha, Manuel
Por el apoyo, amistad y compañerismo brindado.*

A:

*Luis, Sergio, Paco, Nancy e Irma por sus palabras de aliento y apoyo para
la conclusión de este trabajo.*

Gracias a todos por enseñarme un mundo nuevo.

INTRODUCCION	1
CAPITULO 1. Elementos de un Sistemas de Información	
1.1 Sistemas de Información	4
1.1.1 Tipos y usos de los sistemas de información	5
1.1.1.1 Características sistemas transaccionales	6
1.1.1.2 Características sistemas de apoyo a las decisiones	6
1.1.1.3 Características sistemas estratégicos	6
1.1.1.4 Sistemas integrales de administración	7
1.1.2 Tendencias futuras	7
1.2 Bases de datos	8
1.2.1 Definición de base de datos	8
1.2.2 Tipos de bases de datos	9
1.2.3 Funciones de las bases de datos	9
1.2.3.1 Conocimientos necesarios	9
1.2.4 Sistema manejador de base de datos (DBMS)	11
1.2.5 Administrador de Base de datos (DBA)	12
1.2.6 Modelos de base de datos	12
1.2.6.1 Modelo jerárquico	12
1.2.6.2 Modelo de red	13
1.2.6.3 Modelo relacional	15
1.2.6.4 Modelo Orientado a Objetos	15
1.2.6.5 Bases de datos distribuidas	15
1.2.7 Data Warehouse	16
1.2.7.1 Arquitectura de un data wherehouse	16
1.2.7.2 Elementos constituyentes de una arquitectura data ware house	17
1.2.7.3 Operaciones de un dataware house	17
1.2.8 Minería de datos	19
1.2.8.1 Fases de un proyecto de minería de datos	19
1.2.8.2 Fases de un proyecto de minería de datos: filtrado de datos	19
1.2.8.3 Fases de un proyecto de minería de datos: Selección de Variables	19
1.2.8.4 Fases de un proyecto de minería de datos: algoritmos de extracción de conocimiento	20
1.2.8.5 Fases de un proyecto de minería de datos : interpretación y evaluación.	20

CAPITULO 2.Herramientas CASE. Ingeniería de software asistida por computadora	
2.1 Crisis del software	22
2.2 Ingeniería de software	23
2.3 Paradigmas de la ingeniería de software	24
2.3.1 Modelo en cascada	24
2.3.2 Modelo de prototipos	25
2.3.2.1 Ventajas del uso de prototipos	25
2.3.2.2 Desventajas del uso de prototipos	25
2.3.3 Modelo en espiral	26
2.3.3.1 Ventajas del modelo espiral	26
2.3.3.2 Desventajas del modelo espiral	26
2.3.4 Técnicas de cuarta generación (T4G)	27
2.3.5 Combinación de paradigmas	28
2.4 Ingeniería de software asistida por computadora	29
2.4.1 Objetivos de una herramienta CASE	29
2.4.2 Clasificación de las herramientas CASE	30
2.4.2.1 De acuerdo al grado de integración	30
2.4.2.2 De acuerdo a las fases del ciclo de vida que abordan	30
2.4.2.3 De acuerdo a su funcionalidad	30
2.5 Herramienta CASE Oracle Developer	31
2.5.1 Características	32
2.5.2 Ventajas	33
2.5.3 Desventajas	34
2.5.4 Viabilidad de la herramienta Oracle Developer	34
2.6 Componentes de Oracle Developer	35
2.6.1 Forms	36
2.6.1.1 Módulo de pantalla	36
2.6.1.1.1 Disparadores	36
2.6.1.1.2 Bloques de datos y relaciones	37
2.6.1.1.3 Elementos	37
2.6.1.1.4 Lienzos y ventanas	37
2.6.1.1.5 Menús contextuales, alertas y editores	38
2.6.1.1.6 Grupos de registro y listas LOV	38
2.6.1.1.7 Objetos reutilizables	38
2.6.1.2 Forms: Modulo de menú	39
2.6.2 Reports	39
2.6.3 Graphics	40
2.6.4 Introduccion a PL/SQL	42
2.6.4.1 PL/SQL y el tráfico de red	42
2.6.4.2 Características de PL/SQL	43

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON

2.6.4.2.1 Estructura de bloque	43
2.6.4.2.2 Tratamiento de errores	43
2.6.4.2.3 Variables y tipos	43
2.6.4.3 Entorno de desarrollo y ejecución	44
2.6.4.4 PL/SQL en el servidor	45
2.6.4.5 PL/SQL en el cliente	45
CAPITULO 3. Ciclo de vida de la base de datos	
3.1 Estudio Inicial de la Base de Datos	47
3.1.1 Analizar la situación de la compañía	48
3.1.2 Definición de Problemas y restricciones	50
3.1.3 Definición de los objetivos	50
3.1.4 Determinación del alcance y límites de la base de datos	50
3.2 Diseño de la base de datos	51
3.2.1 Modelo Conceptual	51
3.2.2 Selección del software SGBD	54
3.2.3 Diseño lógico	56
3.2.4 Reglas de Normalización	57
3.2.5 Diseño físico	59
3.2.5.1 Organización de archivos	60
3.2.5.2 Tipos de organización	60
3.3 Ejecución y carga	61
3.4 Etapa de Pruebas y evaluaciones	62
3.5 Operación	63
3.6 Mantenimiento y Evolución	63
CAPITULO 4. Sistema de seguimiento y control de los estudiantes de posgrado del IMP.	
4.1 Estudio Inicial de la Base de Datos.	66
4.1.1 Definición del problema y restricciones.	72
4.1.2 Definición de los objetivos, alcances y límites.	72
4.2. Diseño de la Base de Datos	73
4.2.1. Modelo conceptual	73
4.2.2 Diseño Lógico.	77
4.3 Diseño Físico, Ejecución y Carga.	83
CAPITULO 5. Implementación y mantenimiento del sistema	
5.1 Diseño de Formas.	94
5.2. Menús	102
5.3 Consultas	102
5.4 Formas	102
5. 5 Catálogos	103
5.6 Control	104

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGON

5.7 Ventana	104
5.8 Barra de menús de Oracle Developer	107
5.9 Pantallas de usuario	108
5.10. Programa de Trabajo	137
5.11. Requisitos de Hardware	138
5.12 Realización de pruebas	138
5.13. Guía de instalación de Oracle Developer (cliente)	140
5.14. Mantenimiento del sistema	144
5.15 Pruebas de satisfacción del cliente	145
Conclusiones	147
Bibliografía	149

INTRODUCCIÓN

La creciente demanda de las empresas de recursos humanos calificados en tareas de investigación ha generado un constante traslado de recursos de la academia a la industria, que en los países más ricos se refleja en los problemas de reclutamiento y renovación de las plantas de profesores en las universidades. Esta situación los ha obligado a desarrollar políticas y a encauzar recursos para la formación de plantas académicas.

En México el desarrollo del Posgrado ha tenido un lento proceso de crecimiento y consolidación. Sin duda, se cuenta con muchos programas con calidad comparable a nivel internacional, sin embargo se observa una gran heterogeneidad entre los programas, así como entre las instituciones. Por otra parte, los programas se encuentran desvinculados entre sí y su relación con las actividades de investigación no se ha logrado establecer en muchas instituciones.

La presencia del Instituto Mexicano del Petróleo, ha significado la conformación de una opción para el desarrollo de investigación en las disciplinas relacionadas con la industria, así como la generación de servicios tecnológicos.

La decisión presidencial de dotar al Instituto Mexicano del Petróleo de la capacidad para otorgar grados académicos a nivel de Posgrado, constituye un paso importante en la tarea de establecer las bases para vigorizar el sistema de educación superior, mejorar la capacidad del país para atender las necesidades de la industria, en particular de la industria petrolera, y fortalecer los sistemas institucionales de innovación del país.

Al crearse el área de Posgrado en el Instituto Mexicano del Petróleo se hace indispensable la creación de un sistema que permita administrar los recursos tanto humanos como materiales dentro del área, un sistema amigable con el usuario y que facilitara el trabajo de captura de los aspirantes y almacenara la información de los alumnos, es por ello que se decide crear el Sistema de Seguimiento y Control de los Estudiantes de Posgrado del Instituto Mexicano del Petróleo.

Con el desarrollo del sistema se pretende minimizar las horas hombre que se dedican al registro de cada aspirante en el área de la Coordinación Académica, el manejo de la información para las evaluaciones y la programación de las entrevistas, se llevara un control de las actividades de cada aspirante y alumno durante el desarrollo de cada proceso, de los profesores, materias, bibliografía, calificaciones, etc.

El trabajo se presenta dividido en cinco capítulos:

Capitulo 1. Sistemas de Información.

En este capitulo se da una semblanza de lo que son los sistemas de información, tipos, características de cada uno de ellos, las bases de datos que son , tipos, funciones, se da una introducción a las nuevas tecnologías como son el Data Mining y el Data Warehouse.

Capitulo 2. Herramientas CASE. Ingeniería de software asistida por computadora.

En este capitulo iniciamos abordando temas de crisis del software, paradigmas de la ingeniería de software, herramientas CASE para el desarrollo de sistemas y se concluye con la documentación de la herramienta que se utilizó para el desarrollo del sistema Oracle Developer.

Capitulo 3. Ciclo de vida de las Bases de Datos.

En este capitulo se profundiza sobre lo que son las bases de datos, la metodologías, los procedimientos a seguirse en la creación de bases de datos.

Capitulo 4. Sistema de Seguimiento y Control de los Estudiantes de Posgrado del IMP.

En este capitulo se encuentra la documentación del desarrollo del sistema creado para la Coordinación Académica de la Dirección Ejecutiva de Investigación y Posgrado del Instituto Mexicano del Petróleo.

Capitulo 5. Implementación y Mantenimiento del Sistema.

En este capitulo se detalla el procedimiento de instalación del programa final a los usuarios así como las dificultades que se presentaron durante la misma, las correcciones preventivas y correctivas que se emplearon.

Capítulo I
**ELEMENTOS DE UN
SISTEMA DE INFORMACIÓN**

Los sistemas de información se originan con los usuarios, una razón para ello es que las solicitudes, de estos sistemas se originan de una necesidad de la organización que los usuarios perciben; por ejemplo, la necesidad de resolver un problema en particular, de manejar funciones rutinarias, etc.

1.1. SISTEMAS DE INFORMACION

Un sistema de información es un conjunto de elementos que interactúan entre si con el fin de apoyar las actividades de una empresa o negocio.

Estos elementos son de naturaleza diversa y normalmente incluyen:

- Equipo computacional, el hardware necesario para que el sistema pueda operar, características de los equipos de los usuarios finales, tipo de servidores, forma de comunicación entre las terminales y/ o los servidores de aplicación.
- Recurso humano que interactúa con el sistema de información.
- Datos o información fuente que son introducidos al sistema, son todas las entradas que este necesita para generar como resultado la información que se desea.
- Los programas que son ejecutados por la computadora y producen diferentes tipos de resultados.
- La comunicación entre el hardware y el software que facilitan la transmisión de texto, datos, imágenes y voz en forma electrónica.
- Procedimientos que incluyen las políticas y reglas de operación tanto en la parte funcional del proceso de negocio, como los mecanismos para hacer trabajar una aplicación en la computadora.

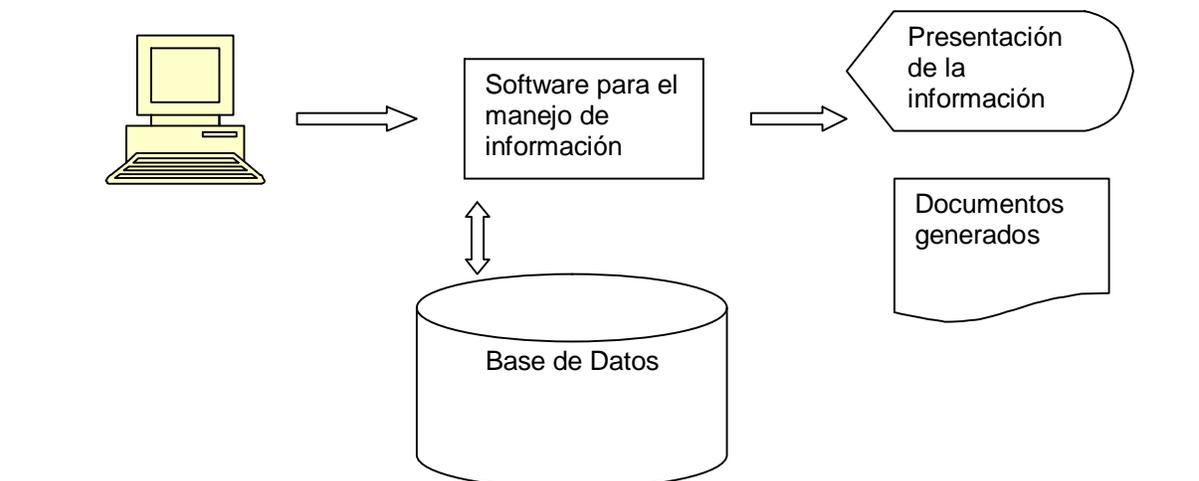


Fig 1.1 Elementos de una Base de Datos

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de información: es el proceso mediante el cual el sistema de información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que son proporcionadas en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

Almacenamiento de Información: es una de las actividades o capacidades más importantes que tiene una computadora, ya que por medio de esta actividad el sistema puede recordar la información guardada en la sesión o proceso anterior.

Procesamiento de la información: es la capacidad del sistema para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Los cálculos pueden efectuarse con datos introducidos recientemente o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones.

Salida de información: es la capacidad del sistema de información para sacar la información procesada o bien datos de entrada al exterior.

Como puede observarse un sistema de información permite procesar grandes volúmenes de datos de forma rápida, lo que retribuye en menor tiempo de respuesta para la empresa y una mejor toma de decisiones por parte de los directivos de la organización.

1.1.1 Tipos y usos de los sistemas de información

Los sistemas de información cumplirán tres objetivos básicos dentro de las organizaciones:

- Automatizar los procesos operativos
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones
- Lograr ventajas competitivas a través de su implantación y uso.

Los sistemas de información que logran la automatización de procesos operativos dentro de una organización son llamados sistemas transaccionales, ya que su función principal consiste en procesar transacciones tales como pagos, cobros, pólizas, etc. Los sistemas de información que apoyan el proceso de toma de decisiones son los sistemas de apoyo a la toma de decisiones (DSS), sistema para la toma de decisiones de grupo (GDSS), sistemas expertos de apoyo (EIS).

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

El tercer sistema, de acuerdo con su uso u objetivos que cumplen, es el de los sistemas estratégicos, los cuales desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de la información.

1.1.1.1 Características sistemas transaccionales:

- Suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Son el primer tipo de sistemas de información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo, para continuar con los mandos intermedios, y posteriormente, con la alta administración a medida que evolucionan.
- Muestran una intensa entrada y salida de información; sus cálculos y procesos suelen ser simples y poco complejos. Requieren mucho manejo de datos para poder realizar sus operaciones y como resultado entregan grandes volúmenes de información.
- Tienen la propiedad de ser recolectores de información, a través de ellos se cargan las grandes bases de información para su posterior utilización. Estos sistemas son los encargados de integrar una gran cantidad de la información que se maneja en la organización.
- Son fácilmente adaptables a paquetes de aplicación que se encuentran en el mercado, ya que automatizan los procesos básicos que son similares o iguales en otras organizaciones.

1.1.1.2 Características sistemas de apoyo a las decisiones:

- Suelen introducirse después de haber implantado los sistemas transaccionales.
- La información que generan sirve de apoyo a lo mandos intermedios y superiores para la toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información.
- Suelen ser sistemas interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.

1.1.1.3 Características sistemas estratégicos:

- Su función principal no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse in house, dentro de las organizaciones, por lo tanto pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Su forma de desarrollo se basa en incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van generando nuevas funciones o procesos.
- Las ventajas que se logran a través de estos sistemas no son eternas, existe un período de vigencia similar al tiempo que tardan los competidores en alcanzar las diferencias o ventajas obtenidas del sistema estratégico (SIE).

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

1.1.1.4 Sistemas integrales de administración.

Un sistema de la naturaleza de SAP R/3 se puede considerar obviamente como transaccional, pero también tiene un sentido estratégico, pues un sistema integral de administración puede ofrecer servicios o productos diferenciados al cliente, lo cual se puede convertir en la base de competencia favorable para la compañía que posea este recurso de información.

Tipos de Sistemas de Información	Sistemas transaccionales	Llevan acabo las actividades cotidianas de la empresa. Los procedimientos estándares de operación que facilitan el manejo de las transacciones, se incluyen, en general en los programas de cómputo que controlan la entrada de datos, el procesamiento de los detalles , almacenamiento y presentación de datos, como de información.
	Sistemas de apoyo a las decisiones	Tienen como finalidad ayudar a los directivos que enfrentan problemas de decisión únicos (no recurrentes)
	Sistemas estratégicos	Están diseñados in house, son sistemas que se desarrollan para procesos específicos dentro de la organización
	Sistemas integrales de administración	Están diseñados para dar soporte a todos aquellos asuntos donde es necesario tomar decisiones y que se presentan con frecuencia, es posible estudiar todas las variables y factores de decisión con la finalidad de desarrollar datos que contengan la información más útil para la toma de futuras decisiones.

1.1.2. Tendencias futuras

La tecnología será parte del trabajo en equipo de las empresas. Esta tecnología será utilizada para reducir el trabajo, mejorar la calidad, dar mejores servicios a los clientes o para cambiar la forma en que se trabaja. Se incrementara el uso de la tecnología multimedia principalmente en la educación.

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Esta tecnología incluye una combinación de texto, gráficas, sonido, video y animaciones. La multimedia ofrece la oportunidad de un aprendizaje interactivo capaz de mostrar una variedad de información. Las organizaciones cambiarán a la arquitectura cliente-servidor, a la vez que los usuarios trabajarán con computadoras (clientes) conectadas en red a un servidor.

1.2. BASES DE DATOS.

En el punto 1.1 se mencionó que una de las actividades y parte importante de los Sistemas de Información es el almacenamiento de la información, y la mejor manera de realizar ésta actividad es a través de una Base de Datos.

1.2.1. Definición de base de datos

De manera simple una Base de Datos es un conjunto de datos relacionados entre sí, que son almacenados de manera ordenada permitiendo un acceso a la información de forma más rápida y eficientemente, así como también evita la redundancia de la información.

Las bases de datos proporcionan la infraestructura requerida por los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en la Base de Datos de la organización para el proceso de toma de decisiones o para lograr ventajas competitivas.

Los sistemas transaccionales o los sistemas estratégicos (SIS) son los encargados de recolectar la información que contendrá la Base de Datos, por medio de las funciones de creación, bajas o modificación de la información. La forma de operar estos sistemas puede ser batch, o en línea, lo cual depende de la manera en que se actualice la información de la Base de Datos.

Por otro lado un sistema de Base de Datos tiene cuatro componentes principales: datos, hardware, software y usuarios.

Datos: son la Base de Datos propiamente dicha. Una Base de Datos se constituye con datos almacenados y utilizados por los sistemas de una organización en particular.

Software: esta constituido por un conjunto de programas que se conoce como sistema manejador de Base de Datos (DBMS Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la Base de Datos.

Usuarios: existen tres clases de usuarios relacionados con una Base de Datos

1. Programador de aplicaciones, quien crea programas de aplicación que utiliza la Base de Datos.
2. Usuario final quien accesa a la Base de Datos por medio de un lenguaje de consulta o programas de aplicación.

3. Administrador de la Base de Datos (DBA Data Base Administrator) quien se encarga del control general del sistema de Base de Datos.

1.2.2. Tipos de bases de datos

Desde el punto de vista de la organización lógica:

- a) Jerárquicas. (Progress)
- b) Relacionales. (Oracle, Access, Sybase...)

Desde el punto de vista de número de usuarios:

- a) Monousuario (dBase, Access, Paradox...)
- b) Multiusuario cliente/servidor (Oracle, Sybase...)

1.2.3. Funciones de las bases de datos

- a) Permitir la introducción de datos por parte de los usuarios (o programadores).
- b) Salida de datos.
- c) Almacenamiento de datos.
- d) Protección de datos (seguridad).
- e) Elaboración de datos.

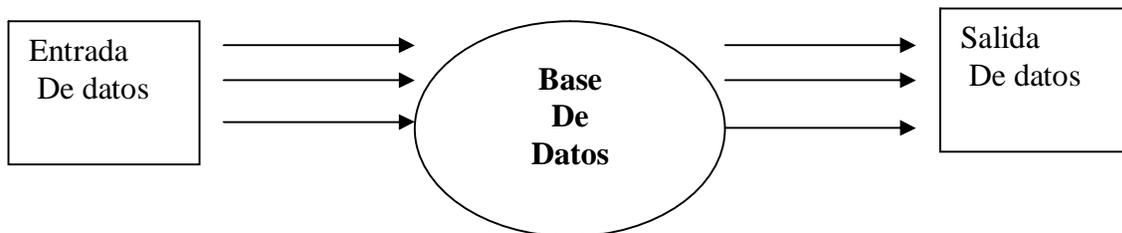


Fig. 1.2. Funciones de una Base de Datos

Básicamente, la comunicación del usuario-programador con la Base de Datos se hace a través de un lenguaje denominado SQL: Structured Query Language (Lenguaje estructurado de consultas)

1.2.3.1. Conocimientos necesarios:

Para un programador de bases de datos el conocimiento mínimo debe ser de:

- Conocimiento básico de las estructuras internas del DBMS (Oracle en nuestro caso).
- Lenguaje SQL
- Utilidades básicas: (SQL*Plus, Export, Import...)
- PL/SQL
- Tareas simples de administración
- Tuning básico de SQL.

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Tareas más propias de un administrador de bases de datos pueden ser:

- Los conocimientos propios de un programador de bases de datos.
- Conocimiento profundo de estructuras internas de Oracle.
- Conocimiento profundo de los catálogos de sistema.
- Utilidades de administración (SQL*DBA, Server Manager...)
- Tuning avanzado de SQL, red, memoria, discos, CPU...

Una Base de Datos consta de varios componentes:

- Motor:** el programa ejecutable que debe estar en memoria para manejar la Base de Datos. Cuando este programa está ejecutándose se dice que la Base de Datos está levantada (startup), en caso contrario se dice que la Base de Datos está bajada (shutdown).
- Servicio de red:** Es un programa que se encarga de establecer las conexiones y transmitir datos entre cliente y servidor o servidor y servidor.
- Listener (escuchador):** Es un programa residente en memoria que se encarga de recibir las llamadas que llegan a la Base de Datos desde la red, y de pasárselas a esta. Una Base de Datos que no tenga un listener cargado, no podrá recibir llamadas remotas. El listener se comunica con el servicio de red.
- Utilidades:** Programas de utilidad como pueden ser:
 - Intérpretes de consultas.
 - Programas de administración de Base de Datos.
 - Programas de copia de seguridad.
 - Monitores de rendimiento.

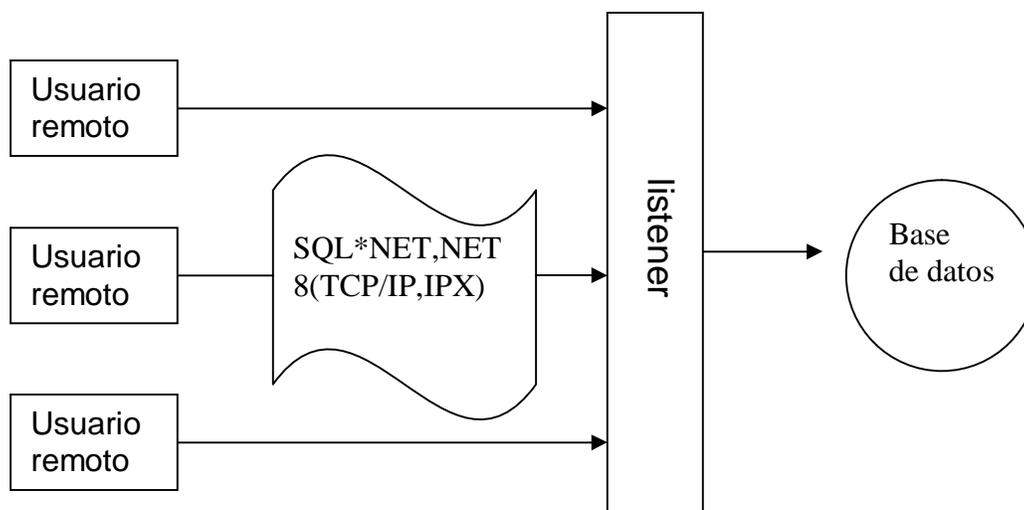


Fig. 1.3 Componentes de una Base de Datos

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

A todo el conjunto de la Base de Datos se le denomina RDBMS: Relational DataBase Manager System: Sistema de gestión de bases de datos relacionales. El primer fabricante en diseñar un RDBMS fue IBM, aunque fue Oracle, en 1979, la primera empresa hacer una implementación comercial de un sistema de bases de datos relacionales.

1.2.4. Sistema Manejador de Base de Datos (DBMS)

EL DBMS es un conjunto de programas que se encarga de manejar la creación y todos los accesos a la Base de Datos. Se compone de un lenguaje de definición de datos (DDL Data Definition Lenguaje), de un lenguaje de consulta (SQL Structured Query Lenguaje) y un lenguaje de manipulación de datos (DML Data Manipulation Lenguaje).

El lenguaje de definición de datos (DDL) es utilizado para describir todas las estructuras de información y los programas que se usan para construir, actualizar e introducir la información que contiene una base de dato. El DDL contiene un diccionario de datos que se utiliza para almacenar y crear las definiciones de los datos, incluyendo localización, forma en que se almacenan y algunas otras características. Este lenguaje de datos debe permitir describir los datos y las estructuras de los archivos del sistema, especificando la forma en que serán agrupados en registros o divididos en campos. Una vez que se ha elaborado la definición de la Base de Datos, el DBMS se encarga de construir y generar las estructuras de información de manera automática.

El lenguaje de manipulación de datos (DML) es utilizado para escribir programas que crean, actualizan, y extraen información de las Bases de Datos. A pesar de que el DBMS proporciona gran ayuda al programador, en ocasiones es necesario escribir programas para extraer datos dando respuesta a requisiciones especiales.

El lenguaje de consulta (SQL) es empleado por el usuario para extraer información de la Base de Datos. Este lenguaje permite al usuario hacer requisiciones de datos sin tener que escribir un programa. La secuencia conceptual de operaciones que ocurren para acceder a cierta información que contiene la Base de Datos es la siguiente:

1. El usuario solicita cierta información contenida en la Base de Datos
2. El DBMS intercepta este requerimiento y lo interpreta.
3. El DBMS realiza las operaciones necesarias para acceder y / o actualizar la información solicitada.

Una de las ventajas del DBMS es que puede ser invocado desde programas de aplicación que pertenecen a sistemas transaccionales escritos en algún lenguaje de alto nivel, para la creación o actualización de las bases de datos o bien para efectos de consulta a través de lenguajes propios de las bases de datos o lenguajes de cuarta generación.

1.2.5. Administrador de la base de datos (DBA)

El DBA es la persona encargada de definir y controlar las bases de datos corporativas, además proporciona asesoría a los usuarios y ejecutivos que la requieran. Sus funciones incluyen:

- Apoyar y asesorar durante la adquisición del DBMS, tanto del paquete corporativo como los paquetes que servirán de herramientas para usuarios finales que deseen crear sus propias bases de datos.
- Definir la información que contendrán las bases de datos corporativas.
- Mantener la relación y comunicación estrecha con los especialistas del DBMS, que suelen laborar con el proveedor que vendió el paquete que maneja la Base de Datos
- Diseñar las estructuras de almacenamiento y estrategias de acceso a la Base de Datos.
- Atender y servir como punto de enlace entre los usuarios de la organización, asegurando que las necesidades de información de los diferentes usuarios se encuentren contenidas en las Base de Datos.
- Definir estándares y procedimientos para respaldar y recuperar la información que contienen las Base de Datos.
- Proporcionar asesoría técnica a analistas y programadores que se encuentran desarrollando aplicaciones que crean y/o accesan la Base de Datos.

1.2.6. Modelos de base de datos

Existen cuatro diferentes modelos para la estructuración y organización de la información en una Base de Datos y estos son: el modelo jerárquico, el modelo en red y el modelo relacional y orientado a objetos.

1.2.6.1 Modelo Jerárquico.

La forma de esquematizar la información se realiza a través de representaciones jerárquicas o relaciones padre/hijo, de manera similar a la estructura de un árbol. Así el modelo jerárquico puede representar dos tipos de relaciones de uno a uno y relaciones de uno a muchos. El primer tipo que existe una relación uno a uno si el padre de la estructura de información tiene un solo hijo y, viceversa, si el hijo tiene un solo padre. En el segundo tipo se dice que la relación es de uno a muchos si el padre tiene más de un hijo, aunque cada hijo tenga un solo padre.

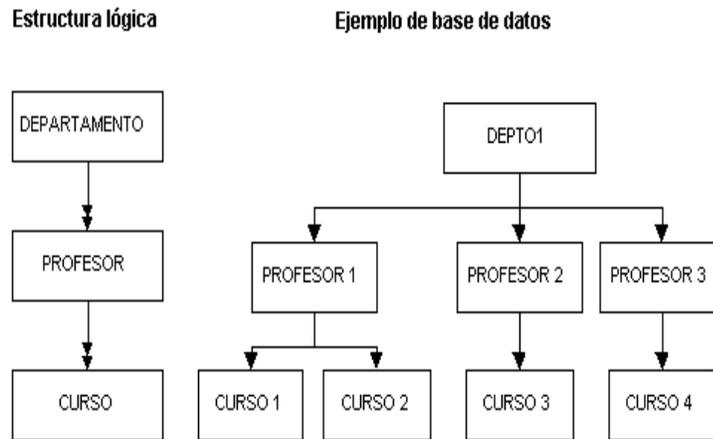


Figura 1.4 Base de datos jerárquica. Estructura lógica y ejemplo

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que son las propiedades o atributos de las entidades. Las relaciones entre entidades están representadas por las ramas. En la Figura 1.4 cada departamento es una entidad que mantiene una relación de uno a muchos con los profesores, que a su vez mantienen una relación de uno a muchos con los cursos que imparten.

A pesar de que la representación jerárquica es muy fácil de entender y comunicar, en la actualidad este modelo de representación de la información está dejando de emplearse por razones como por ejemplo:

- La representación de información de donde se requieran relaciones de muchos a muchos tiende a complicarse, de tal forma que si un hijo llega a tener dos o más padres, la información de este hijo debe almacenarse en varios lugares diferentes de la Base de Datos, lo cual depende de cada uno de sus padres.
- Otra dificultad que representa el modelo jerárquico de representación de datos es respecto a las bajas. En este caso, si se desea dar de baja a un padre, ello necesariamente implicaría dar de baja a todos y cada uno de los hijos que depende de este.

1.2.6.2 Modelo de Red

Una estructura de datos en red, o estructura plex, es muy similar a una estructura jerárquica, de hecho no es más que un súper conjunto de ésta. Al igual que en la estructura jerárquica, cada nodo puede tener varios hijos pero, a diferencia de ésta, también puede tener varios padres.

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Este modelo permite la representación de muchos a muchos, de tal forma que cualquier registro dentro de la Base de Datos puede tener varias ocurrencias superiores a él. El modelo de red evita la redundancia de la información, las desventajas descritas del modelo jerárquico desaparece con el modelo de red, la mayor parte de la complejidad y problemas de este modelo surge debido a la dificultad de manejar las conexiones o ligas entre los registros y sus correspondientes registros conectores.

La Figura 1.5 muestra una disposición plex. En esta representación, los nodos C y F tienen dos padres, mientras que los nodos D, E, G y H tienen sólo uno.

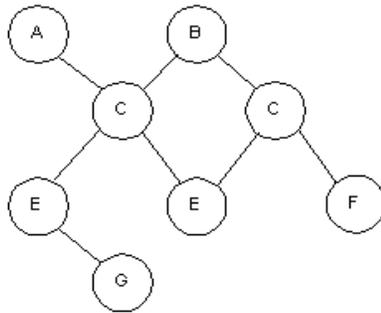


Fig.1.5 Estructura de datos de red

- El registro padre se denomina propietario del conjunto, mientras que el registro hijo se denomina miembro.
- Un conjunto está formado en un solo registro propietario y uno o más registros miembros.
- Una ocurrencia de conjuntos es una colección de registros, uno de ellos es el propietario y los otros los miembros.
- Todos los registros propietarios de ocurrencias del mismo tipo de conjunto deben ser del mismo tipo de registro.
- El tipo de registro propietario de un tipo de conjunto debe ser distinto de los tipos del registro miembro.
- Sólo se permite que un registro miembro aparezca una vez en las ocurrencias de conjuntos del mismo tipo.
- Un registro miembro puede asociarse con más de un propietario, es decir, puede pertenecer al mismo tiempo a dos o más tipos de conjuntos distintos. Esta situación se puede representar por medio de una estructura multianillo.
- Se pueden definir niveles múltiples de jerarquías donde un tipo de registro puede ser miembro en un conjunto y al mismo tiempo propietario en otro conjunto diferente.

Sus principales desventajas residen en la dificultad de crear y dar mantenimiento a las bases de datos.

1.2.6.3 Modelo relacional

El modelo relacional para una la representación de una Base de Datos se esta empleando con mas frecuencia en la practica debido a las ventajas que ofrece sobre los dos modelos anteriores.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. El usuario de las bases de datos relacionales tiene conocimiento de las tablas que están definidas y su interacción con la información contenida en la Base de Datos se reduce a operaciones lógicas que se efectúan con las diferentes tablas.

1.2.6.4 Modelo orientado a objetos

Estos modelos son utilizados con los lenguajes orientados hacia objetos, donde estos, además de guardar los datos y los procedimientos que los manipulan, contienen las relaciones con otras entidades. De aquí surgen los DBMS orientados a objetos (OODBMS Object Oriented Data Base Management System). Los OODBMS proveen interfaces gráficas para administrar el DBMS. Las bases de datos con orientación hacia objetos son muy eficaces en sistemas que usan componentes de multimedia, como los sistemas geográficos.

El diseño orientado a los objetos (DOO), al igual que otras metodologías de diseño orientadas a la información crea una representación del campo del problema del mundo real y lo hace corresponder con el ámbito de la solución, que es el software. A diferencia de otros métodos, el DOO produce un diseño que interconecta objetos de datos y operaciones de procesamiento en una forma que modularía la información y el procesamiento, en lugar de dejar aparte el procesamiento.

A veces es difícil distinguir claramente el análisis orientado a los objetos y el diseño orientado a los objetos. El análisis orientado a los objetos es una actividad de clasificación, se analiza un problema con el fin de determinar clases de objetos que sean aplicables en el desarrollo de la solución. El diseño orientado a los objetos permite indicar los objetos que se derivan de cada clase y las interrelaciones entre ellos, además de proporcionar una notación que refleje las relaciones entre los objetos.

1.2.6.5 Bases de datos distribuidas

Muchas organizaciones se encuentran distribuidas geográficamente requieren del uso compartido de la información, para ello pueden utilizarse las Base de Datos distribuidas, las cuales no están almacenadas totalmente en un solo lugar físico y se comunican por medio de enlaces de comunicaciones a través de una red de computadoras distribuida geográficamente.

CAPITULO I. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN.

Las bases de datos distribuidas se están utilizando cada vez más en la misma medida en que se usan las arquitecturas cliente-servidor y groupware.

Los principales problemas generados por el uso de la tecnología de Base de Datos distribuidas se refieren a la duplicidad de datos y a su integridad en el momento de realizar actualizaciones a los mismos.

Existen dos modalidades para crear Base de Datos distribuidas. Una alternativa es fragmentar la Base de Datos, la fragmentación consiste en instalar en cada computadora solo la parte de la Base de Datos que usará con mas frecuencia en la zona geográfica. La segunda modalidad es la réplica consiste en realizar una copia total de la Base de Datos e instalarla en cada una de las computadoras de las organizaciones

1.2.7 Data Warehouse

Según la definición de W.H. Inmon. “Un data warehouse es un conjunto de datos integrados orientados hacia una materia, que varían en el tiempo y que no son transitorios, los cuales apoyan el proceso de toma de decisiones de una administración”.

1.2.7.1. Arquitectura de un Data Warehouse

Una de las razones por las que el desarrollo de un data warehouse crece rápidamente, es que realmente es una tecnología muy entendible. De hecho, data warehousing puede representar mejor la estructura amplia de una empresa para administrar los datos informacionales dentro de la organización. A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia data warehousing, es esencial tener una Arquitectura Data Warehouse.

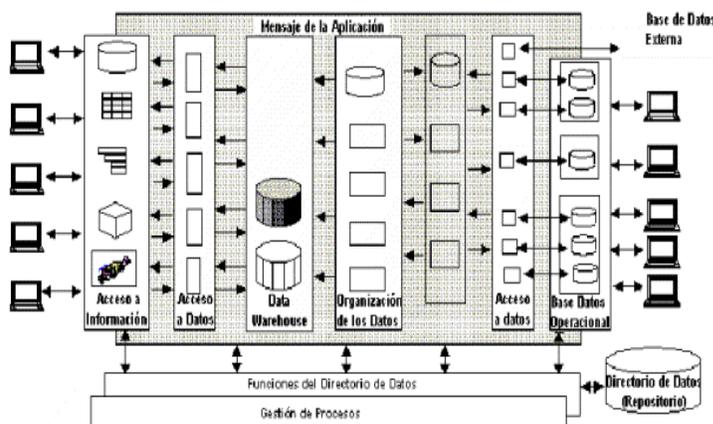


Figura 1.6 Elementos de un datawarehouse

1.2.7.2 Elementos constituyentes de una Arquitectura Data Warehouse

Una Arquitectura Data Warehouse (Data Warehouse Architecture - DWA) es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, que existe para los usuarios finales que disponen de una computadora dentro de la empresa.

La arquitectura se constituye de un número de partes interconectadas:

- Base de Datos operacional / Nivel de Base de Datos externo
- Nivel de acceso a la información
- Nivel de acceso a los datos
- Nivel de directorio de datos (Metadata)
- Nivel de gestión de proceso
- Nivel de mensaje de la aplicación
- Nivel de data warehouse
- Nivel de organización de datos

1.2.7.3 Operaciones en un Data Warehouse

En la Figura 1.7 se muestra algunos de los tipos de operaciones que se efectúan dentro de un ambiente data warehousing.

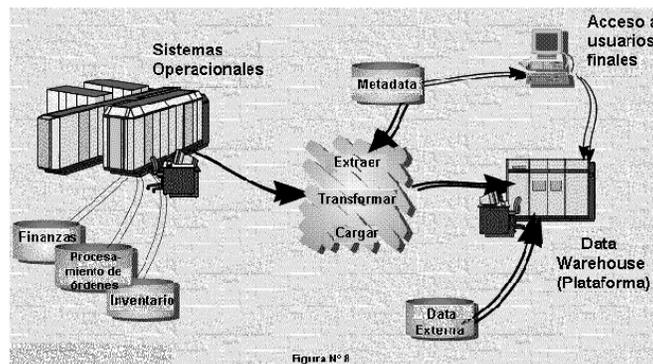


Fig. 1.7 Operaciones de un Data Where House.

- a) Sistemas Operacionales
- b) Extracción, Transformación y Carga de los Datos
- c) Metadata
- d) Acceso de usuario final
- e) Plataforma del data warehouse
- f) Datos Externos

1.2.8 Minería de Datos

Las técnicas de minería de datos se emplean para mejorar el rendimiento de procesos de negocio o industriales en los que se manejan grandes volúmenes de información estructurada y almacenada en bases de datos.

1.2.8.1 Fases de un Proyecto de Minería de Datos

Los pasos a seguir para la realización de un proyecto de minería de datos son siempre los mismos, independientemente de la técnica específica de extracción de conocimiento usada, como lo muestra la figura 1.8.

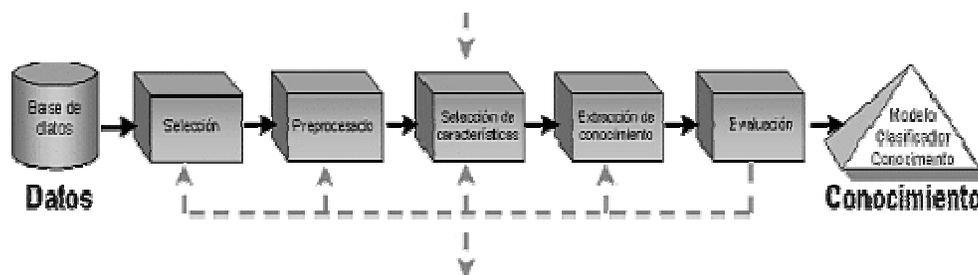


Fig. 1.8 Fases de un proyecto de Minería de Datos

El proceso de minería de datos pasa por las siguientes fases:

- Filtrado de datos
- Selección de Variables
- Extracción de Conocimiento
- Interpretación y Evaluación

1.2.8.2 Fases de un proyecto de Minería de Datos: Filtrado de datos

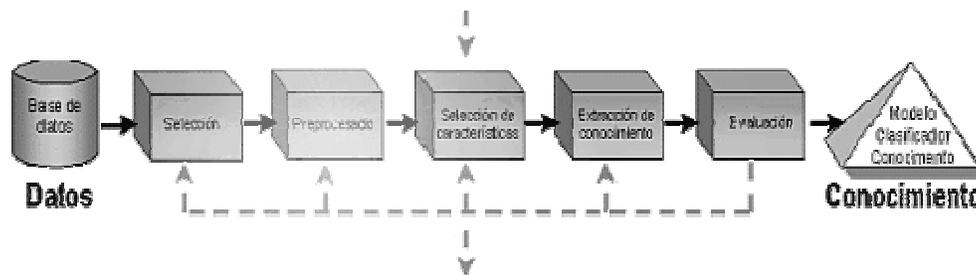


Fig. 1.9 Fases de un proyecto: filtrado de datos

El formato de los datos contenidos en la fuente de datos (Base de Datos, Data Warehouse) nunca es el idóneo, y la mayoría de las veces no es posible ni siquiera utilizar ningún algoritmo de minería sobre los datos "en bruto". Mediante el preprocesado, se filtran los datos (de forma que se eliminan valores incorrectos, no válidos, desconocidos según las necesidades y el algoritmo a usar), se obtienen muestras de los mismos (en busca de una mayor velocidad de respuesta del proceso), o se reducen el número de valores posibles (mediante redondeo, clustering,...).

1.2.8.3 Fases de un proyecto de Minería de Datos: Selección de variables

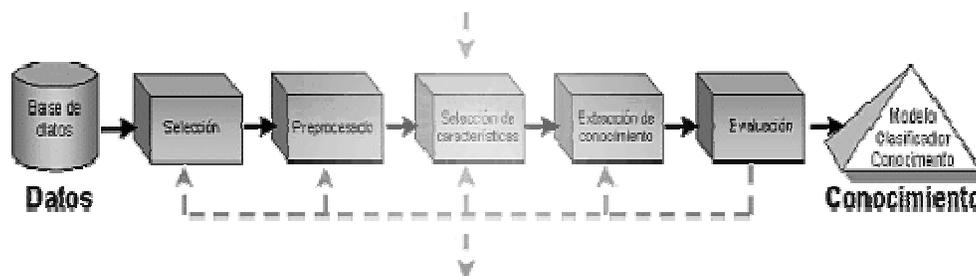


Fig. 1.10 Fases de un proyecto: Selección de variables

Aún después de haber sido preprocesados, en la mayoría de los casos se tiene una cantidad ingente de datos. La selección de características reduce el tamaño de los datos eligiendo las variables más influyentes en el problema, sin apenas sacrificar la calidad del modelo de conocimiento obtenido del proceso de minería. Los métodos para la selección de características son básicamente dos: aquellos basados en la elección de los mejores atributos del problema, y aquellos que buscan variables independientes mediante tests de sensibilidad, algoritmos de distancia o heurísticos.

1.2.8.4 Fases de un proyecto de Minería de Datos: Algoritmos de Extracción de Conocimientos

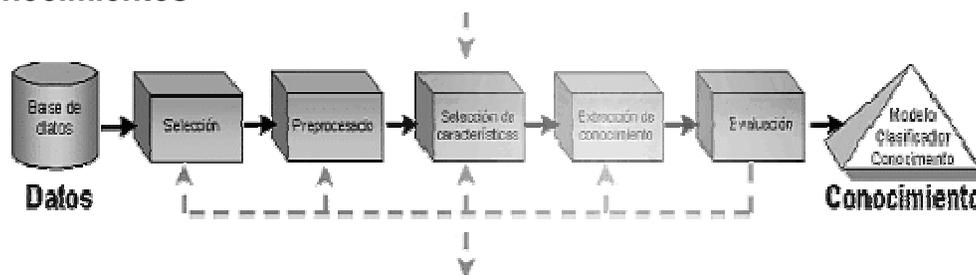


Fig. 1. 11 Fases de un proyecto de Minería de Datos Algoritmos de Extracción de Conocimiento

Mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un preprocesado diferente de los datos.

1.2.8.5 Fases de un proyecto de Minería de Datos: Interpretación y evaluación

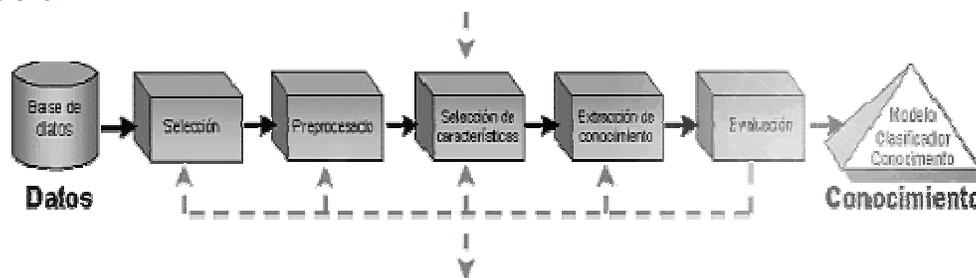


Fig. 1. 12 Fases de un proyecto de Minería de Datos: Interpretación y evaluación

Una vez obtenido el modelo, se debe proceder a su validación, comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

**HERRAMIENTAS CASE INGENIERIA DE SOFTWARE ASISTIDA POR
COMPUTADORA**

CAPITULO II.



Durante los primeros años de la computación, fue el hardware el que sufrió continuos cambios a diferencia del software que se tenía solo como un añadido al hardware. El software era desarrollado sin una planificación adecuada, y existían pocos métodos para aplicaciones específicas además de tener poca distribución.

2.1 CRISIS DEL SOFTWARE.

El software ha evolucionado, surgió la multiprogramación y los sistemas multiusuarios; así como los sistemas en tiempo de real (que eran capaces recibir, analizar y transformar datos de múltiples fuentes) y se generaron los sistemas de generación de bases de datos. La gran mayoría de las veces, el software desarrollado era utilizado por la misma persona u organización, que además lo escribía, lo ejecutaba y, si fallaba, lo depuraba. El software se estableció como un producto, con una mayor distribución en el mercado, y de una manera multidisciplinaria. En finales de los setenta y la década de los ochenta, se llegó a los sistemas distribuidos (múltiples computadoras comunicándose con otras y ejecutando funciones concurrentes).

Sin embargo, a pesar de lo mencionado anteriormente, se presentaron problemas en el desarrollo de software, tales como:

- Imprecisión en la planificación del proyecto y estimación de los costos.
- Baja calidad del software.
- El software se entrega muy a menudo con retrasos y con unos costes superiores a los presupuestados.
- El software no es fiable y necesita de un mantenimiento permanente.
- A menudo el software es imposible de mantener, tienen un diseño poco estructurado, carece de transparencia y no se puede modificar ni mejorar.

Por otra parte se exige que el software sea eficaz y barato tanto en el desarrollo como en la compra. Estas son las dificultades mas importantes que han influido para que el software haya caído en una crisis, aunque también existe otro factor determinante a tomar en cuenta, el software no lleva un crecimiento paralelo al hardware (el cual cambia a un ritmo acelerado), y esto provoca que no se aproveche todo su potencial, A continuación se muestra el informe del Government Accounting Office sobre gastos en proyectos de desarrollo software (Fig.2.1).

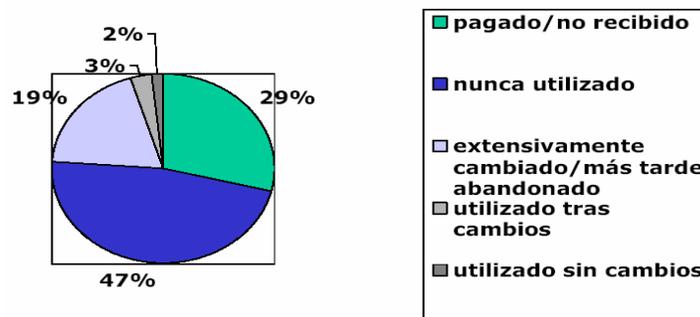


Fig.2.1 Informe del Government Accounting Office

2.2 INGENIERÍA DE SOFTWARE.

Con el objeto de dar solución a los problemas que llevaron a la crisis del software fue necesario el crear una nueva disciplina que combinara métodos para todas las fases del desarrollo del software, mejores herramientas para la automatización de estos métodos, y técnicas para proporcionar una mejor calidad de software: disciplina que se le conoce como ingeniería de software. A continuación se presentaran algunas definiciones de ingeniería de software hechas por diferentes autores, posteriormente buscaremos los puntos en común que guardan estas definiciones para poder construir una definición propia.

- “El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales.”¹
- “(1) La aplicación de un enfoque del desarrollo, la operación y el mantenimiento del software que sea sistemático, disciplinado y cuantificable; esto es, la aplicación de la ingeniería al software. (2) El estudio de tales enfoques.”²
- “La disciplina tecnológica y de gestión que concierne a la producción y el mantenimiento sistemático de productos software desarrollados y modificados dentro de unos plazos estipulados y costes estimados.”³
- “Ingeniería es la aplicación sistemática de conocimiento científico en la creación y construcción de soluciones, que satisfacen una buena relación efectividad/precio, a problemas prácticos al servicio de la humanidad. La ingeniería del software es la forma de ingeniería que aplica los principios de la ciencia de la computación y las matemáticas en la obtención de soluciones, que satisfacen una buena relación efectividad/precio, a los problemas software.”⁴

¹ (Bauer, F. L. *Software Engineering*. Information Processing 71, 1972)

² (IEEE STD 610-1990).

³ (FAIRLEY, R. *SOFTWARE ENGINEERING CONCEPTS*. NEW YORK: MCGRAW-HILL, 1985).

⁴ (SEI REPORT ON UNDERGRADUATE SOFTWARE ENGINEERING EDUCATION 1990)

2.3 PARADIGMAS DE LA INGENIERÍA DEL SOFTWARE.

A continuación se muestran los diferentes paradigmas de la ingeniería de software que se han usado o se siguen usando en la actualidad para el desarrollo de programas informáticos.

2.3.1 Modelo en cascada.

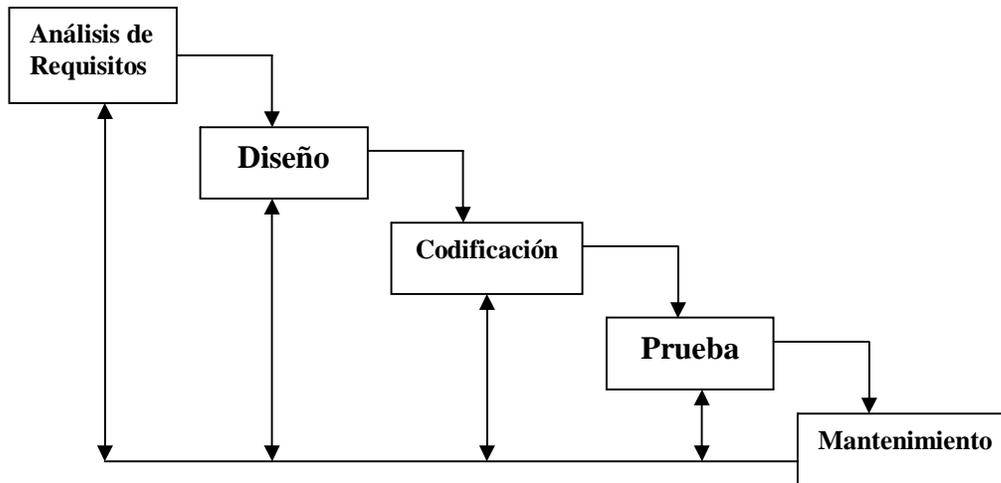


Fig.2.2 Modelo en Cascada.

Análisis de Requisitos: En esta etapa el objetivo primordial es reconocer los elementos básicos del problema tal y como lo percibe el usuario, evaluar el flujo y la estructura de la información, que funciones debe realizar el sistema, las características de la interfaz y definir las restricciones al diseño.

Diseño: En la etapa de diseño se traducen los requisitos del sistema en una representación de software, enfocándose en la estructura de los datos, la arquitectura del software, el detalle procedimental y la característica de la interfaz.

Codificación: Se traduce el diseño en una forma legible para la máquina, es decir, se genera el código fuente del software, para posteriormente obtener el software ejecutable.

Prueba: En esta etapa se realizará un proceso de ejecución para descubrir algún error dentro del sistema, de tal manera que se asegure en la medida de lo posible que se obtienen los resultados que se requieren.

Mantenimiento: En esta etapa, se le harán modificaciones al software ya que este, indudablemente, sufrirá cambios después de la entrega al cliente. Estas modificaciones ocurrirán debido a errores encontrados, a cambios del entorno

- El usuario puede creer que tiene frente a sí el sistema completo y operable.
- Posibilidad de subestimar el proyecto, ya que las salidas están pronto disponibles.

2.3.3 Modelo en Espiral.

El modelo espiral combina las mejores características del ciclo de vida clásico (modelo en cascada) y del modelo por prototipos, aunque añade un elemento que no aparece en esos dos paradigmas, el análisis de riesgo (Fig.2.4). Este modelo contiene cuatro actividades principales:

- Planificación. Determinación de objetivos, alternativas y restricciones.
- Análisis de riesgo: Análisis de alternativas e identificación/resolución de riesgos.
- Ingeniería: Desarrollo de producto de siguiente nivel.
- Evaluación del cliente: Valoración de los resultados de ingeniería.

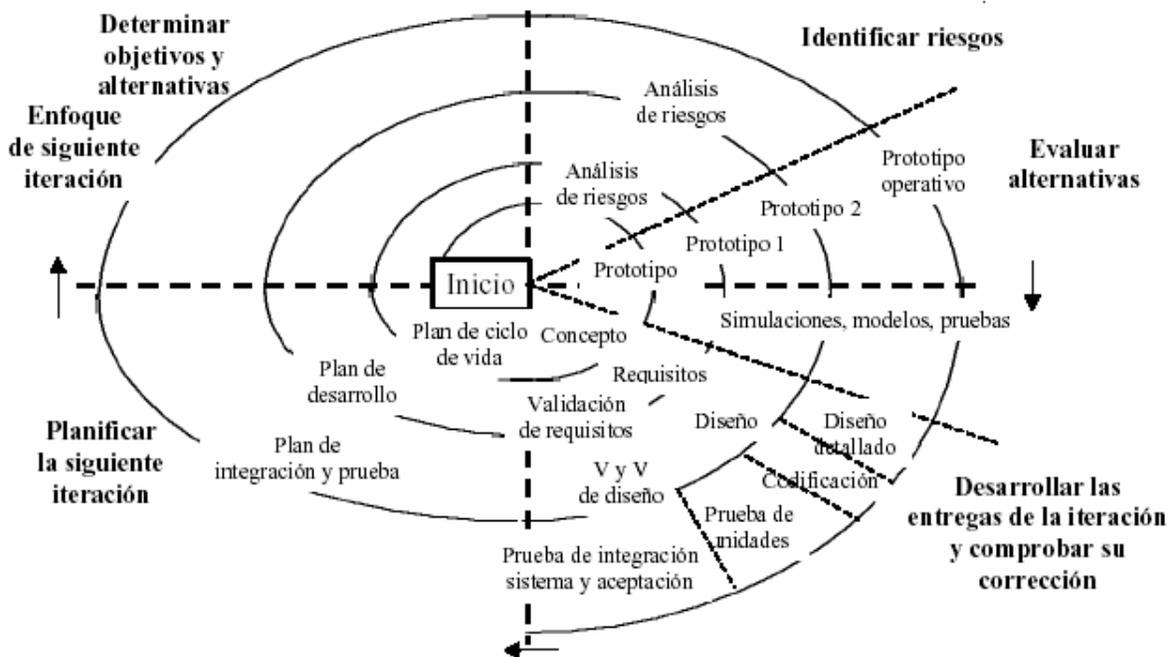


Fig. 2.4. Modelo en espiral

2.3.3.1 Ventajas del Modelo Espiral

- Utiliza un enfoque evolutivo, permitiendo una consideración directa de los riesgos en todas las etapas del proyecto.
- Permite la utilización de prototipos como un mecanismo de reducción del riesgo.
- Mantiene los pasos del modelo en cascada incorporados dentro de un marco de trabajo interactivo.

2.3.3.2 Desventajas del Modelo Espiral

- Criticidad del análisis de riesgo.

- Es difícil convencer a los clientes de que el enfoque evolutivo es controlable.
- Es necesaria una gran habilidad para el análisis de riesgos.
- Muy nuevo y desconocido, poco probado aún.

2.3.4 Técnicas de cuarta generación (T4G).

Las técnicas de cuarta generación hacen referencia a herramientas que facilitan el desarrollo de software. Herramientas que con solo especificar ciertas características de lo que queremos que realice el software, usando instrucciones sencillas, pueden generar código fuente, consultar a bases de datos a través de lenguajes procedimentales, manipulación de datos, generación de informes, definición de pantallas, facilidades gráficas de alto nivel, y hojas de cálculo. En la figura 2.5 se muestra el modelo de este paradigma.

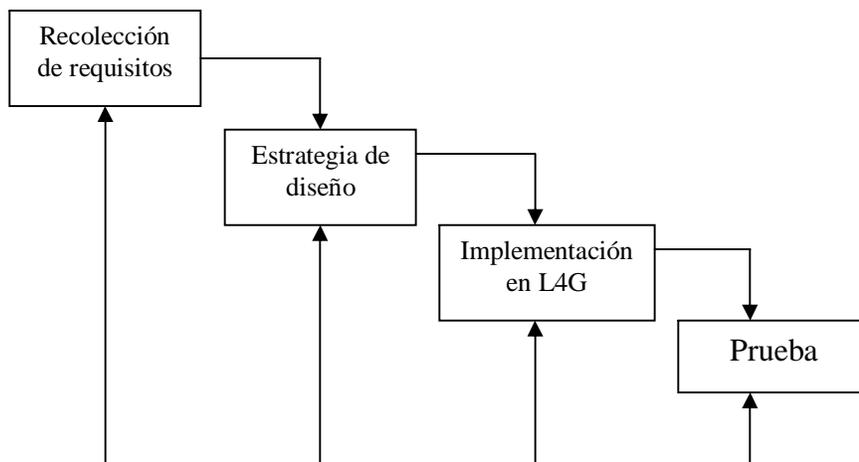


Fig. 2.5 Técnicas de cuarta generación.

Al igual que los paradigmas mencionados anteriormente, se debe empezar por la recolección de requisitos, que son traducidos a un prototipo operativo, sin embargo, aunque sea posible pasar de la recopilación de requisitos a la implementación, es necesario realizar un diseño adecuado y profundo del software que se quiere desarrollar, sobre todo cuando el proyecto es de dimensiones considerables. La implementación con lenguajes de cuarta generación permite enfocarse en la representación de los resultados, aunque para que se pueda dar esto, debe de existir una estructura de datos relevante y bien diseñada.

Posteriormente se debe hacer una prueba completa, realizar una documentación con sentido. El desarrollar un software con un lenguaje de cuarta generación deberá ser construido de manera que facilite su mantenimiento.

Sin duda, esta clase de herramientas ayudan a generar software a un ritmo acelerado, pero no se debe olvidar que siempre se requerirá de un buen análisis y

diseño para garantizar una mejor calidad en el software, ya que si no existe estos pasos, aunque tengamos la mejor herramienta de cuarta generación se pueden presentar muchas dificultades.

2.3.5 Combinación de paradigmas

Hay situaciones en las que la combinación de los paradigmas es una necesidad tomando lo mejor que brinda cada paradigma, aunque la combinación o no de los paradigmas lo dictará la naturaleza del problema, el realizar dicha combinación puede resultar en un mejor desarrollo de software. En la Fig. 2.6 se muestra la combinación de los paradigmas.

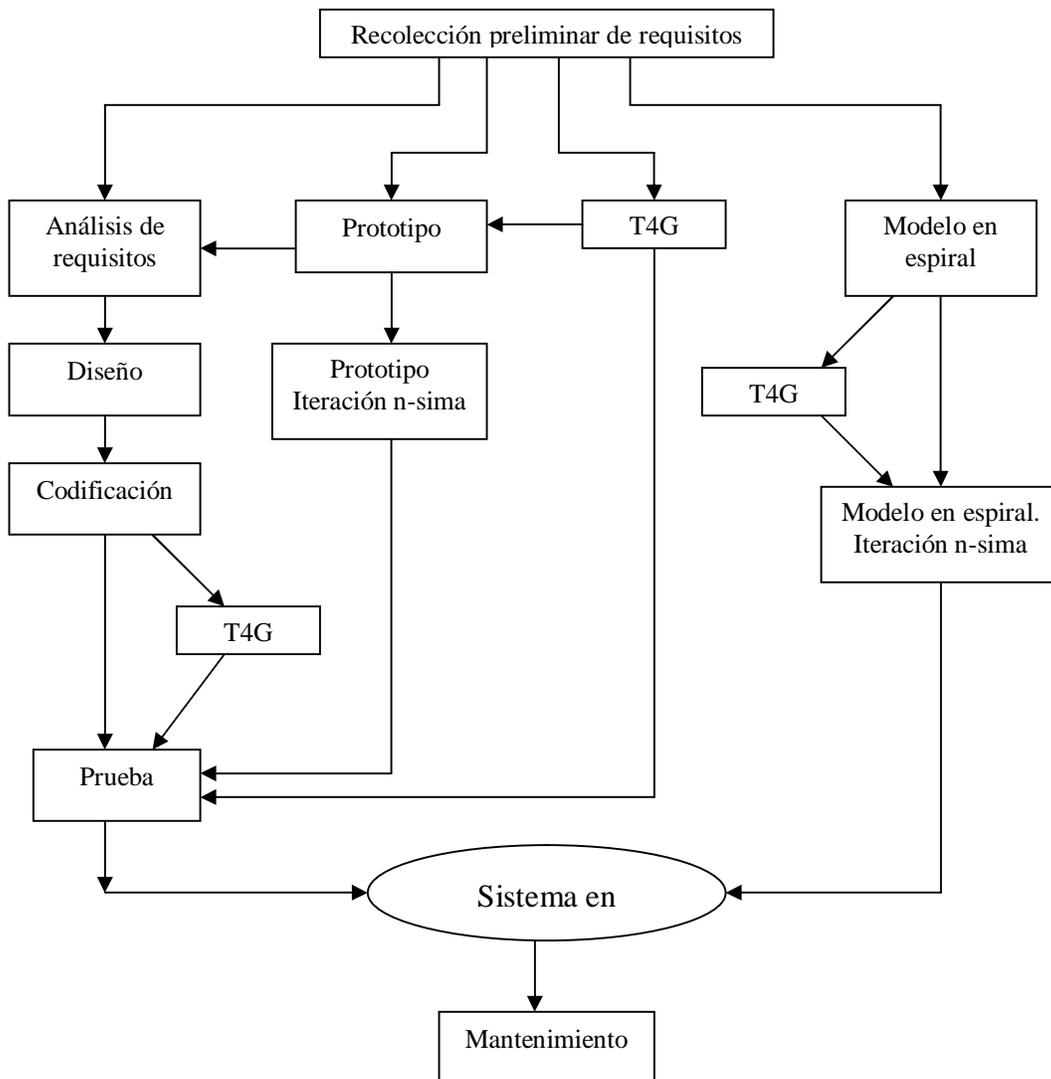


Fig. 2.6 Combinación de paradigmas.

2.4 INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADORA

“la ingeniería de software asistida por computadora tiene el potencial de llegar a ser el avance tecnológico mas importante en la historia del desarrollo del software”⁵

La tecnología CASE son herramientas y metodologías asociadas que permiten enlazar y automatizar todas o alguna de las fases del ciclo de vida del software contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información. El desarrollo de las herramientas CASE comenzó a principios de los ochenta; con la creación de herramientas que se enfocaron a las primeras etapas del ciclo de vida del software: análisis y diseño, de esta manera se logró simplificar y agilizar la documentación y construcción de diagramas, los cuales estaban basados en las metodologías existentes en ese entonces. En una segunda fase, era posible realizar una comprobación automática de los diagramas y almacenarlos en diccionarios o bases de datos llamados depósitos CASE. La siguiente meta fue la capacidad de generar código automáticamente en base a las especificaciones hechas en la etapa de análisis y diseño.

Las herramientas CASE permitieron tener una reutilización del software teniendo la oportunidad de usar los componentes que se encuentran en los depósitos CASE, teniendo acceso a los módulos de código fuente, planes de proyectos, modelos de prototipos, modelos de datos y especificaciones de diseño.

2.4.1 Objetivos de una herramienta CASE

- Permitir la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta se consigue agilizar el trabajo.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones.
- Simplificar el mantenimiento de los programas.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la reutilización de componentes software.
- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.
- Automatizar:
 - El desarrollo del software
 - La documentación
 - La generación del código
 - El chequeo de errores
 - La gestión del proyecto

⁵ Pressman S. Roger Ingeniería del software. Un enfoque práctico. Edt. Mc Graw-hill Pag. 747

- Permitir:
 - La reutilización del software
 - La portabilidad del software
 - La estandarización de la documentación

2.4.2 Clasificación de las herramientas CASE

Hoy en día hay una gran variedad de herramientas CASE que se enfocan en diferentes fases del ciclo de vida del software, y aunque no se puede determinar una clasificación precisa, se trata de ubicar diferentes grupos de herramientas de acuerdo a lo que se puede realizar con ellas.

2.4.2.1 De acuerdo al grado de Integración.

- Toolkits (Juego de herramientas): Conjunto de herramientas software que automatizan alguna fase del ciclo de vida: análisis, diseño, etc.. Integración baja.
- Workbenchs (Banco de trabajo): Conjunto de herramientas software que automatizan el ciclo de vida en más de una fase. Integración media.
- Methodology Companion (Compañero de metodología): Cubren todo el ciclo de vida basándose en una metodología.

2.4.2.2 De acuerdo a las fases del ciclo de vida que abordan.

- CASE frontales (fornt-end): Herramientas de apoyo a las primeras fases: Análisis, Diseño.
- CASE dorsales (back-end) : Herramientas de apoyo a las últimas fases: Implementación (generación de código), Pruebas, Mantenimiento.
- ICASE (Integrated-CASE): Contemplan todo el ciclo de desarrollo. Análisis, Diseño, Implementación (generación de código), Pruebas, Mantenimiento.

2.4.2.3 De acuerdo a su funcionalidad.

- Herramientas de planificación de sistemas de gestión: Ayudan a comprender como se mueve la información entre las distintas unidades organizativas.
- Herramientas de gestión de proyectos: Ayudan a la planificación, agenda de desarrollo; costes, duración y esfuerzo; control de la productividad y calidad del proyecto, así como un seguimiento completo desde los requisitos iniciales hasta el producto final.
- Herramientas de análisis y diseño: Permiten la creación un modelo del sistema a construir y la evaluación de la calidad de éstos.

CAPITULO II. HERRAMIENTAS CASE INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADORA

- Herramientas de creación de prototipos: Estas pueden crear prototipos en papel(imágenes que permiten ver el funcionamiento y comportamiento del sistema), definición de pantallas, creación de estructuras en diversos lenguajes.
- Herramientas de programación: Engloba a compiladores, editores y depuradores que usan los lenguajes de programación convencionales, la programación orientada a objetos, lenguajes de cuarta generación y generadores de aplicaciones.
- Herramientas de integración y pruebas: Analizan el código fuente sin ejecutar y otros estando en ejecución, simulación de hardware o de elementos externos o herramientas que ayudan a la planificación, desarrollo y control de las pruebas.
- Herramientas de soporte: Permiten automatizar la documentación, gestionar redes y software del sistema, controlar la calidad del software y realizar una gestión de bases de datos.
- Herramientas de mantenimiento: Permiten ingeniería inversa (tomando el código fuente y generar modelos de diseño), análisis de la sintaxis del programa, y herramientas de reingeniería (convertir de un lenguaje a otro o de un lenguaje a un modelo mas entendible).

2.5 HERRAMIENTA CASE ORACLE DEVELOPER

Oracle developer es un entorno integrado de desarrollo que permite crear aplicaciones de bases de datos en ambiente web o para sistemas cliente/servidor de una manera mas sencilla a diferencia de usar lenguajes de programación como COBOL, C, C++, o Java.

Estos tipos de lenguajes proporcionan ciertas ventajas como son la flexibilidad, control y rendimiento, pero nos presentan también desventajas como lograr que la aplicación soporte un crecimiento continuo o programar de tal manera que el software falle o vaya mas lento sin mencionar que se tiene que escribir una cantidad considerable de código.

Oracle Developer permite generar aplicaciones de bases de datos para cualquier negocio de manera fácil a través de pocas instrucciones o propiedades modificables, con esto se logra crear aplicaciones que respondan a las necesidades del usuario de manera expedita, y si los requerimientos del cliente fueran mas haya de lo que Developer nos permite realizar de manera sencilla, se resuelve con un poco de codificación adicional en PL/SQL.

2.5.1 Características

En el punto 2.4.3 mencionamos una clasificación de herramientas CASE de acuerdo a su funcionalidad, a pesar de haber expuesto otras clasificaciones desde diferentes puntos de vista, Oracle Developer encaja muy bien en la descripción de herramientas de codificación, como un generador de aplicaciones que utiliza una base de datos central, características seleccionadas mediante menús interactivos y reglas específicas de la aplicación para generar software.

Las aplicaciones desarrolladas en Oracle Developer pueden implementarse en una arquitectura cliente/servidor o tres capas, con la facilidad de que se puede mover la aplicación de una capa a otra con solo recompilar la aplicación.

Developer ofrece un entorno de desarrollo integrado, por mencionar: un navegador de objetos que permite ver un panorámica completa de nuestro proyecto ya sea que estemos trabajando con Oracle forms, reports o graphics; nos permite ver cada uno de los objetos que existen en Developer para crear las aplicaciones, desde los bloques de datos, listas de valores, lienzos, editores para insertar código PL/SQL, compiladores etc.

Contiene una biblioteca de clases que dan soporte a diferentes objetos como son ventanas, cajas de texto, casillas de verificación, etiquetas, inserción de imágenes, etc, un ambiente muy similar a Visual Basic, con lo que el desarrollador no tiene que preocuparse por programar el comportamiento de dichos objeto sino solo decirles que tienen que hacer con un mínimo de código.

Cuenta con asistentes y plantillas que permiten un mayor facilidad para la construcción de las formas, reportes o gráficas brindando un mínimo de información sobre como o que queremos ver en nuestra aplicación viendo los resultados de inmediato.

Aunque los asistentes pueden ahorrarnos mucho trabajo puede ser que no nos satisfaga del todo, por lo cual Developer permite cambiar todas las propiedades que queramos aun después de utilizar los asistentes, esto a través del navegador de objetos o definitivamente crear todo de manera manual.

Developer pone a nuestra disposición disparadores especializados asociados a los principales objetos (bloques de datos, cajas de texto, etc.) que con la programación adecuada, en PL/SQL, permite tener un mayor control de la mayoría de los eventos que puedan ocurrir cuando la aplicación este en ejecución, sin embargo también nos permite crear nuestros propios triggers. También permite una integración con la tecnología Java así como con componentes Active X.

2.5.2 Ventajas.

La interfaz visual con la que se trabaja en Developer ofrece diseñar las formas tal y como se verán al momento de que se ejecuten. Además Developer admite generar reportes y gráficas en formatos estandar como son el PDF, RTF y HTML, facilitando su distribución y por lo mismo, no se necesitará de Oracle Developer para visualizar dichos reportes o gráficas. Cabe señalar que con la ayuda de los asistentes, la construcción tanto de las formas, reportes o gráficas se vuelve un trabajo sumamente sencillo.

Developer brinda en sus componentes toda la flexibilidad y potencia de las bases de datos Oracle, conservando todas sus reglas de integridad y relaciones siguiendo el modelo de entidad-relación, permitiendo trabajar directamente con los objetos de las bases de datos, ya sean tablas, vista, secuencias, procedimientos, etc.

Sin duda, Developer explota al máximo todo el poder de las bases de datos Oracle, pero al mismo tiempo permite que nuestras aplicaciones trabajen con otros servidores de bases de datos haciendo uso del adaptador OCA (Open Client Adapter; Adaptador abierto para clientes).

OCA es una interfaz que permite hacer llamadas ODBC (Open Data Base Connectivity; Conectividad abierta de bases de datos) en lugar de llamadas Oracle en Oracle Developer, de esta manera Developer se puede comunicar con bases de datos que soporten el protocolo ODBC, como por ejemplo: Microsoft SQL Server y Access, Sybase System 10, Informix Ingres, IBM DB2/400, y Tandem NonStop SQL.

Developer permite una integración en entre sus componentes: Forms, Reports, y Graphics; con esto, podemos llamar un Informe o grafica a partir de una forma, o mostrar un informe desde una gráfica, en fin, una comunicación total entre estos productos Oracle.

Dicha integración no es solo con productos Oracle Developer, también puede integrarse con otros productos ajenos a Oracle de gran importancia como son los JavaBean (especificación de componentes que permiten usar sistemas de clases Java como componentes en los applets java, servlets o aplicaciones Java), y ActiveX que es una gama de tecnologías específicas de Microsoft que permiten usar interfaces estandarizadas para diversos aspectos del desarrollo de software.

Gracias a que Developer esta basado en objetos, podemos estandarizar los objetos con los que dispone modificando las clases principales, y por herencia, todos los objetos reciben las nuevas propiedades. Podemos crear objetos nuevos y guardarlos para poder usarlos en un futuro, obteniendo unos de los puntos mas deseados en el desarrollo de aplicaciones, la reutilización.

2.5.3 Desventajas

Una primer desventaja de Developer es el estar ligado completamente al DBMS Oracle; a pesar de que el adaptador OCA ayuda a mitigar esta dolencia de Developer, no lo convierte en un sistema completamente abierto, pues a pesar de que Developer se puede comunicar con otros servidores de bases de datos como SQL Server, no se pueden explotar todas sus características y potencialidad, sin embargo, si puede ser una buena opción si se requiere extraer información que se encuentre en otro servidor de bases de datos a parte del de Oracle.

Otra desventaja visible, aunque puede no ser tan grave, es cuando se quiere migrar la aplicación de un sistema operativo a otro, siendo concretos, de un ambiente de Microsoft Windows a uno basado en Un x. Para hacer esto se tiene que recompilar todos el c digo fuente que conforman la aplicaci n, aunque es necesario aclarar que esto solo suceder a cuando la aplicaci n se implement  en una arquitectura cliente/servidor.

2.5.4 Viabilidad de la Herramienta Oracle Developer

Para analizar la viabilidad de desarrollar una aplicaci n con Oracle Developer debemos de verlo desde dos puntos de vista, si contamos con Oracle como servidor de bases de datos o tenemos uno diferente. Primero veremos el primer punto.

Como hemos visto, Oracle Developer nos permite usar una arquitectura cliente/servidor o de tres capas; usando cliente/servidor ofrece crear una aplicaci n r pidamente y que funcione eficientemente gestionando autom ticamente la inserci n, actualizaci n y eliminaci n de los datos.

Si la arquitectura cliente/servidor ya no satisface las necesidades de la empresa puede llevarse la aplicaci n a la web usando las mismas formas, reportes y graficas que tenemos sin ninguna dificultad. Utilizar otras opciones para llevar una aplicaci n a la web (usando Java, php, asp, CGI, etc.) implica una codificaci n muy elevada para lograr las potentes capacidades de validaci n de datos con una interfaz de usuarios y el lenguaje de manipulaci n de datos automatizado (DML, que incluye inserci n, actualizaci n y eliminaci n) que incorpora Forms por default y el hecho de que Reports y Graphics pueden generar reportes y gr ficas en formatos html y pdf.

Cabe se alar que no se tiene que implementar forzosamente la aplicaci n en cliente/servidor para despu s migrarla a tres capas, se pueden crear los formularios, reportes y gr ficas para llevarlos inmediatamente a la web; y si se tiene ya la aplicaci n desarrollada en Oracle Developer con arquitectura en dos capas lo mas l gico y sensato es optar por el mismo Developer para llevar la aplicaci n a la web. Pasar de una arquitectura de cliente/servidor a tres capas es relativamente una tarea que se puede realizar f cilmente, sin embargo hay otras situaciones a tomar en cuenta, hablando particularmente de Forms, para decidir el usar Oracle Developer

CAPITULO II. HERRAMIENTAS CASE INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADORA

para llevar nuestra aplicación a la web. El explorador del cliente debe tener las características necesarias para mostrar las formas sin ninguna dificultad, se requiere como mínimo contar con Java 1.1 Virtual Machine o una versión posterior; con esto el cliente necesitará contar con IE 5 o posterior, Netscape 4.06 o posterior.

El ancho de banda también es un factor importante al momento de decidir el utilizar Oracle Developer para desarrollar aplicaciones para la web. Dependiendo del formulario, los archivos de formulario Java pueden llegar a ocupar hasta un mega en total y es un tamaño considerable para descargar con una conexión telefónica a 56Kb. Jinitiator ayuda que la aplicación se ejecute de un modo bastante mas rápido gracias a que guarda los archivos de clases en el cliente, pero implica que el cliente tiene que descargar jinitiator.

En hardware también hay unos inconvenientes, y esto es porque tanto en cliente/servidor como en tres capas, Oracle Forms mantiene la conexión activa con la base de datos durante la vida de la sesión lo cual requiere mas memoria en servidor, y el inconveniente se agudiza si queremos que nuestra aplicación este en la web, ya que por cada usuario concurrente se puede llegar a 1.5 MB de memoria por usuario.

Con estos datos podemos tomar una decisión partiendo de si la aplicación se requiere para una Intranet o internet. Si la aplicación se utilizará en cliente/servidor en una Intranet Developer es altamente eficiente, inclusive si se desea usar tres capas; pero si se requiere que la aplicación salga a la web puede llegar a ser mas factible buscar otras opciones; incluso Oracle tiene el paquete de PL/SQL para web que puede ser una de esas opciones. Si se tiene un servidor de bases de datos diferente a Oracle se debe optar definitivamente por otras opciones, aun y cuando la aplicación solo fuera para una Intranet y se utilizará el adaptador OCA para conectar Developer con ese otro servidor de bases de datos no se podría utilizar Developer a toda su potencia como ya se había mencionado anteriormente. Como conclusión, Oracle Developer puede ser una gran opción cuando se planea desarrollar una aplicación que se utilizará en una arquitectura cliente/servidor e incluso en tres capas para una intranet, y para una implementación en la web no brinda, al menos por ahora, la mismas ventajas y funcionalidad, hablando sobre todo de Oracle Forms ya que Reports y Graphics brindan un soporte mas amplio para ser usados en la web.

2.6 COMPONENTES DE ORACLE DEVELOPER

Oracle Developer esta compuesto por varios componentes, pero los mas importantes son Forms, Reports y Graphics. Project Builder, Procedure Builder, Schema builder y Query Builder son los otros componentes de Oracle Developer pero su objetivo solamente es apoyar a Forms, Reports y Graphics

2.6.1 Forms

Con el componente Forms se desarrollan las pantallas con las que interactuaran los usuarios para realizar todas las transacciones con la base de datos, aunque éste proceso es transparente para el usuario. Forms también proporciona el marco de trabajo de software para el desarrollo de módulos de bibliotecas cliente PL/SQL y menús.

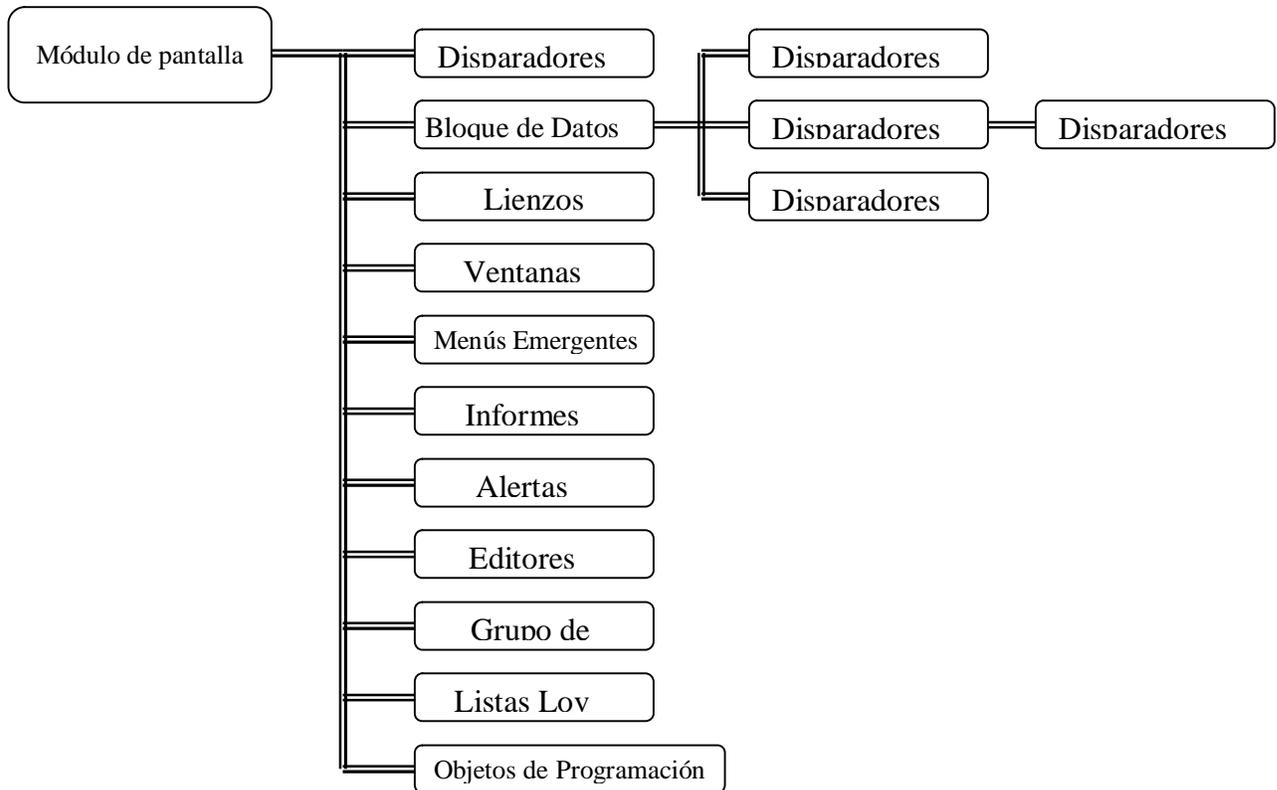


Fig. 2.7 Jerarquía de objetos en un módulo de pantalla.

2.6.1.1 Módulo de pantalla.

El módulo de pantalla es el componente principal de una aplicación de bases de datos. Este módulo esta compuesto por diversos objetos que nos ayudan a elaborar la aplicación que permita manejar los datos a través del lenguaje de manipulación de datos, estos objetos se presentan el la figura 2.7.

2.6.1.1.1 Disparadores

Los disparadores son bloques de código PL/SQL asociado a otro objeto, como puede ser una pantalla, bloque de datos o un elemento (caja de texto, casilla de verificación, etc.).

El Disparador se ejecuta cuando se producen determinados sucesos y en ese momento. Oracle Developer pone a disposición del desarrollador una amplia gama de disparadores predefinidos, se pueden crear disparadores personalizados pero no será muy frecuente la necesidad de crearlos.

2.6.1.1.2 Bloques de datos y Relaciones

Los bloques de datos es una colección de elementos como son las cajas de texto, etiquetas, casillas de verificación o de opción, etc., que nos permiten visualizar y navegar entre los registros de la base de datos, y realizar las acciones comunes en bases de datos: insertar, actualizar y eliminar. También nos permite realizar consultas a la base de datos.

Se pueden crear tantos bloques de datos necesitemos, solo dependerá de la memoria y de cómo los organicemos dentro del lienzo para una mejor visualización. Se pueden crear bloques de datos maestro-detalle, bloques de datos que tienen una relación padre-hijo. Oracle Developer nos permite crear bloques de datos manualmente o por medio de un asistente. La relación es un objeto que sirve para estructurar pantallas donde hacemos uso de bloques maestro-detalle, dichas relaciones pueden estar basadas en las relaciones de la base de datos.

2.6.1.1.3 Elementos.

Un elemento es la unidad primaria para la construcción de pantallas. Ya se han dado algunos ejemplos de elementos; tales como controles ActiveX, componentes Java Bean, elementos de lista, objetos OLE, botones, elementos de texto, etc. Cada objeto tiene una serie de propiedades que determinan como se comportarán durante la ejecución de la aplicación. Para hacer referencia es estos elementos dentro del código PL/SQL que se escriba en los disparadores es necesario utilizar la siguiente sintaxis: <bloque de datos>.<nombre de elemento>. Generalmente los elementos corresponderán a campos de una tabla de base de datos.

2.6.1.1.4 Lienzos y ventanas.

Los lienzos son los fondos donde se ubican los elementos. Los bloques de datos se pueden situar en diferentes lienzos y estos pueden ser de cinco tipos:

Pestaña: muestran pestañas independientes en las cuales se pueden colocar los bloques de datos.

Apilado: Se presentan sobre otros lienzos presentado contenidos condicionales o separables.

Barra de herramienta Horizontal y Vertical: Este tipo de lienzo presentará iconos y botones que ejecuten alguna acción.

Contenido: Es el lienzo principal donde se deben de situar todos los demás lienzos.

Los lienzos necesitan de otro objeto para poder ser visualizados, estos objetos son las ventanas. Las ventanas son objetos con las propiedades y características clásicas ya conocidas.

2.6.1.1.5 Menús contextuales, alertas.

Los menús contextuales también son objetos conocidos por sistemas basados en ventanas; estos permiten acceder a realizar acciones relevantes sobre los elementos sobre los que este el ratón. Las alertas permiten mostrar mensajes de advertencia cuando queremos prevenir a los usuarios sobre algún suceso de importancia e inclusive tomar una decisión de acuerdo a la respuesta del usuario y el botón que pulse.

2.6.1.1.6 Grupos de registros y listas LOV

Un grupo de registros es una estructura de datos que asemeja a una tabla con filas y columnas. Pueden ser estáticos o de consulta. Los estáticos son un especificación de columnas que se introducen a través de un cuadro de diálogo especial, los de consulta se definen mediante una instrucción SQL SELECT proporcionando su estructura de columnas.

Las listas Lov es un cuadro de diálogo especial que muestra un grupo de registros y permite elegir una fila del grupo devolviendo un único valor. Las listas lov se pueden asociar a un elemento y de esa manera ayudar al usuario a introducir un valor válido.

2.6.1.1.7 Objetos reutilizables.

Esta sección se pueden establecer los atributos de las clases sobre las cuales los objetos de Oracle Developer heredan sus atributos. También es posible empaquetar objetos en grupos reutilizables para un uso posterior, con solo copiar el grupo se copiarán todos los objetos pertenecientes a este. También es posible crear una biblioteca de objetos que sea accesible para cualquier proyecto.

Es también en este bloque donde se puede acceder a las unidades de programa, estas unidades de programa pueden ser procedimientos, paquetes y funciones PL/SQL definidos dentro del módulo aunque también es posible introducirlos a la biblioteca para que sean accesibles para otros módulos.

2.6.1.2 Forms. Módulo de menú.

Dentro de Oracle Forms encontramos el módulo de menú, el cual es mucho mas sencillo que el módulo de pantalla (Fig. 2.8).

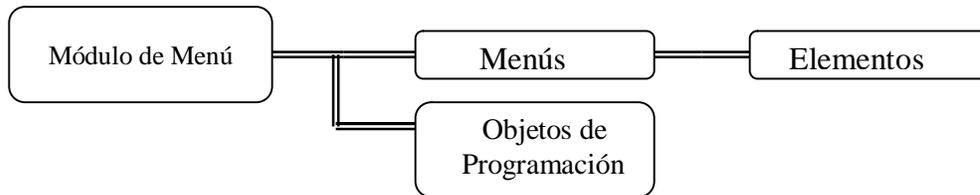


Fig. 2.8 Objetos del módulo de menú.

En este módulo es donde podemos crear menús, ya sean de pantalla o contextuales, tal y como se usan en Microsoft Windows. La programación sobre las acciones del menú se realiza con PL/SQL. El código se ejecuta al momento en que el usuario da clic en una acción del menú.

Tanto Forms, Reports y Graphics permiten acceder a los objetos de la base de datos como son las tablas, vistas y unidades de programas pero solo aquellas a las que el usuario tenga acceso.

2.6.2. Reports

El componente Reports de Developer es la parte del entorno en la que se diseñan los módulos de informes. Los componentes básicos de un informe son su modelo de datos, pantalla de parámetros, disparadores y diseño (Fig. 2.9).

El modelo de datos del informe es la estructura de los datos y sus diferentes representaciones en el informe; este modelo de datos se crea a través de un editor gráfico especial. Los parámetros del sistema son definidos por Reports y los definidos por el usuario los define el desarrollador. Se pueden establecer parámetros en tiempos de ejecución o crear un objeto de pantalla para que los parámetros sean introducidos por el usuario final del sistema.

Una consulta es una instrucción SQL que devuelve los valores de los datos que constituyen la base del informe. La consulta puede incrustarse en el informe o usar una consulta externa, esto es texto SQL almacenado en un archivo independiente que se puede utilizar en diferentes aplicaciones. Las consultas se pueden construir de tal manera que agrupen los datos de acuerdo a las necesidades de los clientes.

Las columnas de la base de datos son las columnas del Select de la consulta. Las columnas de fórmula son las columnas en las que se realizan cálculos usando bloques PL/SQL. Las columnas de resumen son para mostrar información de resumen sobre múltiples registros en el informe como subtotales y totales.

Los vínculos de datos se utilizan en informes maestro-detalle para vincular un grupo de filas a otro. El diseño del informe es un componente que nos permite crear la estructura gráfica de los informes, dicha estructura esta separada por la cabecera, el cuerpo de las páginas (donde se visualizarán los datos con un formato específico), y la página de terminación. Hay diferentes tipos de diseños predeterminados que presentan los informes en forma tabular, maestro-detalle, pantalla, carta formulario, etiqueta de correo o en forma matricial. Los disparadores son bloques de código PL/SQL que se pueden ejecutar antes o después de un informe, entre páginas, antes o después de la pantalla de parámetros.

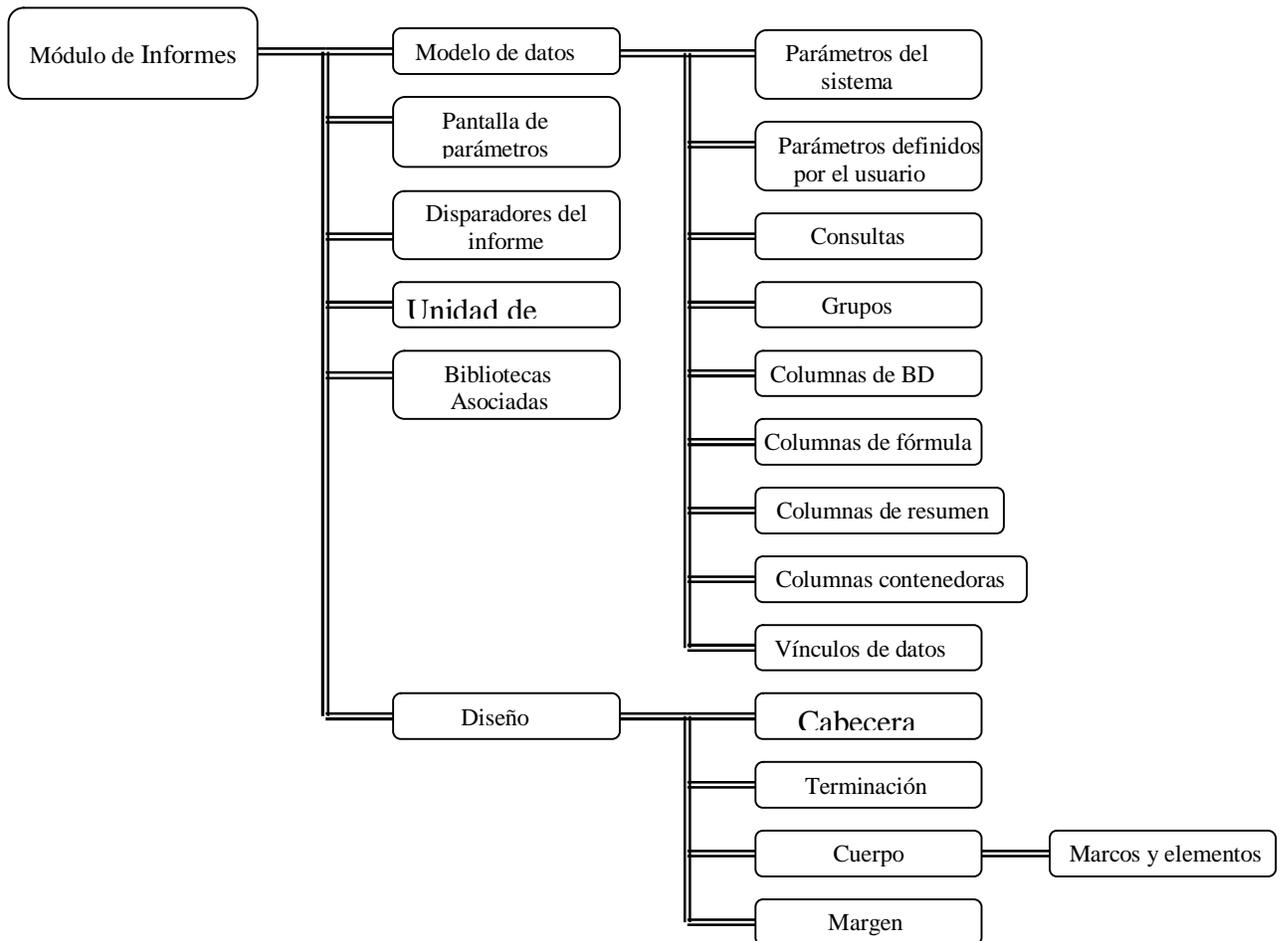


Fig. 2.8 Objetos del módulo de Reports

2.6.3. Graphics

El componente Graphics de Developer es la parte del entorno en la que se desarrollan módulos de presentación. Un modulo de presentación puede estar compuesto por uno o mas elementos gráficos con o sin referencia a la base de datos.

El diseño contiene los elementos gráficos de la presentación en una jerarquía que representa sus relaciones, y relaciona dichos elementos con las columnas de una consulta. Tenemos diferentes tipos de gráficos para representar nuestra información, son gráficos que generalmente estamos familiarizados sobre todo con los usados en excel de Microsoft Office.

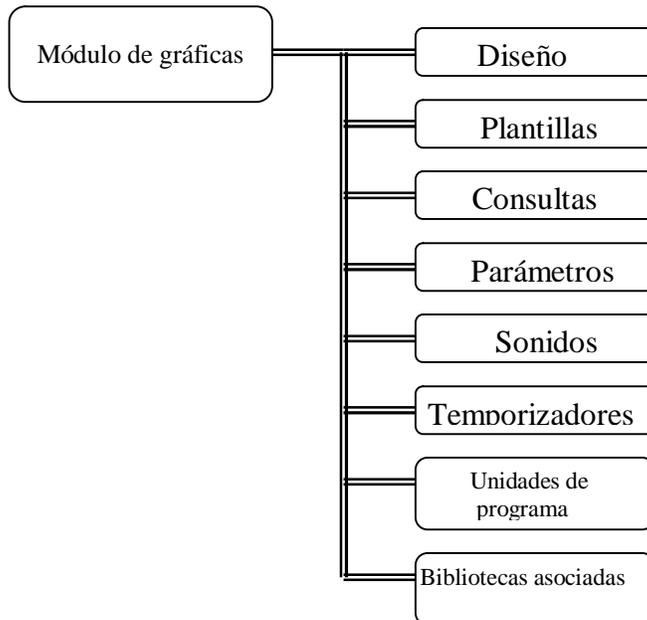


Fig. 2.9 Objetos del módulo de Graphics.

- Columna
- Barras
- Línea
- Mixto
- Doble "Y"
- Tarta
- Tabla
- Dispersión
- Alta-baja
- Gantt

Las plantillas son un conjunto de gráficos predefinidos que ayudan a construir gráficos con idéntico formato inclusive en la misma presentación. Las consultas son, al igual que en Reports, una sentencia SQL Select que definen los datos a representar gráficamente. Otro componente que es similar en Reports son los parámetros; son variables que se definen y usan en todo el módulo, dentro del código PL/SQL, estos se pueden establecer en la línea de ordenes o pasando una lista de parámetros desde otro programa como Forms o Reports. Graphics nos da la opción de agregar objetos sonidos a nuestros gráficos, creando presentaciones multimedia completas. El temporizador actúa en la presentación como una alarma de reloj, que al llegar al final que se ha determinado ejecuta un bloque PL/SQL asociado. En la figura 2.9 se puede ver los objetos del módulo de Graphics.

2.6.4 Introducción a PL/SQL

Es un sofisticado lenguaje de programación que se utiliza para acceder a base de datos Oracle desde distintos entornos. PL/SQL está integrado con el servidor de base de datos, de modo que el código PL/SQL puede ser procesado de forma rápida y eficiente.

PL/SQL (Procedural Language /SQL) lenguaje procedimental/SQL , como su propio nombre lo indica, PL amplía la funcionalidad de SQL añadiendo estructuras de las que pueden encontrarse en otros lenguajes procedimentales como:

- Variables y tipos (tanto predefinidos como definidos por el usuario).
- Estructuras de control, como bucles e instrucciones IF-THEN-ELSE.
- Procedimientos y funciones.
- Tipos de objetos y métodos.

2.6.4.1 PL/SQL y el tráfico de Red.

Muchas aplicaciones de bases de datos se construyen utilizando un modelo cliente-servidor o modelo de tres niveles. En el modelo cliente-servidor, el programa en sí reside en una máquina (cliente) y envía solicitudes a un servidor de base de datos para obtener información. Las solicitudes se llevan a cabo mediante SQL. Normalmente esto da lugar a múltiples comunicaciones por la red, una por cada instrucción SQL. En la fig.2.10 se muestra esta situación. Si se compara con la fig. 2.11 puede verse que se pueden empaquetar varias instrucciones SQL en un único bloque PL/SQL, que se envía al servidor como una sola unidad.

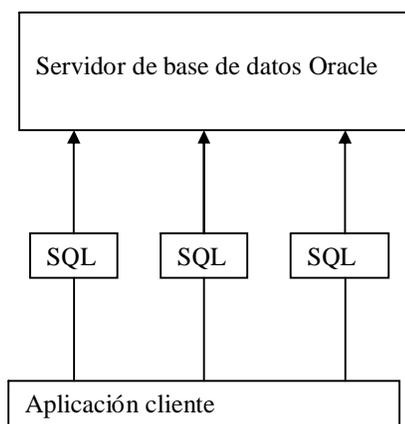


Fig 2.10 Utilizando SQL

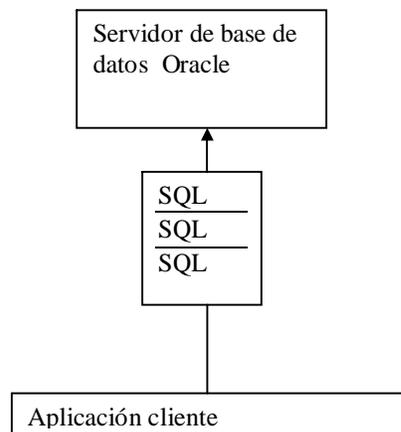


Fig. 2.11 Utilizando PL/SQL

El rendimiento crece aun cuando el cliente y el servidor estén siendo ejecutados sobre la misma máquina. En este caso no existiría red, pero el empaquetar las instrucciones SQL sigue produciendo un programa mas simple, que realiza menos llamadas a la base de datos. Las ventajas de empaquetar PL/SQL también se aplican al modelo de tres niveles. En este caso, el cliente (que con frecuencia se utiliza en un explorador HTML) se comunica con un servidor de aplicaciones, que a su vez se comunica con la base de datos.

2.6.4.2 Características de PL/SQL

PL/SQL, como cualquier otro lenguaje de programación sigue una estructura y una sintaxis que hay que seguir al momento de programar procedimientos o triggers de manera que el compilador no señale ningún error en tiempo de compilación.

2.6.4.2.1 Estructura de Bloques

La unidad básica de PL/SQL es el bloque, todos los programas PL/SQL están compuestos por bloques, que pueden ser anidados. Normalmente cada bloque define una unidad lógica de trabajo en el programa, separando así unas tareas de otras. Un bloque tiene la siguiente estructura:

```
DECLARE
/* SECCIÓN DECLARATIVA.- AQUÍ SE INCLUYEN LAS VARIABLES PL/SQL, TIPOS, CURSORES, ETC */
BEGIN
/*SECCIÓN EJECUTABLE.- AQUÍ SE INCLUYEN LAS INSTRUCCIONES SQL Y PROCEDIMENTALES. ESTA ES LA
SECCIÓN PRINCIPAL Y OBLIGATORIA DEL BLOQUE*/
EXCEPTION
/*SECCIÓN DE TRATAMIENTO DE EXCEPCIONES.- AQUÍ SE INCLUYEN LAS INSTRUCCIONES PARA EL
TRATAMIENTO DE LAS EXCEPCIONES*/
```

2.6.4.2.2 Tratamiento de errores

La sección de tratamiento de excepciones del bloque se utiliza para responder a los errores de ejecución con los que se encuentre el programa. Al separar el código de tratamiento de errores del cuerpo principal del programa, se consigue que la estructura de este sea clara.

2.6.4.2.3 Variables y tipos

La información se transmite entre PL/SQL y la base de datos mediante variables, una variable es una zona de almacenamiento que puede ser leída o asignada por el programa. Toda variable tiene un tipo específico asociado. Las variables PL/SQL pueden ser del mismo tipo que las columnas de una base de datos:

CAPITULO II. HERRAMIENTAS CASE INGENIERIA DE SOFTWARE ASISTIDA POR COMPUTADORA

```
V_STUDENTSNAME VARCHAR2(20);
```

O pueden ser de tipo adicionales:

```
V_LOOPCOUNTER BINARY_INTEGER;
```

También admite tipos definidos por el usuario, como por ejemplo, tablas y registros. Los tipos definidos por el usuario permiten personalizar la estructura de los datos que manipula un programa.

```
TYPE T_STUDENTRECORD IS RECORD (  
FIRSTNAME VARCHAR2(10);  
LASTNAME VARCHAR2(10) );
```

- Instrucciones condicionales.-Un programa PL/SQL puede ejecutar una porción u otra del código en función del resultado de una expresión condicional. La estructura mas significativa de este tipo de instrucciones se conoce como instrucción IF.
- Estructuras de bucle.-PL/SQL admite diferentes tipos de bucle. Un bucle permite ejecutar repetidamente una cierta secuencia de instrucciones
- Cursores.-Un cursor se emplea para procesar múltiples filas extraídas de la base de datos (con una instrucción select). Utilizando un cursor, el programa puede recorrer el conjunto de filas devuelto, una por una, y procesarlas.
- Procedimientos y funciones.-Los procedimientos y funciones (conocidos como subprogramas) son un tipo especial de bloque PL/SQL que se puede almacenar en la base de datos en formato compilado y que después son llamados desde un bloque.
- Paquetes.-Los subprogramas, junto con las variables y tipos pueden agrupar en un paquete. Un paquete consta de dos partes, la especificación y el cuerpo, y permite que objetos relacionados se almacenen juntos en la base de datos.
- Colecciones.-Las colecciones PL/SQL son similares a las matrices de los lenguajes de 3GL. PL/SQL proporciona tres tipos diferentes de colecciones, tablas organizadas por índice , tablas anidadas y varrays o matrices variables .

2.6.4.3 Entornos de desarrollo y ejecución

Una aplicación de base de datos puede descomponerse en tres partes:

La interfaz de usuario.- determina el aspecto y estilo de la aplicación de cara al usuario. Esta capa trata la entrada de datos por parte del usuario y la presentación de los datos a la salida de la aplicación.

La lógica de la aplicación.- controla las tareas llevadas a cabo por la aplicación

La base de datos.- almacena los datos de la aplicación de forma persistente y fiable.

El modelo de dos niveles o cliente-servidor es un modelo convencional para el diseño de aplicaciones. La aplicación se divide en dos partes: el cliente y el servidor. El cliente gestiona la interfaz de usuario, mientras que el servidor contiene la base de datos

2.6.4.4 PL/SQL en el servidor

PL/SQL se encuentra disponible en el servidor de base de datos, siendo esta la ubicación original del motor PL/SQL. Como el servidor de base de datos también procesa instrucciones SQL, tanto las instrucciones SQL como los bloques PL/SQL pueden enviarse a la base de datos para su procesamiento. SQL *PLUS es un ejemplo de una aplicación cliente de este tipo, en la que las instrucciones SQL y bloques PL/SQL, introducidos de forma interactiva mediante la línea de comandos SQL, se envían al servidor para su ejecución.

2.6.4.5 PL/SQL en el cliente

Además del motor de PL/SQL en el servidor, dos de las herramientas de desarrollo de Oracle (forms y reports) contienen un motor de PL/SQL. La propia herramienta de desarrollo se ejecuta en el cliente y no en el servidor. Con PL/SQL en el lado del cliente las instrucciones procedimentales dentro de los bloques se ejecutan en el cliente y no se envían al servidor.

A lo largo de éste capítulo se dio una vista general de lo que es la ingeniería de software con sus diferentes paradigmas, y la existencia de herramientas CASE basadas en alguna fase de los diferentes paradigmas que pueden ayudar a los desarrolladores de software a crear aplicaciones en un menor tiempo y con una mayor eficiencia. También se mostró de manera específica la herramienta CASE de cuarta generación que se utilizó para construir el sistema para el área de Posgrado dentro del Instituto Mexicano del Petróleo; ahora sigue el turno de la Base de Datos, cuyo ciclo de vida de analiza en el siguiente capítulo.

Capitulo III.
CICLO DE VIDA DE LAS BASES DE DATOS.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

La creación de una Base de Datos generalmente es una operación difícil, larga, costosa y que no puede improvisarse; es por esta situación que también está sometida a un ciclo de vida al igual que un sistema de información. El diseño de una base de datos suele ser una tarea técnica, aunque también está enfocada a las personas, ya que la responsabilidad de las decisiones relativas a todo el proceso de creación de una base de datos no corresponde únicamente a los informáticos, sino que, en ciertas fases, son los directivos y los usuarios quienes tienen un papel trascendental en la creación de la base de datos. A continuación se observaran las diferentes fases del ciclo de vida de las bases de datos (Fig.3.1) y posteriormente se procederá a examinar cada una esas etapas.

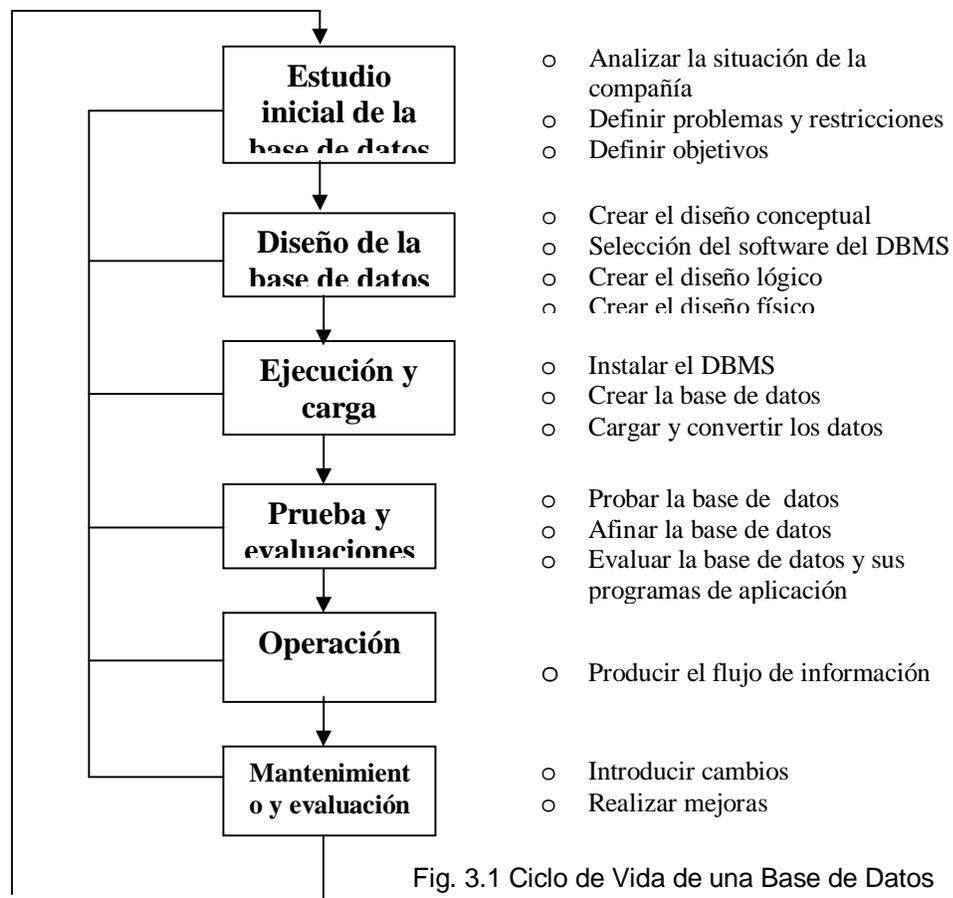


Fig. 3.1 Ciclo de Vida de una Base de Datos

3.1 ESTUDIO INICIAL DE LA BASE DE DATOS.

Al igual que el ciclo de vida del desarrollo de software, el desarrollo de una base de datos debe iniciar con el análisis de requerimiento y el conocer el funcionamiento interno de la empresa o institución para poder realizar un buen diseño.

3.1.1 Analizar la situación de la compañía

Esta fase denominada también como de análisis previo, estudio de oportunidad o viabilidad, debe preceder a cualquier operación de concepción o diseño de una base de datos. La palabra análisis literalmente significa descomponer un todo en sus partes para investigar su naturaleza, función etc. En ella se ha de concretar la voluntad de los directivos para abordar el proyecto; por otra parte, la situación de la empresa describe las condiciones en las que opera una compañía, su estructura organizacional y su misión, definiendo objetivos claros y concretos que sirvan de pauta en todo el desarrollo.

Se debe estar al tanto de cuál es el ambiente operativo de la compañía y cuál es su misión en ese ambiente para poder crear un diseño que concuerde con dicho ambiente y misión.

La recolección de la información necesaria para el análisis, puede realizarse de diversas maneras tales como:

- Entrevistando al personal de la empresa, concretamente, a aquellos que son considerados expertos en las áreas de interés.
- Observando el funcionamiento de la empresa.
- Examinando documentos, sobre todo aquellos que se utilizan para recoger o visualizar información.
- Utilizando cuestionarios para recoger información de grandes grupos de usuarios.
- Utilizando la experiencia adquirida en el diseño de sistemas similares.

Es necesario tener en cuenta la estructura organizacional de la compañía para definir los flujos de información requeridos, consultas y reportes específicos de acuerdo al nivel que ocupa un usuario en dicha estructura, ya que un ejecutivo y un contador no van a requerir el mismo tipo de reportes o consultas.

El conocer la estructura orgánica es importante no sólo por lo citado anteriormente sino para determinar la unidad que tendrá la responsabilidad de la gestión y control de la base de datos donde dicha unidad es imprescindible para que el entorno de bases de datos funcione debidamente.

Se pueden distinguir dos objetivos referentes a esta unidad: el primero es el administrador de datos, el cual tendrá responsabilidad sobre la planificación, organización y control de los datos de la empresa; el segundo es el administrador de la base de datos, quien lleva la responsabilidad de la administración del entorno de base de datos. A continuación se presentan algunas de las funciones y responsabilidades tanto del administrador de datos como del administrador de la base de datos.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

Administrador de datos.

- Realizar el diseño conceptual y lógico de la base.
- Apoyar al personal de sistemas durante el desarrollo de aplicaciones.
- Formar al personal.
- Establecer estándares de diseño de BD, desarrollo y contenido del diccionario de datos.
- Diseñar la documentación incluida en el diccionario.
- Desarrollar políticas de gestión de datos.
- Desarrollar planes estratégicos y tácticos para la manipulación de los datos.
- Desarrollar los requisitos de los elementos del diccionario de datos.
- Desarrollar normas para denominación.
- Controlar la integridad y seguridad de los datos.
- Planificar la evolución de la BD de la empresa.
- Identificar oportunidades de compartición de datos.
- Trabajar con los auditores de la base.
- Proporcionar controles de seguridad.

Administrador de bases de datos.

- Realizar el diseño físico de la base de datos.
- Asesorar la adquisición de hardware y software.
- Soportar el DBMS.
- Resolver problemas del DBMS y del software asociado.
- Monitorizar el rendimiento del DBMS.
- Ayudar en el desarrollo de planes que aseguren la capacidad hardware.
- Asegurar la integridad de los datos, comprobando que se implantan los controles adecuados.
- Seguridad y confidencialidad.
- Proporcionar facilidades de prueba.
- Integrar paquetes, procedimientos, utilidades, etc., de soporte al DBMS.
- Desarrollar estándares, procedimientos y documentos.

Al detallar las responsabilidades de estas funciones se debe tener en cuenta la separación de funciones; dicha separación debe efectuarse entre:

- El personal de desarrollo de sistemas y el de explotación.
- Explotación y control de datos.
- Administrador de bases de datos y desarrollo.
- Administrador de bases de datos y el administrador de la seguridad.

El hecho de que se plantee realizar esta separación de funciones no implica que éstas sean desempeñadas por personas distintas ya que tal situación no es posible en el caso de pequeñas y medianas empresas, pero es un aspecto importante de control a tomarse en cuenta.

3.1.2 Definición de Problemas y restricciones.

El proceso de definición del problema inicialmente podría parecer no estar estructurado. Los usuarios finales de la empresa a menudo no son capaces de describir con precisión el alcance de las operaciones de la empresa o identificar los problemas reales que se presentan durante las operaciones de la misma, es natural que los problemas de la empresa sean vistos desde una perspectiva diferente para un gerente que para una persona que hace el trabajo rutinario dentro de la empresa.

Es por este motivo que la información recogida debe incluir las principales áreas de aplicación y los diferentes grupos de usuarios, la documentación utilizada o generada por estas áreas de aplicación o grupos de usuarios, las transacciones requeridas por cada área de aplicación o grupo de usuarios y una lista priorizada de los requerimientos de cada área de aplicación o grupo de usuarios.

Aunque se puede llegar a una definición del problema muy completa y que satisfaga las necesidades de la compañía de una manera precisa, existen ciertos factores que actúan como restricciones para llegar a la definición final del problema y que pueden hacer que esa definición tenga que modificarse, tales como el tener un cierto presupuesto, el personal a disposición para trabajar con la base de datos, o bien que se tenga que trabajar con cierto hardware o software que la compañía ya posee.

3.1.3 Definición de los objetivos.

El diseñador debe asegurarse de que los objetivos de la bases de datos, tal y como él los ve, correspondan a los visualizados por el usuario para resolver la lista de problemas que surgieron en la etapa anterior; en definitiva, se trata de determinar cuál es el objetivo inicial del sistema propuesto, si el sistema interactuará con otros sistemas ya existentes o futuros o si el sistema compartirá datos con otros sistemas o usuarios.

3.1.4 Determinación del alcance y límites de la base de datos.

El diseñador debe reconocer la existencia de dos conjuntos de limitaciones: alcance y límite. El alcance del sistema define a la extensión del diseño, de acuerdo con los requerimientos operativos. El conocimiento del alcance del diseño de una base de datos ayuda a definir las estructuras de datos requeridas, el tipo y número de entidades, el tamaño físico de la base datos, etc. El sistema propuesto también esta sujeto a límites, los cuáles son externos al sistema, los límites también son impuestos por el hardware y el software existentes. Idealmente el desarrollador puede elegir el hardware y software que mejor logren los objetivos del sistema, la selección del software es un aspecto importante del ciclo de vida del desarrollo de sistemas.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

En el mundo real un sistema debe diseñarse entorno al hardware existente. El alcance y los límites llegan a ser factores que obligan al diseño a adoptar un molde específico y la tarea del diseñador es el diseñar el mejor sistema dentro de esas limitaciones.

El diseñador completa el diseño de los procesos del sistema, esto incluye todas las especificaciones técnicas para las pantallas, reportes y otros dispositivos que pudieran ser utilizados para hacer del sistema un generador de información eficiente. La información recogida se debe estructurar utilizando *técnicas de especificación de requisitos*, como por ejemplo técnicas de análisis, diseño estructurado y diagramas de flujo.

3.2 DISEÑO DE LA BASE DE DATOS

La segunda fase se enfoca en el diseño del modelo de base de datos que soporta las operaciones y objetivos de la compañía. En el proceso de diseño de una base de datos hay que concentrarse en las características de los datos requeridos para construir el modelo de dichas bases.

Se tienen dos visualizaciones de los datos dentro del sistema: la visualización de los datos dentro de la empresa como fuente de información y la del diseñar la estructura de los datos, su acceso y las actividades requeridas para transformarlos en información.

Cuando se comienzan a examinar los procedimientos requeridos para completar el diseño de la base de datos, se debe recordar que:

- El proceso del diseño de una base de datos guarda cierta relación con el análisis y diseño de un sistema más grande. El componente de datos es sólo un elemento de un sistema de información más grande.
- Los analistas o programadores de sistema están a cargo de diseñar los demás componentes del sistema. Sus actividades crean los procedimientos que ayudarán a transformar los datos dentro de la base de datos en información útil.
- El diseño de base de datos no constituye un proceso secuencial, es un proceso iterativo que proporciona una retroalimentación continua para rastrear los pasos previos.

3.2.1 Modelo Conceptual

El modelado conceptual constituye la primera fase de desarrollo de bases de datos y pretende representar el problema del mundo real tal y como se percibe, sin tener en cuenta cómo este problema puede ser representado para que sea entendido por los programas de computador. Se puede dividir en dos etapas:

A) Análisis de requisitos.

Es la etapa de percepción, identificación y descripción de los fenómenos del mundo real a analizar, esta parte se obtiene del estudio inicial del punto 3.1. En esta etapa el objetivo es saber ¿Qué se va a representar? Para poder responder esta pregunta y poder elaborar un esquema descriptivo de la realidad es necesario estudiar las reglas de la empresa, éstas proveen el marco para el análisis del sistema; y realizar entrevistas a los usuarios de los diferentes niveles de la organización, éstos nos proveen de los detalles sobre los datos; y analizar la documentación existente.

Este esquema descriptivo se expresará en lenguaje natural, esto permite que el usuario pueda establecer en sus propios términos cuál es el problema a resolver. La comunicación entre el usuario y el analista es de suma importancia para el diseño de la base de datos, si se presentan problemas en el entendimiento entre estas dos partes, repercutirán en el diseño de la base de datos. A continuación se muestran algunas actitudes más usuales en cada grupo respecto al otro. Dichas actitudes deben tratar de sanearse lo más que se pueda para poder lograr un mejor diseño conceptual.

Cómo ven los usuarios a los analistas.

- No entienden el negocio, es decir, la actividad de la empresa.
- Intentan decirnos cómo realizar nuestro trabajo.
- No consiguen instrumentar de manera aceptable las especificaciones del sistema.
- Dicen que no a todas nuestras sugerencias.
- Ponen mucho énfasis en aspectos técnicos.
- Siempre piden más presupuestos.
- Siempre se retrasan.
- Nos piden tiempo y esfuerzo en detrimento de nuestro trabajo.
- No pueden responder de forma rápida y satisfactoria a los cambios necesarios en el sistema.

Cómo ven los analistas a los usuarios.

- No saben lo que quieren.
- Tienen muchas necesidades políticas.
- Quieren todo ya.
- No son capaces de establecer prioridades entre las necesidades.
- Quieren poner sus necesidades específicas por delante de las de la compañía u organismo.
- Rehúsan responsabilidades sobre el sistema.
- No son capaces de dar una definición clara del sistema para que funcione.
- Son incapaces de respetar la planificación.
- No dicen todo lo que saben sobre el sistema.

B) Etapa de conceptualización.

En esta etapa se transforma el esquema descriptivo al esquema conceptual apoyándose en un modelo de datos que en este caso es el modelo entidad-interrelación. En esta etapa se define ¿Cómo representar el esquema descriptivo que se obtiene?. Para lograr esta transformación se deben interpretar las frases del lenguaje natural obtenidas en la primera etapa, convirtiéndolas en elementos del modelo E/R. No existen reglas determinadas para lograr dicha transformación, pero existen ciertos principios generales que pueden ayudar a generar el esquema conceptual aunado al criterio del diseñador.

A continuación se mencionan dos enfoques que se pueden emplear para obtener el esquema conceptual con base a los análisis del esquema descriptivo, el enfoque lingüístico propuesto por Peter Chen, pionero en el modelado conceptual y el enfoque de categorización de objetos de Carswell y Navathe.

El enfoque lingüístico de Chen pretende el que se dependa menos de la intuición de los diseñadores y más de métodos estructurados. Estas recomendaciones son las siguientes:

- Un sustantivo (nombre común) que actúa como sujeto o complemento directo en una frase es, en general, un tipo de entidad, aunque podría ser un atributo. Un alumno toma una o más actividades académicas.
- Los nombres propios suelen indicar ejemplares de un tipo de entidad. “Joel Domínguez Rangel” puede ser un ejemplar de alumno.
- Un verbo transitivo o una frase verbal es un tipo de interrelación. Un alumno toma una o más actividades académicas.
- Una preposición o frase preposicional entre dos nombres suele ser un tipo de interrelación, o puede establecer la asociación entre una entidad y sus atributos. “El área del departamento” puede indicar una interrelación entre dos entidades, área y departamento, o se puede estar asociando “área” como atributo de la entidad departamento.

El enfoque de la categorización de los objetos se basa en el papel o rol que un determinado objeto desempeña en el proceso de la información, y señala lo siguiente:

- Una entidad es un objeto de datos que tiene más propiedades que su nombre o se utiliza como operando en una sentencia de selección, borrado o inserción. Un profesor posee una serie de propiedades como nombre, apellidos, dirección, etc. Y cuando un profesor deje la escuela se tiene que dar de baja de la base de datos.
- Un atributo es un objeto de datos al que se le asigna un valor o se utiliza como operando en una operación aritmética, booleana o una cadena de caracteres. Asignar una matrícula a un alumno.
- Una interrelación es un objeto de datos que hace posible la selección de una entidad por medio de una referencia a un atributo de otra entidad. Una

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

actividad académica (materia) es cursada por uno o más alumnos; por lo que podemos saber qué alumnos están inscritos en una determinada actividad académica.

A partir de estos enfoques, los cuales es posible fusionar, se puede obtener un modelo conceptual.

3.2.2 Selección del software DBMS

La selección del DBMS es crítico para la operación fluida del sistema de operación. Si bien, los factores que afectan la decisión de compra varían de acuerdo a las necesidades y presupuesto de la empresa, podemos mencionar algunos puntos importantes a tener en cuenta al momento de evaluar un DBMS.

- Modelo subyacente. Jerárquico, red, relacional, orientado a objetos.
- Portabilidad. A través de plataformas, sistemas y lenguajes.
- El diccionario de datos.
- La seguridad e integridad de los datos.
- Las capacidades de consulta, de manipulación y de informes.
- El soporte a los requisitos de programación especializada.
- Requerimientos para el hardware de DBMS: procesador(es), RAM, espacio en disco, etc.
- Costos de compra, mantenimiento, operación, licencia, instalación, entrenamiento y conversión.

El Sistema Manejador de Bases de Datos debe soportar un modelo existente de bases de datos, tal como el jerárquico, de red relacional u orientado a objetos. La facilidad con la que un producto de programación pueda ser transferido de un sistema informático a otro, o de un entorno a otro es también un elemento importante al momento de elegir el DBMS que se implementará en la organización; es decir, debe brindar portabilidad.

Un sistema de bases de datos efectivo debe admitir el crecimiento de la base sin pérdida de eficiencia y la modificación de su estructura no tiene que comprometer la integridad de los datos. El diccionario de datos ayuda a cumplir dichos objetivos permitiendo que las definiciones de los datos se mantengan separadas de éstos. Con esto se logrará hacer cambios en las definiciones de datos sin que tenga ningún efecto en los datos que se encuentran almacenados.

La seguridad e integridad de los datos son esenciales para lograr operaciones efectivas sobre la base de datos que también son importantes para seleccionar un DBMS, de manera específica hablamos de:

- Controles de acceso
- Controles de concurrencia
- Facilidades de cifrado
- Controles de seguridad y recuperación.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

Se deben establecer límites de acceso a los datos de acuerdo a los privilegios que se asignen a los usuarios que vayan a interactuar con la Base de Datos; dichos controles deberá proteger los datos frente a cambios no autorizados, o de una corrupción que se pueda dar de manera accidental por parte del usuario. El control puede ir desde sólo ejecutar consultas hasta tener los privilegios para añadir, corregir, copiar o borrar datos, o la manipulación de los objetos de la base de datos como son las tablas, vistas, procedimientos, etc.

Una base de datos será utilizada generalmente en sistemas multiusuarios, por lo que es necesario un mecanismo para controlar la concurrencia. Los DBMS deben proveer esta característica importante para evitar inconsistencias en la información. Deben contar con técnicas de concurrencias; como llevar a cabo bloqueos durante las transacciones impidiendo a otros usuarios la recuperación y actualización de los elementos bloqueados, dichos bloqueos pueden ser por el campo de un registro, por un registro, tabla o toda la base de datos.

El cifrado puede proveer protección adicional a secciones confidenciales de una base de datos. Los datos se protegen mediante algún algoritmo de codificación, de esta manera sólo un usuario autorizado contará con el algoritmo para descifrarlos. Otra técnica de protección de la confidencialidad (y también de la integridad) que puede tener un DBMS es la criptografía, que permite transformar el contenido de la base, haciéndolo incomprensible a cualquier usuario que acceda a dichos datos sin la correspondiente clave de descifrado.

Los DBMS deben proporcionar mecanismos que permitan recuperar la base de datos contra fallos lógicos o físicos que destruyan los datos contenidos. Al ocurrir un fallo que dé lugar a pérdida de memoria volátil, es preciso realizar la operación que se suele denominar *recuperación en caliente*, en la que el sistema consulta el fichero diario para determinar las transacciones que hay que deshacer porque no han sido completadas y las que hay que rehacer porque, si bien se han completado, no habían sido grabadas en la base de datos cuando se produjo el fallo.

Recuperación en frío. En caso de un fallo de memoria secundaria que afecte a la base de datos, se lleva a cabo una *recuperación en frío*, que consiste en utilizar una copia de seguridad de la BD, también llamada de *respaldo (backup)* La copia de seguridad permitirá, junto con los ficheros diarios que se han ido produciendo desde que se realizó la copia de seguridad, reconstruir la BD llevándola de forma consistente a la situación anterior a que se produjera el fallo.

Un DBMS debe proporcionar la capacidad de generar informes estructurados en una variedad de formatos. Además debe brindar un lenguaje que sea poderoso, fácil de aprender y usar, que permita la manipulación de los datos y la realización de consultas de una manera eficiente. Soporte a los requisitos de programación especializada. Desarrollar programas especializados para interactuar con los DBMS requiere de facilidades para satisfacer el desarrollo de aplicaciones y sus

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

pruebas; así como un lenguaje para expresar las estructuras estándares de programas procedimentales o brindar una interfaz con uno o más lenguajes existentes.

Se han mencionado los puntos importantes que se deben tener en cuenta al elegir un DBMS, dichos puntos se enfocan al funcionamiento del Sistema Manejador de Bases de Datos y como tratan la información almacenada. Puede ser que la empresa cuente ya con un hardware específico y se tenga que adquirir un DBMS que se adapte a dicho hardware o que la empresa tenga la posibilidad de comprar un equipo de acuerdo al DBMS que se elija.

Por lo tanto, el procesador o procesadores que pueda tener el servidor, la memoria Ram, o el espacio en disco, así como los costos que implica la adquisición del hardware y desde luego el DBMS que se decida comprar son puntos importantes al momento de implementar una base de datos, y en ocasiones, puede ser uno de los primeros puntos, sino es que el primero, al momento de elegir el DBMS a implementar, ya que depende directamente de los recursos de la empresa.

3.2.3 Diseño lógico

En este nivel se transforma el esquema conceptual, adaptándolo al modelo de datos en el que se apoya el DBMS que se va a utilizar, en este caso se trata de un modelos de datos relacional; se debe comparar las características y restricciones del modelo conceptual con las del modelo de ejecución del DBMS seleccionado. A continuación se presentan unas reglas para obtener el esquema lógico a partir del esquema conceptual.

1. Entidades: Todos los tipos de entidad presentes en el esquema conceptual se convierten en tablas manteniendo el número y tipo de atributo. Los atributos identificadores pasan a ser llave primaria, los atributos identificadores alternativos tendrán valores únicos y los otros atributos pasan a ser columnas que pueden o no tomar valores nulos.
2. Interrelaciones N:M. Este tipo de relación se convierte en una tabla que tendrá como llave primaria la concatenación de los identificadores principales de las entidades asociadas.
3. Interrelaciones 1:N.
 - a. Se puede propagar los atributos identificadores principales del tipo de entidad que tiene la cardinalidad máxima 1 a la que tiene N (en el sentido de la flecha) desapareciendo el nombre de la interrelación (solución habitual).
 - b. También es posible convertir la interrelación en una tabla como si se tratase de una interrelación N:M, donde la llave primaria de dicha tabla creada será la llave primaria de la tabla a la que le corresponde la cardinalidad N.

Se hará uso de la opción “b” solo en tres casos:

1. El número de ejemplares interrelacionados de la entidad que propaga su clave es muy pequeño, por lo que existirían muchos valores nulos en la clave propagada.
 2. Cuando se prevé que dicha interrelación en un futuro se convertirá en un tipo N:M.
 3. Cuando la interrelación tiene atributos propios y no se desea propagarlos.
4. Interrelaciones 1:1 En este caso de relación se procede a la transformación de acuerdo a la cardinalidad señalada en el modelo conceptual.
- a. Si en ambas entidades se presenta una cardinalidad (0,1), será conveniente transformar la interrelación en una tabla donde la llave primaria de dicha tabla será la concatenación de las llaves primarias de las tablas involucradas.
 - b. Si la cardinalidad de una de las entidades es (0,1) y la otra es (1,1), es conveniente propagar la clave de la entidad con cardinalidad (1,1) en la tabla resultante de la entidad de cardinalidad (0,1).
 - c. En el caso en que las cardinalidades de las entidades involucradas sea (1,1), la propagación de la clave puede hacerse en cualquiera de las tablas resultantes, teniendo como punto de decisión, la tabla que puede tener accesos más frecuentes a los datos de las tablas.
5. Atributos de interrelaciones: Si la interrelación se transforma en una tabla, todos sus atributos pasan a ser columnas de dicha tabla; en caso de que se propague la llave, los atributos migran junto con la llave a la tabla que corresponda, aunque es más factible convertir la interrelación que contiene atributos en una tabla.

3.2.4 Reglas de Normalización

Las reglas de normalización

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. El aplicar las reglas de normalización ayuda a prevenir errores lógicos en la manipulación de datos, las estructuras de datos son mas simples y estables, por lo que su mantenimiento es mas sencillo permitiendo agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Otra ventaja de la normalización de base de datos es el consumo de espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

Existen varios niveles de normalización y cada nivel está basado en la que le antecede: Primera Forma Normal, Segunda Forma Normal, Tercera Forma Normal, Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal. Conforme se va aplicando cada nivel de normalización nos acercamos a una Base de Datos verdaderamente relacional.

Primera Forma Normal

“Una relación R satisface la primera forma normal (FN1) si, y solo si, todos los dominios subyacentes de la relación R contienen valores atómicos”.

Esta primera regla formal consiste en descomponer las tablas en las que existan columnas repetidas en tantas tablas como valores estén presentes. El objetivo es no tener datos repetidos en las tablas, consiguiendo una mayor eficiencia en el manejo de los de datos ya que la base de datos estará organizada en partes mas pequeñas y por consiguiente será mas fácil de mantener y escalar.

Segunda Forma Normal

“Una relación R satisface la segunda forma normal si, y solo si, R esta en primera forma normal y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esta relación”.

Todos los atributos de las tablas que no son clave (llave primaria) deben depender completamente de esta. No debe existir una dependencia entre atributos no primos, cada uno de los atributos tienen que depender de la llave primariao de lo contrario se tienen que descomponer en las relaciones que sean necesarias hasta que se cumpla esta regla.

Tercera Forma Normal

“Una relación R satisface la tercera forma normal si, y solo si, satisface la segunda forma normal y cada atributo no primo de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación”.

Las dependencias transitivas se dan en el momento en que existe mas de una forma de llegar a hacer referencias a un atributo de una relación o que un atributo que no es llave primaria depende de otro atributo que tampoco es llave primaria. Si suponemos que un atributo R.a es la llave primaria de la relación R, y tiene dos atributos, que no son llave primaria, que depende entre si R.b → R.c y ambas dependen de la llave primaria R.a; se dice que hay una dependencia funcional transitiva, para lo que se tiene que realizar otra descomposición para eliminar dicho problema.

Forma Normal Boyce-Codd

“Una relación R satisface la forma normal Boyce-Codd si, y solo si, se encuentra en FN1, y cada determinante funcional es una clave candidata de la relación R.”

3.2.5 Diseño físico

Teniendo presentes los requisitos de los procesos, características de los DBMS, del SO y del hardware, en el diseño físico se pretende:

- Disminuir los tiempos de respuesta.
- Minimizar el espacio de almacenamiento.
- Evitar las reorganizaciones.
- Proporcionar la máxima seguridad.
- Optimizar el consumo de recursos.

El diseño físico se enfoca tanto en la ubicación de los datos en los dispositivos de almacenamiento como en el funcionamiento del sistema. Los actuales DBMS relacionales brindan una mayor flexibilidad al proporcionarle al administrador la capacidad de definir la estructura interna de la base de datos a partir de algunos parámetros, los cuales es posible modificarlos a fin de realizar ajustes (tuning) y optimizar la base de datos.

Gracias a esa flexibilidad, la base de datos puede empezar a funcionar de inmediato, la eficiencia va aumentando al irse efectuando sucesivos ajustes y se mantiene la independencia físico/lógica, por lo que el administrador ha de elegir las técnicas de estructuración física que permitan un acceso más eficiente a los datos.

En el caso del diseño físico no se cuenta con un modelo formal como sucede con el modelo conceptual o lógico, por lo que se hace dependiente del manejador de bases de datos que se haya seleccionado. Sin embargo se pueden seguir unas consideraciones de carácter general respecto a la organización de archivos.

Los datos de la base, en su nivel intermedio se organizan en archivos (colecciones de registro) que se almacenan en discos magnéticos, siendo el registro físico (página o bloque) la unidad básica. Dicho bloque puede contener varios registros lógicos (filas), su tamaño depende del producto que utilicemos así como del sistema operativo.

El sistema gestor de bases de datos accede a los datos haciendo uso de los mecanismos de gestión de archivos del sistema operativo o hace uso de métodos de acceso propio. Para acelerar la recuperación de datos de la base, se emplea la memoria intermedia llamada buffer.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

En algunos DBMS, el administrador puede especificar las características de los bloques, de esta manera los bloques se pueden agrupar de forma contigua en unidades mayores. Algunos de los parámetros que se pueden especificar son:

- El porcentaje de espacio libre de cada bloque que se reserva para futuras modificaciones de las filas de la tabla. Este espacio libre evita reorganizaciones en la base de datos, por ejemplo, al actualizar datos, puede ser que no quepan donde se alojaban los antiguos.
- Porcentaje de utilización de cada bloque. Un valor bajo aumenta el espacio no utilizado de la base de datos, pero reduce el procesamiento para el borrado de datos, mientras aumenta el de inserción.
- Número de bloques que se asignan a las extensiones de una tabla, de forma inicial o para sucesivas ampliaciones, así como el crecimiento de las extensiones y el número máximo de éstas.

3.2.5.1 Organización de Archivos

La organización de los archivos que constituyen la base de datos y los métodos de acceso repercuten en los tiempos de respuesta de los sistemas de gestión de bases de datos. Los métodos de acceso básicos son el secuencial (se localiza el registro deseado pasando por los registros anteriores), y el directo (se localiza el registro por su dirección). Para cada tabla se deberá elegir el tipo de organización basándose en tres aspectos:

Tipo de Proceso: Tipo de acceso requerido, secuencial o directo. También si la operación sobre el archivo es de recuperación o actualización.

Factores a Optimizar: Espacio de almacenamiento, tiempos de respuesta, accesos a disco, etc.

Características del archivo: Si es de gran volumen, de crecimiento rápido, si se alimenta de muchos registros (tamaño, volatilidad y crecimiento), etc.

3.2.5.2 Tipos de organización

Organización Consecutiva: los registros se sitúan uno a continuación del otro. Se divide en: serial, los cuales no tienen orden lógico, y secuencial que sí tienen orden (clave de organización). Su método de acceso es secuencial. Se puede utilizar cuando se requiera una recuperación de archivos por lotes, carga masiva de datos o cuando se tienen tablas pequeñas.

Organización Direccionada: Se almacenan los registros de acuerdo a una relación establecida entre el valor de la clave de direccionamiento y la dirección física del registro, la ubicación del registro se obtiene por medio de un algoritmo de transformación sobre la clave de direccionamiento. El método de acceso puede ser secuencial ordenado o directo. Se divide en directas y dispersas.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

- Directas: La clave de direccionamiento no tiene valores duplicados. Los registros siempre están ordenados por la clave de direccionamiento, con un solo acceso a memoria secundaria se localiza el registro.
- Dispersas: Distintas claves pueden tener la misma dirección base. Los registros se les llama sinónimos.

El algoritmo de transformación consta de los siguientes *pasos*:

1. Si la clave de direccionamiento es alfabética tenemos que convertirla a numérica.
2. Convertir la clave en un número ordinal (dirección base) dentro del espacio de direccionamiento a través de una función de transformación.
3. Transformar el ordinal en la dirección física que ocupará el registro.

3.3 EJECUCIÓN Y CARGA

Durante la fase de puesta en ejecución se instala el DBMS, se crea la base de datos y se cargan los datos. En los DBMS más modernos, tales como DB2 de IBM, Server de Microsoft u Oracle, la ejecución de una nueva base de datos requiere la creación de construcciones relacionadas con el almacenamiento, especiales para alojar las tablas del usuario final.

Dichas construcciones pueden ser el grupo de almacenamiento, el espacio de tablas, las tablas, asignación de derechos de uso de la base de datos a un administrador, asignación de los derechos de acceso a los espacios para tablas y a las tablas.

Una vez que se crea la base de datos deben de cargarse los datos en las tablas. Si los datos proceden de aplicaciones anteriores o se tienen en un formato diferente deben convertirse antes de ser guardados, aunque muchos DBMS dan facilidades para la migración de datos.

En esta fase es importante abordar el desempeño, la seguridad, respaldo y recuperación y el control de concurrencia. El desempeño de una base de datos puede variar de acuerdo al hardware y software empleado, el tamaño de la base de datos así como la configuración de los parámetros del sistema y bases de datos, tal como la ubicación de los datos, la ruta de acceso, el uso de índices y el tamaño de la memoria intermedia.

La seguridad es un tema muy importante dentro de los sistemas de información y no puede ser dejado a un lado en el desarrollo de una base de datos y se debe tener una seguridad en nivel físico, esto es, no permitiendo el acceso a personal no autorizado al lugar físico (servidor) donde se encuentra la base de datos.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

La asignación de la seguridad de contraseña asignando los derechos adecuados para el acceso a la base de datos de los diferentes usuarios, que incluso puede darse esta seguridad a nivel de sistema operativo. Los derechos de acceso pueden restringir operaciones para crear, actualizar borrar registros e incluso manipulación de los objetos de la base como tablas, vistas, etc.

En ocasiones puede llegar a perderse información por un uso incorrecto de la base de datos, fallas de corriente eléctrica, etc. Los procedimientos de respaldo y recuperación de datos crean una válvula de seguridad que permite al administrador de la base de datos garantizar la disponibilidad de datos consistentes.

Durante las primeras etapas de la puesta en ejecución el sistema pasa por una fase de codificación, realización de pruebas y depuración, hasta que está listo para ser entregado.

3.4 ETAPA DE PRUEBAS Y EVALUACIONES

La fase de prueba no sirve para demostrar que no hay fallos, sirve para encontrarlos; también para demostrar que los programas parecen trabajar tal y como se especificaba en los requisitos y que las prestaciones deseadas parecen obtenerse, así como medir la fiabilidad y la calidad del software desarrollado.

La fase de pruebas puede ocurrir en paralelo con la programación de aplicaciones. Se prueba y valida el sistema con los requisitos especificados por los usuarios. Para ello, se deben cargar los suficientes datos reales para inmediatamente dar comienzo a las pruebas de la base de datos. Una vez que los datos son cargados, el DBA los debe someter a prueba y afinar la base de datos en cuanto a desempeño, integridad, acceso, concurrente y restricciones de seguridad con el objeto de poder ir ajustando la estructura física e incluso, a veces la estructura lógica, con el fin de optimización.

Si la ejecución de la base de datos no satisface las pruebas, pueden considerarse algunas opciones para mejorar el sistema:

- En cuanto al desempeño, el diseñador debe considerar la afinación del sistema específico y los parámetros de configuración del DBMS, siendo los manuales de referencia técnicos del hardware y software las mejores fuentes de información.
- Modificar el diseño físico, por ejemplo el uso apropiado de índices tiende a ser particularmente efectivo para facilitar los movimientos del puntero, con lo que mejora el desempeño.
- Modificar el diseño lógico.
- Mejorar o cambiar el software del DBMS o la plataforma de hardware o ambos.

3.5 OPERACIÓN

Una vez que la base de datos ha pasado la etapa de evaluación, se considera operativa, en donde su administración, sus usuarios y programas de aplicación constituyen un sistema de información completo.

Una vez en esta fase, los problemas que pudieran no haber sido previstos en la fase de prueba pueden comenzar a manifestarse, pudiendo ser problemas menores o en su defecto ser suficientemente serios como para considerar un trabajo de corrección. Aunque una base de datos esté muy bien diseñada, no está exenta de fallas, por ejemplo, en una base de datos que interactúa con la web puede atascarse debido al inmenso volumen de transacciones, en este tipo de casos los administradores tendrán que identificar la causa del problema y producir soluciones alternas. La demanda de cambios conduce a la fase de mantenimiento y evolución.

3.6.MANTENIMIENTO Y EVOLUCIÓN

Una vez que el sistema está completamente implementado y probado, se pone en marcha. El sistema está ahora en la fase de mantenimiento y el administrador de la base de datos debe estar preparado para realizar actividades de mantenimiento de rutina, algunas de esas actividades periódicas de mantenimiento son:

- Monitorización de las prestaciones del sistema. Si las prestaciones caen por debajo de un determinado nivel, puede ser necesario reorganizar la base de datos.
- Mantenimiento y actualización del sistema. Cuando sea necesario, los nuevos requisitos que vayan surgiendo se incorporarán al sistema, siguiendo de nuevo las etapas del ciclo de vida que se acaban de presentar.
- Mantenimiento preventivo (respaldo).
- Mantenimiento correctivo (recuperación).
- Mantenimiento de adaptabilidad (mejora de desempeño, adición de entidades y atributos, etc.)
- Asignación de permisos de acceso y su mantenimiento para usuarios viejos y nuevos.
- Generación de estadísticas de acceso a la base de datos para mejorar la eficiencia y utilidad de las auditorías del sistema y para monitorear el desempeño del sistema.
- Auditorías de seguridad periódicas basadas en estadísticas generadas por el sistema.
- Resúmenes de uso de sistema (mensuales, trimestrales y anuales) propósitos facturación interna o diseño de presupuestos.

CAPITULO III. CICLO DE VIDA DE LAS BASES DE DATOS

La única constante en el sistema es el cambio, dada la probabilidad de requerimientos de información nueva, la demanda de reportes adicionales y nuevos formatos de consulta, se requerirán cambios de aplicación y, posiblemente, cambios menores en los componentes y contenido en la base de datos.

Siguiendo las fases mencionadas se puede desarrollar una base de datos que sea funcional, eficiente y de fácil mantenimiento al paso de tiempo. Sin embargo, como todo en la vida, la teoría puede quedarse muy corta cuando nos enfrentamos a un proyecto real donde los tiempos para realizar a fondo cada una de las fases del ciclo de vida de las bases de datos o del sistema que la explotará no son los que uno como diseñador y programador quisiera o el presupuesto con el que se cuenta limita la elección de un hardware o software que no sea el mejor que se pueda considerar o incluso, puede ser que la empresa o institución ya cuenta con cierto hardware y software teniendo que adaptar el diseño a estos. A pesar de estos inconvenientes que puede y llegan a presentarse, se debe de tratar de seguir la metodología de diseño elegida lo más fielmente posible haciéndole ver a los usuarios finales que el tomarse el tiempo para realizar todas las fases de desarrollo tendrá como consecuencia la construcción de un mejor sistema que les permita realizar sus tareas diarias de manera más eficiente.

Capitulo IV.
**SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO
DEL IMP.**

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

El diseño es una de las etapas del ciclo de los sistemas más importante, debido a que es aquí en donde se usa toda la información que se recolectó durante el análisis para diseñar la base de datos que almacenará aquellos datos determinados en los requerimientos del sistema, y como bien sabemos, una base de datos bien organizada es fundamental para cualquier sistema de información.

Tal y como mencionamos en el capítulo tres, la creación de una Base de Datos debe realizarse a través de una serie de pasos que permitirán conformar un modelo lógico y físico de Base de Datos sólido, flexible a cambios futuros y sobre todo, que cubra con las necesidades que requiere el usuario. En éste capítulo se mostrará como se desarrolló la Base de Datos de acuerdo al ciclo de vida mostrado en el capítulo dos del presente trabajo.

De igual forma se mostrará el procedimiento que se siguió para el desarrollo del sistema de pantallas que interactúan directamente con la Base de Datos para explotar toda la información contenida en ella pero también con la posibilidad de introducir información nueva si así lo requiere el usuario.

4.1 ESTUDIO INICIAL DE LA BASE DE DATOS.

En los siguientes párrafos se mostrará la situación y operatividad del posgrado dentro del Instituto Mexicano del Petróleo y de ésta manera poder determinar los problemas y restricciones, definir objetivos alcances y límites para la Base de Datos.

El posgrado del IMP ha de atender a las necesidades de la sociedad, en particular de la industria petrolera y química, y ser un factor de promoción de la investigación y la innovación tecnológica en nuestro país.

A lo largo de más de treinta años de servicios el Instituto se ha constituido en una alternativa de atención a las necesidades de la industria, con el desarrollo de actividades de investigación y la prestación de servicios. De este modo se ha conformado un acervo de experiencias y capacidades que hoy habrán de servir a la formación de un programa de posgrado que amplíe las posibilidades de responder a las demandas crecientes de la sociedad mexicana.

La estructura operativa del Instituto se sustenta en cuatro plataformas, de Investigación y Desarrollo, de Soluciones, de Atención a Clientes y de Competencias. La actividad se desarrolla por proyectos, de los cuales, son relevantes para el posgrado los de investigación, los de desarrollo de productos y los de aplicación industrial. En este contexto, la estructura del posgrado está planteada para que el estudiante se beneficie directamente en su formación de las plataformas operativas y de toda la actividad que se realiza cotidianamente en el Instituto, a través de su participación en los proyectos.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

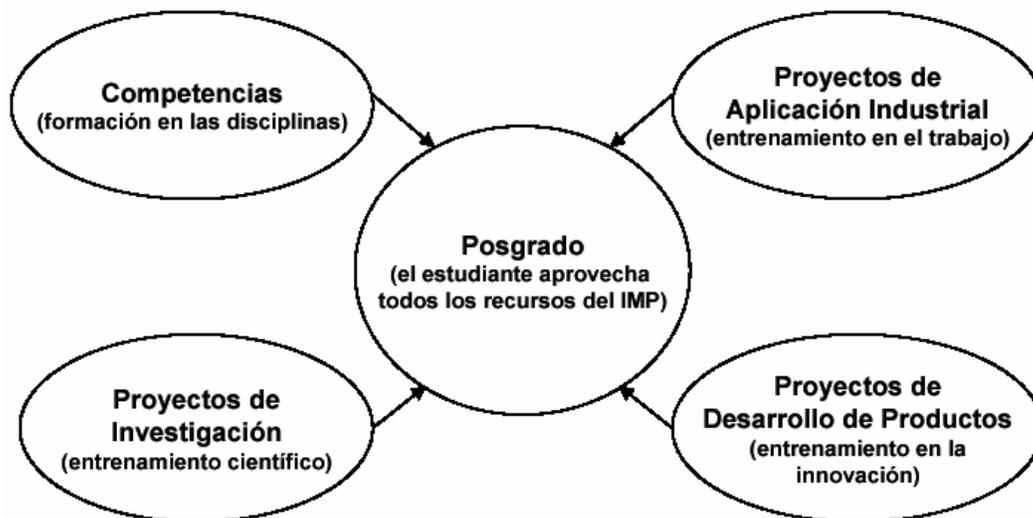


Fig. 4.1. Estructura Operativa del Posgrado IMP

La organización curricular del plan de estudios está conformada a partir de Áreas Temáticas, en tres niveles, especialidad, maestría y doctorado y contempla tanto el tránsito secuencial entre los niveles, como el ingreso directo a cualquiera de ellos. Fig. 4.1

Las Áreas Temáticas corresponden con aquellas áreas de actividad del Instituto en las que se cuenta con:

- I. Un núcleo sólido de investigadores con doctorado, con experiencia en el desarrollo de proyectos de investigación de carácter científico o tecnológico y con experiencia para impartir cursos y dirigir tesis de posgrado.
- II. Un núcleo sólido de especialistas con experiencia en el desarrollo de proyectos de aplicación industrial, de desarrollo o mejora de productos, o de adaptación y transferencia de tecnología.
- III. Proyectos en curso de investigación, de desarrollo de productos y de aplicación industrial, en los que puedan participar los estudiantes de posgrado.
- IV. Infraestructura para la realización de los proyectos.

El estudiante, junto con su Comité de Tutores, seleccionará los cursos y los proyectos (de aplicación industrial, de desarrollo de productos y de investigación), de acuerdo con el Área Temática elegida.

Quienes tienen la capacidad para formar en tal nivel a los estudiantes son los científicos y especialistas que se desempeñan en tareas de investigación y servicios que, por su naturaleza, han de generar nuevo conocimiento, significativo en el ámbito de sus disciplinas. El contacto con los científicos y especialistas se da en las distintas unidades que conforman el plan de estudios.

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

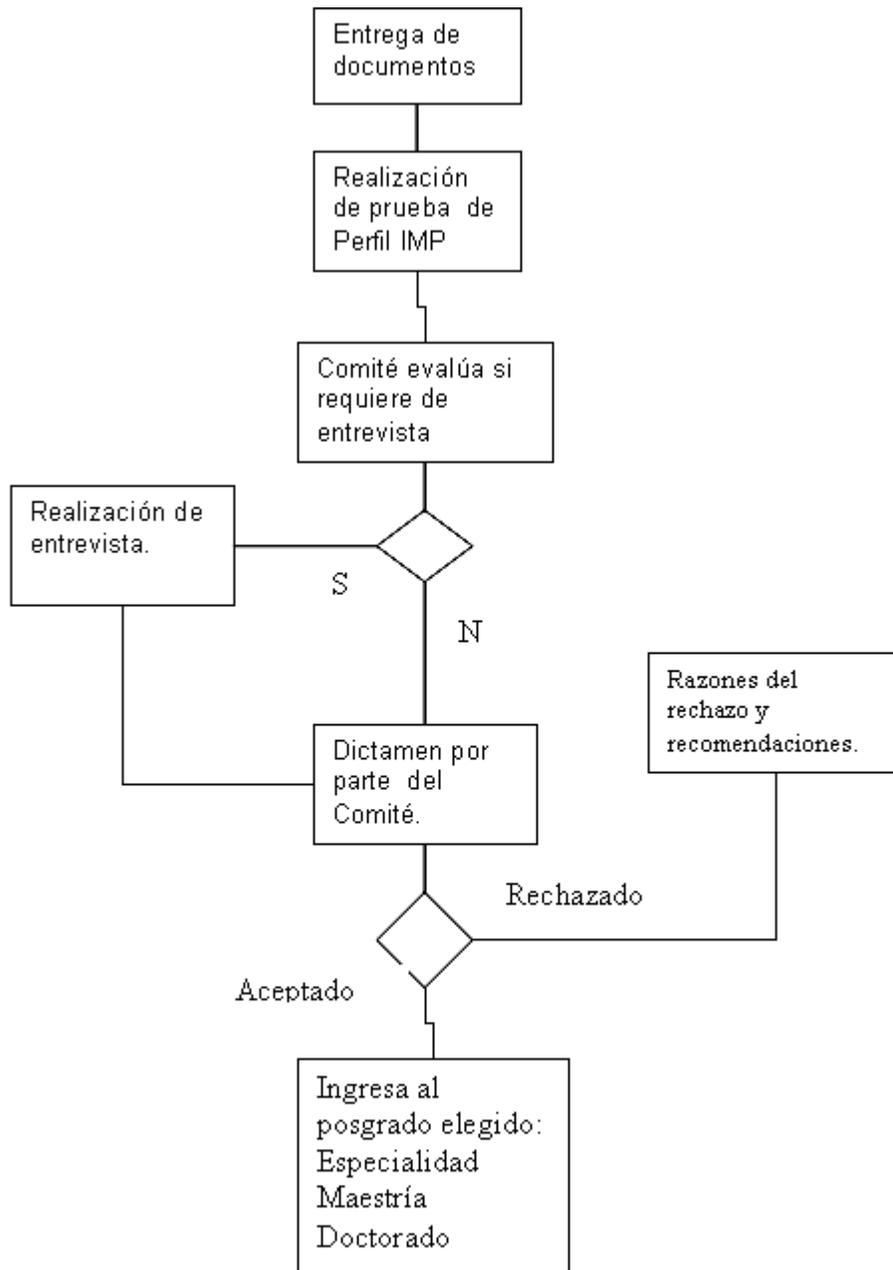


Fig. 4.2 Proceso de Admisión Posgrado IMP

Estas unidades son:

- a. Cursos orientados al aprendizaje de conocimientos básicos y de frontera, de los diversos campos disciplinarios que convergen en las diferentes Áreas Temáticas y que son fundamentales para la formación científica.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

b. Participación en proyectos de investigación, de desarrollo de productos y de aplicación industrial, orientados al aprendizaje de las habilidades de uno y otro campo, de modo que el estudiante integre una perspectiva del proceso de innovación, desde la prueba de principio, hasta la aplicación a la práctica y el resultado industrial. Adicionalmente, los proyectos de aplicación industrial, por su naturaleza, son espacios propicios para el adiestramiento del estudiante en las actividades propias del ámbito laboral.

c. Seminarios de integración por Área Temática en los que participan estudiantes y profesores, orientados:

1. Al adiestramiento en la comunicación oral y escrita.
2. Al ejercicio de la reflexión colectiva.
3. A la exposición de las experiencias de aprendizaje en los cursos y en los proyectos.
4. A la presentación de los avances de investigación.

d. Actividades complementarias como cursos cortos, conferencias y talleres, orientados al aprendizaje de los lenguajes y formas de pensamiento de otros campos disciplinarios, ajenos al propio.

e. En los niveles de maestría y doctorado, la elaboración y defensa ante un jurado del protocolo de un proyecto de aplicación industrial, o de desarrollo de productos o de investigación, orientado al adiestramiento en la concepción y diseño de proyectos.

f. En el nivel de doctorado la realización y defensa ante un jurado de una tesis, de la cual resulten publicaciones o patentes, orientada a la formación del estudiante como investigador capaz de generar y aportar por sí mismo nuevos conocimientos científicos o tecnológicos.

De esta forma, el modelo educativo del posgrado contempla la educación formal y la educación en la acción. fig. 4.3.

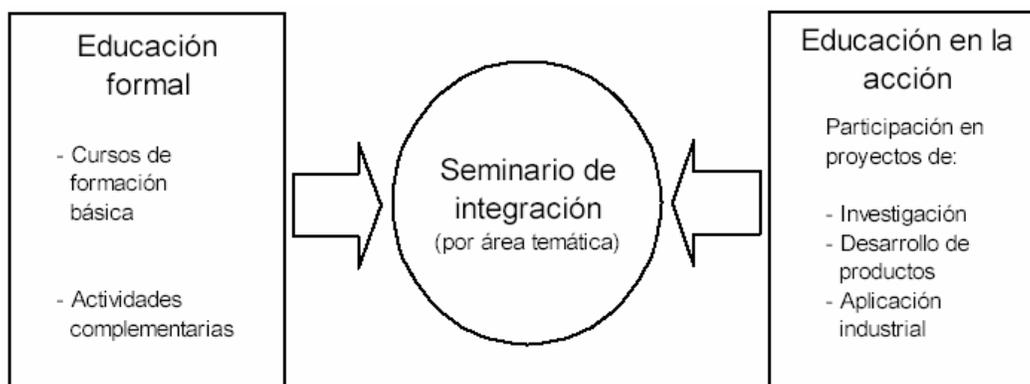


Fig. 4.3 Modelo educativo del Posgrado IMP.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

El conjunto de las actividades mencionadas se integra de diversa forma para cada uno de los grados académicos, con la finalidad de garantizar una estancia mínima en los programas, así como un alto nivel de desempeño, fig. 4.1.4.

De esta forma, el programa de especialidad contempla una permanencia mínima de un año, en que el estudiante habrá de participar en tres proyectos de diferente naturaleza, uno de investigación, uno de desarrollo de productos y uno de aplicación industrial. Así mismo habrá de aprobar un mínimo de seis cursos en las disciplinas básicas del área de especialidad y de participar en el seminario de integración que le corresponda. Durante su estancia en el Instituto sus actividades serán acompañadas por un asesor.

Al igual que en la especialidad, el programa de maestría contempla una permanencia mínima de un año, así como la participación en los tres tipos de proyecto, uno de investigación, uno de desarrollo de productos y uno de aplicación industrial. El estudiante de maestría habrá de aprobar ocho cursos en las disciplinas básicas del área de especialidad y de participar en el seminario de integración que le corresponda. Para la obtención del grado el estudiante habrá de elaborar y defender ante un jurado idóneo un protocolo de un proyecto ejecutable, cuya naturaleza puede ser de investigación, de desarrollo de productos o de aplicación industrial. Durante su estancia en el Instituto sus actividades serán acompañadas por un comité de tutores, quién colaborará en la selección de los cursos básicos y los proyectos. Para la elaboración de la planeación del protocolo contará con la colaboración de un asesor y un comité de lectores.

El programa del doctorado tiene un carácter no escolarizado y contempla una estancia mínima de dos años en el Instituto, ya que se puede acceder a él en forma directa. El estudiante sólo tendrá la obligación de aprobar cursos en las disciplinas básicas en caso de que su comité de tutores lo considere necesario para completar su formación. Durante su estancia en el Instituto el estudiante habrá de participar en el seminario de integración que le corresponda y, si no lo ha hecho ya, tendrá que participar en los tres tipos de proyecto, uno de investigación, uno de desarrollo de productos y uno de aplicación industrial. Para obtener el grado de doctor, el estudiante habrá de elaborar y desarrollar un protocolo de investigación en el ámbito científico o de desarrollo de productos que culmine en una tesis. Tanto el proyecto como la tesis habrán de ser defendidos ante un jurado idóneo. Así mismo, habrá de publicar artículos de investigación o someter a registro patentes. A lo largo de su permanencia en el Instituto sus actividades serán acompañadas por un comité de tutores, quién colaborará en la selección de los proyectos e integrará al estudiante a un grupo de investigación a su cargo. Para la elaboración del protocolo y desarrollo del proyecto de investigación el estudiante contará con la colaboración de un asesor y un comité de tutores.

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

Nivel formativo	Primer año académico			Segundo año académico			Tercer año académico		
	Primer semestre	Segundo semestre	Inter semestre	Tercer semestre	Cuarto semestre	Inter semestre	Quinto semestre	Sexto semestre	Inter semestre
Especialidad	2 cursos de formación básica	2 cursos de formación básica	Actividades complementarias	2 cursos de formación básica					
	Seminario de integración	Seminario de integración		Seminario de integración					
	Participación en proyecto	Participación en proyecto		Participación en proyecto					
Maestría	2 cursos de formación básica	2 cursos de formación básica	Actividades complementarias	2 cursos de formación básica	2 cursos de formación básica	Actividades complementarias			
	Seminario de integración	Seminario de integración		Seminario de integración	Seminario de integración				
	Participación en proyecto	Participación en proyecto		Participación en proyecto					
				Elaboración y defensa ante un jurado de un protocolo para la realización de un proyecto de aplicación industrial, de desarrollo de productos o de investigación.					
Doctorado	Seminario de integración	Seminario de integración	Actividades complementarias	Seminario de integración	Seminario de integración	Actividades complementarias	Seminario de integración	Seminario de integración	Actividades complementarias
	Participación en proyecto	Participación en proyecto		Participación en proyecto					
	Elaboración y defensa ante un jurado de un protocolo para la realización de un proyecto de desarrollo de productos o de investigación.								
	Elaboración y defensa ante un jurado de una tesis que constituya una contribución científica o tecnológica original.								

Fig. 4.4 Planeación del Posgrado

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

4.1.1 Definición del problema y restricciones.

El Instituto Mexicano del Petróleo no cuenta con las herramientas necesarias para llevar un control adecuado de todos los alumnos que aspiren a entrar o ingresen a los diferentes posgrados impartidos. Si se hace el uso de papel para llevar el registro de cada aspirante o alumno, se tiene el riesgo de extravío de los documentos, el no llevar un orden en preciso del historial de los aspirantes o alumnos, y un costo constante en la adquisición de la papelería necesaria.

El llevar el registro en hojas de calculo puede llevar a la redundancia de información, el uso de infinidad de archivos debido a la cantidad información que se requiere tener e incluso el peligro de daño del archivo que contenga la información, y la información no podría estar disponible de manera fácil y rápida para los usuarios de nivel ejecutivo para la toma de decisiones adecuadas.

El acceso a la información no está controlado debidamente, ya que tanto la documentación como los archivos podría llegar a manos de personal que no tenga los privilegios necesarios para tener dicha información con el peligro inminente del mal uso de la misma.

4.1.2 Definición de los objetivos, alcances y límites.

Una vez que se ha estudiado la problemática que se tiene, podemos tener los objetivos del desarrollo de una base de Datos. El Instituto Mexicano del Petróleo requiere de una Base de Datos que permita llevar el registro de cada estudiante que ingrese al posgrado así como su historial a lo largo de su estancia en el Instituto; dicho registro deberá ser almacenado de manera ordenada, accesible en cualquier momento por el usuario, debe mantener una estrecha relación con los proyectos ya existentes y con el personal que labora en el Instituto, los cuales podrán ser tutores o asesores en un momento dado de los estudiantes y poder generar reportes claros y de manera inmediata. Tener un control absoluto sobre quien debe tener acceso a la Base de Datos con los respectivos niveles de acceso para consultar, introducir o cambiar la información de acuerdo a los perfiles establecidos.

La Base de Datos deberá almacenar la información personal y profesional de cada aspirante que desee ingresar al posgrado. Una vez que sea aceptado, se debe llevar el registros de cada actividad académica que tome el alumno, el historial de calificaciones obtenidas y los proyectos del instituto en los cuales tomen parte activa. Se debe tener el control de los profesores que impartirán las actividades académicas y los diferentes tutores que tendrán los alumnos durante toda su estancia en el posgrado. Se necesita contar con el temario, bibliografía y software a utilizar para cada materia y la posibilidad de anexar, eliminar o modificarlos de acuerdo a las necesidades que se presenten.

Para dicha Base de Datos se cuenta ya con el DBMS a utilizar y el servidor que contendrá la información. Tanto el DBMS como el servidor ya se encuentran operando con otros sistemas que tiene el Instituto, por lo cual se utilizarán para la creación de la Base de Datos de posgrado. El instituto cuenta también con un DBA que se hace cargo del DBMS, por lo cual se nos dieron los permisos necesarios para poder diseñar y montar la Base de Datos en dicho DBMS.

4.2 DISEÑO DE LA BASE DE DATOS.

En ésta etapa se muestra el modelo conceptual de la Base de datos creado a partir del análisis hecho en el punto anterior. Es aquí donde se hace uso de los dos enfoques señalados en el capítulo tres, enfoque lingüístico de Chen y el enfoque de categorización de objetos de Carswell y Navathe, para crear un modelo conceptual en base al análisis realizado en el punto anterior de éste capítulo.

4.2.1 Modelo Conceptual

La primera parte que se modelo fue referente al aspirante a ingresar al posgrado y toda aquella información que pudiera ser de utilidad para el comité encargado de realizar las evaluaciones y admisiones al posgrado.

La actividad académica es tomada como una entidad debido a que tiene una serie de propiedades relacionadas directamente entre sí, como es su nombre, objetivo, número de créditos y fecha de registro, Fig.4.5.



Actividad Académica

Fig. 4.5 Representación de la entidad actividad académica

Una actividad académica tiene asociado software específico para brindar al estudiante las herramientas necesarias para desempeñarse de una mejor manera dentro de la actividad académica o tener una bibliografía determinada que sirva de referencia para que el alumno pueda consultarla durante el curso de la actividad. Las actividades pertenecen a un tipo de actividad genérica, o puede estar dividida en diferentes temas y a su vez cada tema contiene subtemas.

Cada uno de estos sustantivos se toman como entidades debido a que tienen un conjunto de atributos más allá de su propio nombre, por ejemplo, el software tiene un nombre, versión o sistema operativo donde funciona; o un tema puede tener uno o más subtemas asociados. Hasta éste momento ya se tiene identificadas diferentes entidades que relacionadas entre sí, ahora hay que detallar que tipo de relación existe entre las entidades, lo que se conoce como cardinalidad.

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

En éste caso la cardinalidad entre las entidades “Actividad Académica” y “Software” es de muchos a muchos, lo cual significa que una actividad académica puede tener uno o más software asociado y a su vez un software puede ser asociado a una o muchas actividades académicas. En cuestión a la relación entre las entidades “Actividad Académica” y “Temas” la relación que se tiene es de uno a muchos, esto debido a que una actividad puede tener uno o muchos temas asociados, pero un tema puede pertenecer solo a una actividad académica, de igual manera un tema puede tener uno o más subtemas, pero un subtema solo puede pertenecer a un tema.

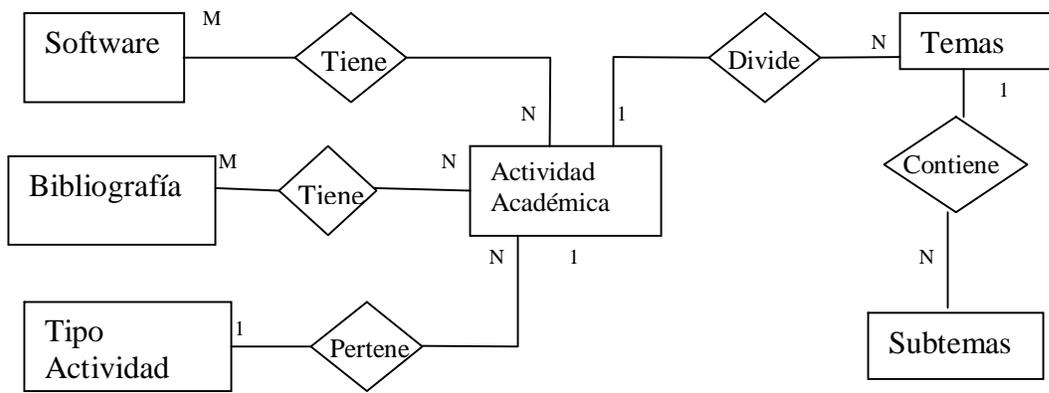


Fig. 4.6 Modelo conceptual del módulo de actividades académicas.

Parte del modelo conceptual es también el identificar los atributos que tiene cada entidad. Dichos atributos son representados por elipses que están conectadas a sus entidades por medio de una línea recta. A continuación se muestra la entidad aspirante con sus principales atributos, Fig.4.7.

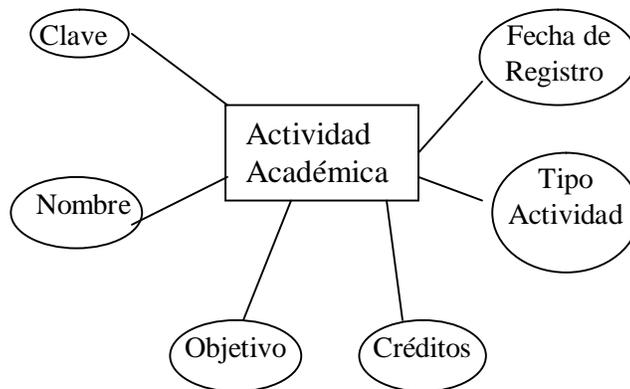
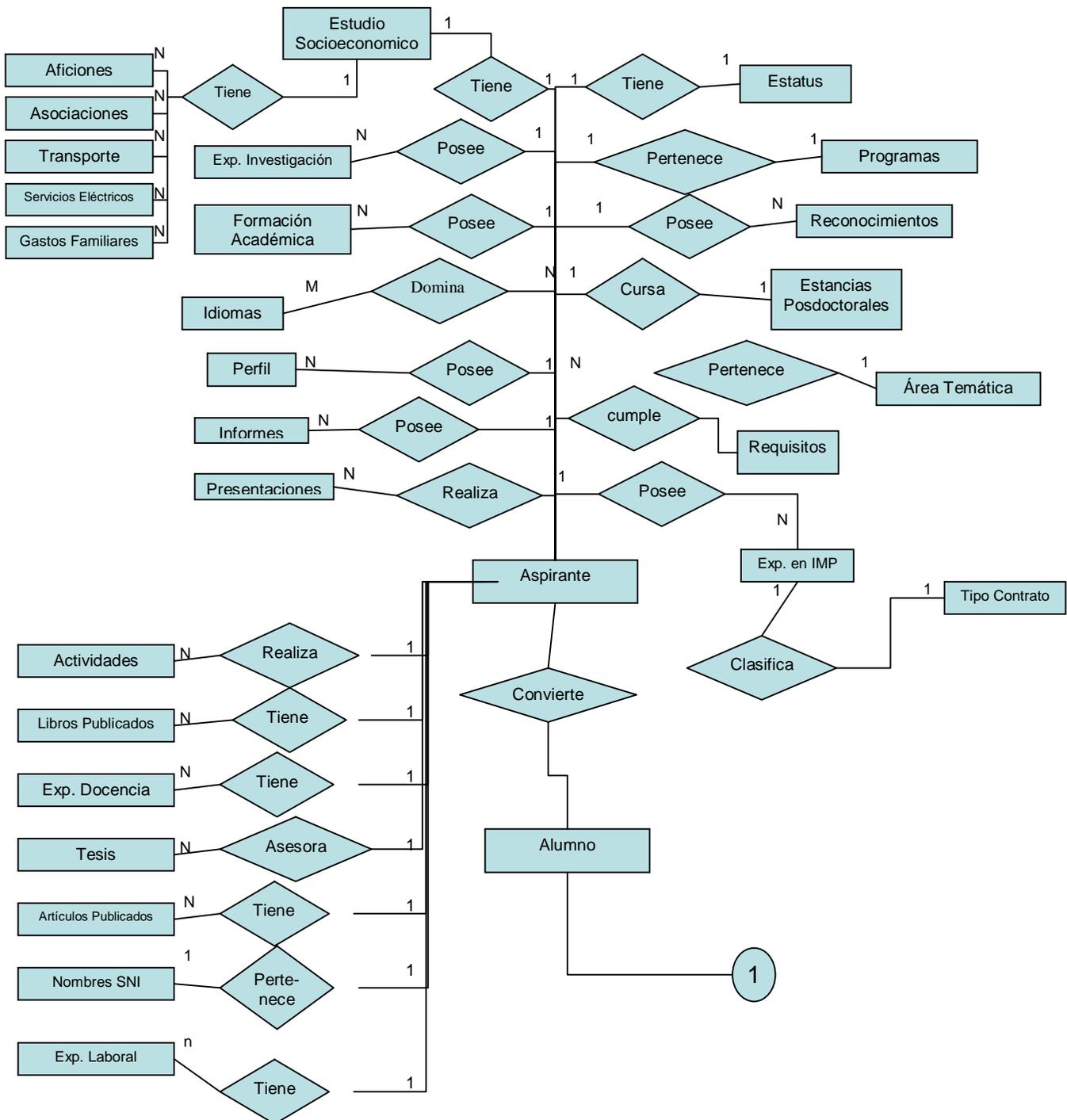


Fig. 4.7 Representación de los atributos de una entidad.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

La manera en que se llevó a cabo la transformación al modelo conceptual del análisis que ya se tenía realizado, en el caso concreto de las actividades académicas, fue utilizado para obtener el modelo conceptual de toda la Base de Datos de posgrado. A continuación se muestra el modelado conceptual, se omitieron los atributos de cada entidad debido al espacio que se necesita para generar el esquema completo.



CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

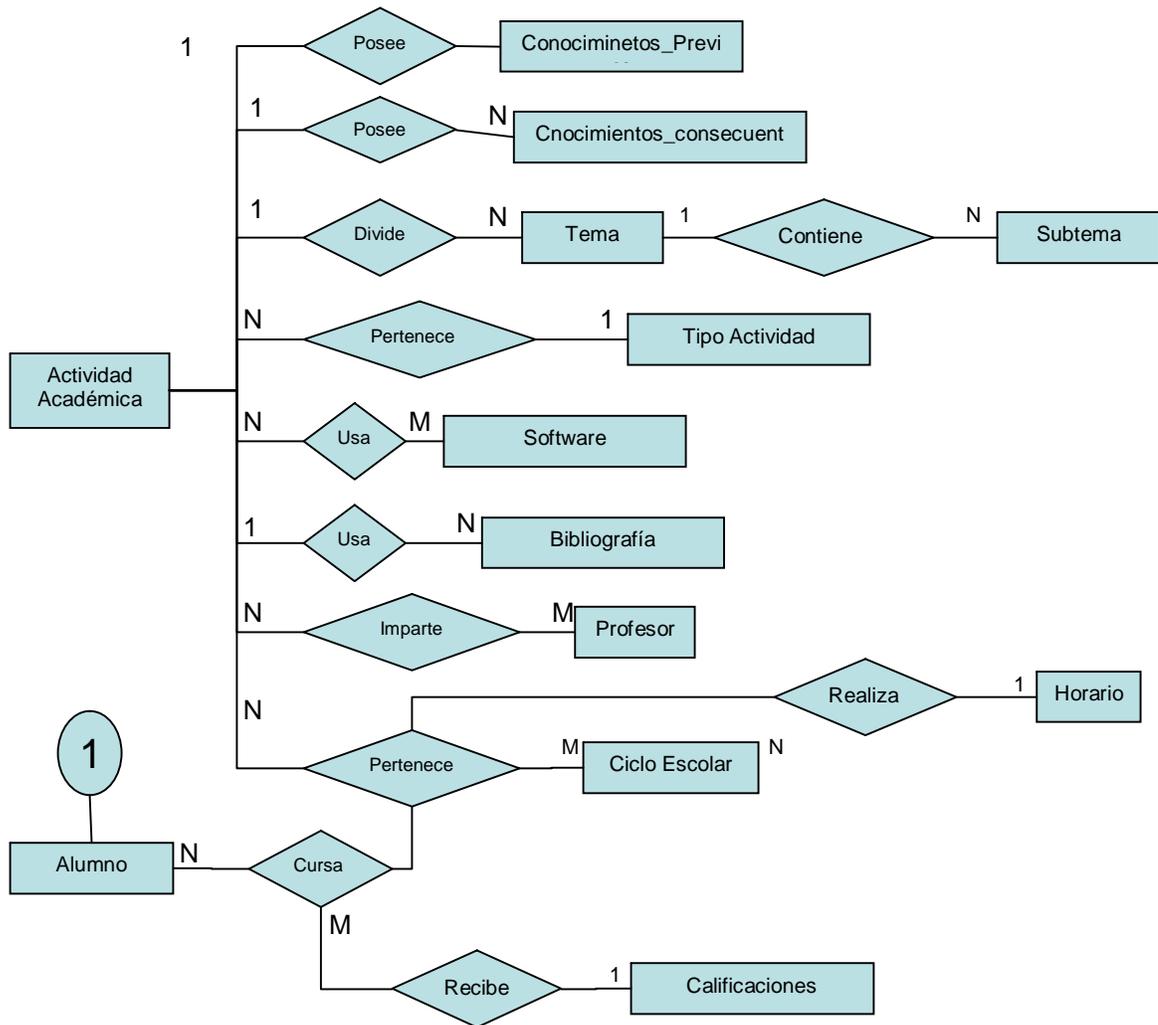


Fig. 4.8 Modelo conceptual del área de Posgrado del IMP.

Hasta el momento ya se ha logrado obtener el modelo conceptual de la Base de Datos, lo cual permite pasar a la siguiente fase, convertir dicho modelo al diseño lógico de Base de Datos que utilice el DBMS. El modelo conceptual lo podemos transformar a un diseño basado en el modelo jerárquico, de red, relacional u orientado a objetos. En nuestro caso, se adaptó a un modelo relacional debido a que utilizó el DBMS relacional Oracle 8i que es la Base de Datos institucional, también por que es modelo de Bases de Datos más utilizado actualmente por las ventajas que ofrece sobre el modelado de red o jerárquico y respecto al orientado a objetos no es soportado en esa versión de Oracle.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

4.2.2 Diseño Lógico.

Una vez que se tuvo el modelo conceptual y el SGBD a utilizar se procede a transformar dicho diagrama a un diseño lógico basado en Oracle 8i. Basándonos en el mismo ejemplo que se utilizó en el punto anterior, la transformación sería de la siguiente forma.

De acuerdo al punto 3.7.2, del capítulo tres, todas las entidades se convierten en tablas manteniendo el número y tipo de atributos, por lo tanto las entidades Actividad Académica, Bibliografía, Software, Tipo de Actividad Temas y Subtemas se convierten en tablas.

Debido a que existen varios esquemas de Base de Datos en el Instituto Mexicano del Petróleo se hace uso de una nomenclatura para saber que tablas pertenecen a las diferentes áreas, dicha nomenclatura consiste en poner una abreviatura que identifique al área seguida de un guión bajo y el nombre de la tabla. En nuestro caso se opto por identificar nuestras tablas con la abreviatura PO y el nombre de la tabla tal y como se muestra a continuación, (Fig. 4.9), con la transformación del modelo conceptual al lógico.

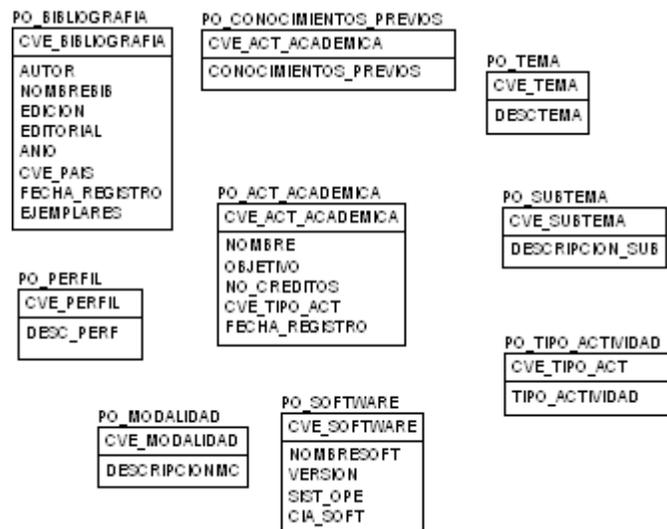


Fig. 4.9 Esquema de base de datos

La relación entre “Tipo de Actividad” y “Actividad Académica” es de uno a muchos (1:N), la manera en que resuelve ésta relación es propagando la clave del “Tipo de Actividad” hacia “Actividad Académica”; el mismo caso se aplica para las relaciones entre “Actividad Académica” y las entidades “Temas” y “Conocimientos Previos”; y la relación entre “Temas” y “Subtemas” .

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

La relación entre “Actividad Académica” y las entidades “Modalidad”, Software, Perfil y Bibliografía es de muchos a muchos (N:M) por lo tanto las relaciones son convertidas en tablas cuya llave primaria es la concatenación de las llaves primarias de ambas tablas.

Aplicando lo dicho anteriormente se observa, por ejemplo, que la relación entre las actividades académicas y bibliografía se obtiene una tabla terciaría llamada “po_actividad_bibliografia”, ésta tabla tiene una llave primaria compuesta por la llave primaria de la tabla “po_bibliografía” (cve_bibliografia) y la llave primaria de “po_actividad_academica” (cve_act_academica). Para el nombre de todas las tablas terciarias se optó por combinar también el nombre de las dos tablas que tienen la relación de muchos a muchos, sin embargo esto no es una regla, se puede poner cualquier otro nombre que identifique a dicha relación. Se muestra el esquema completo del módulo de actividades académicas (Fig. 4.10).

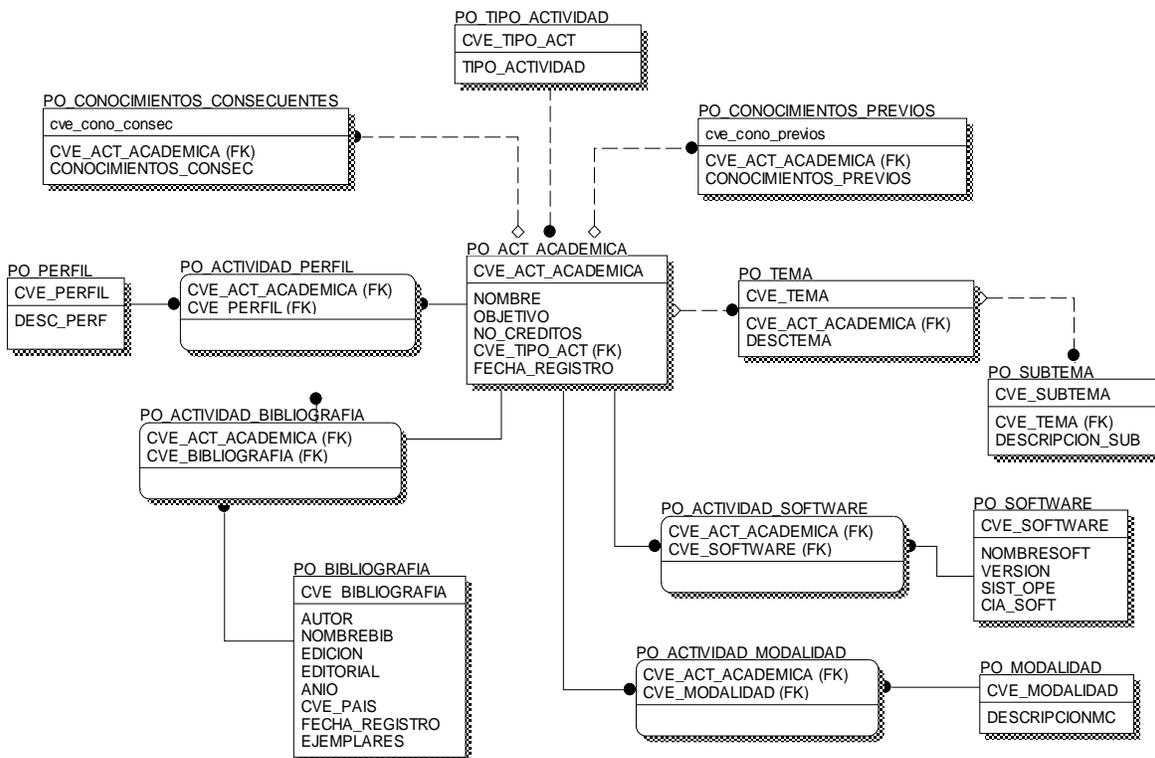


Fig. 4.10 Módulo completo

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

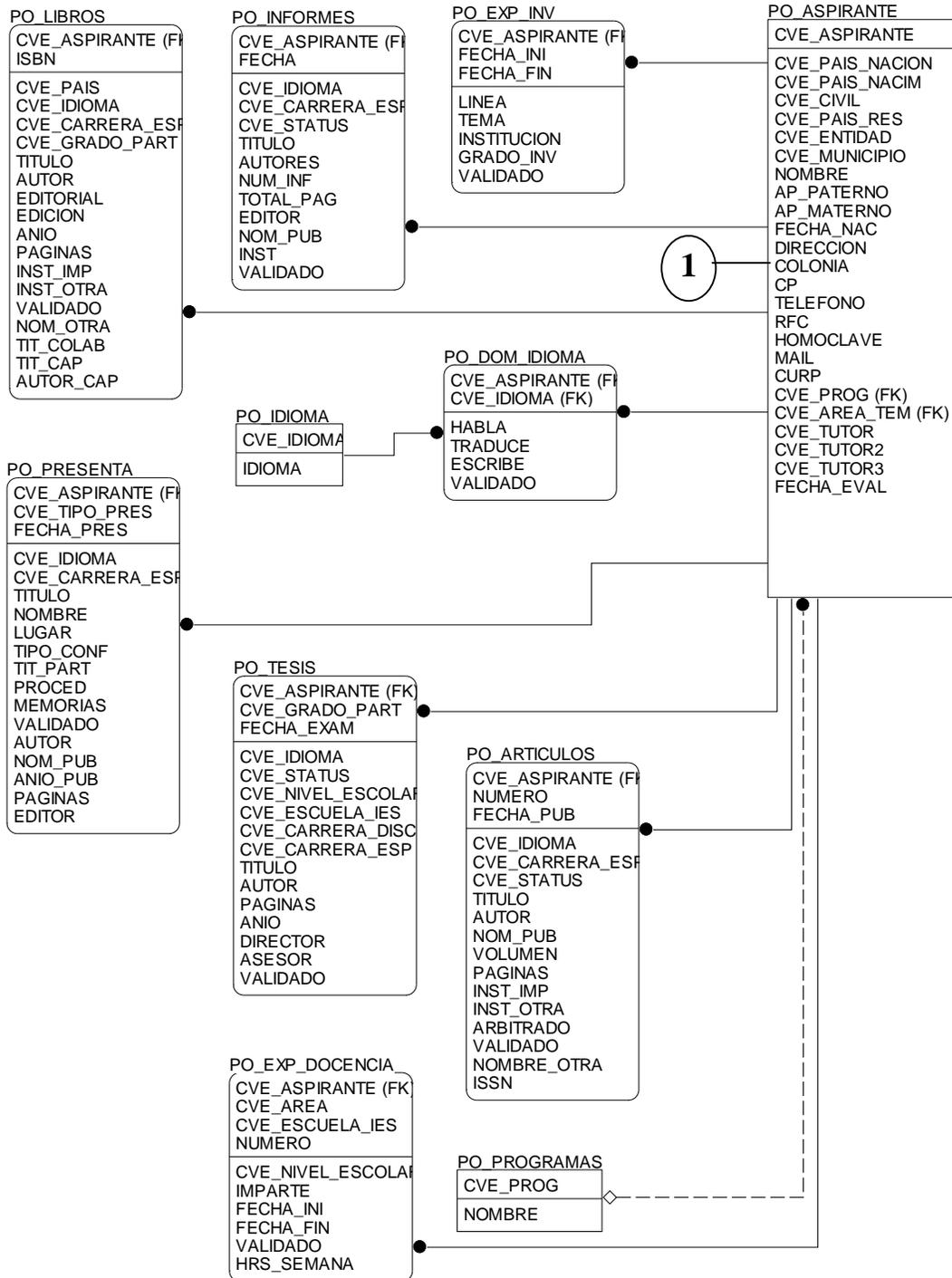


Fig 4.11b Esquema Entidad Relación.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

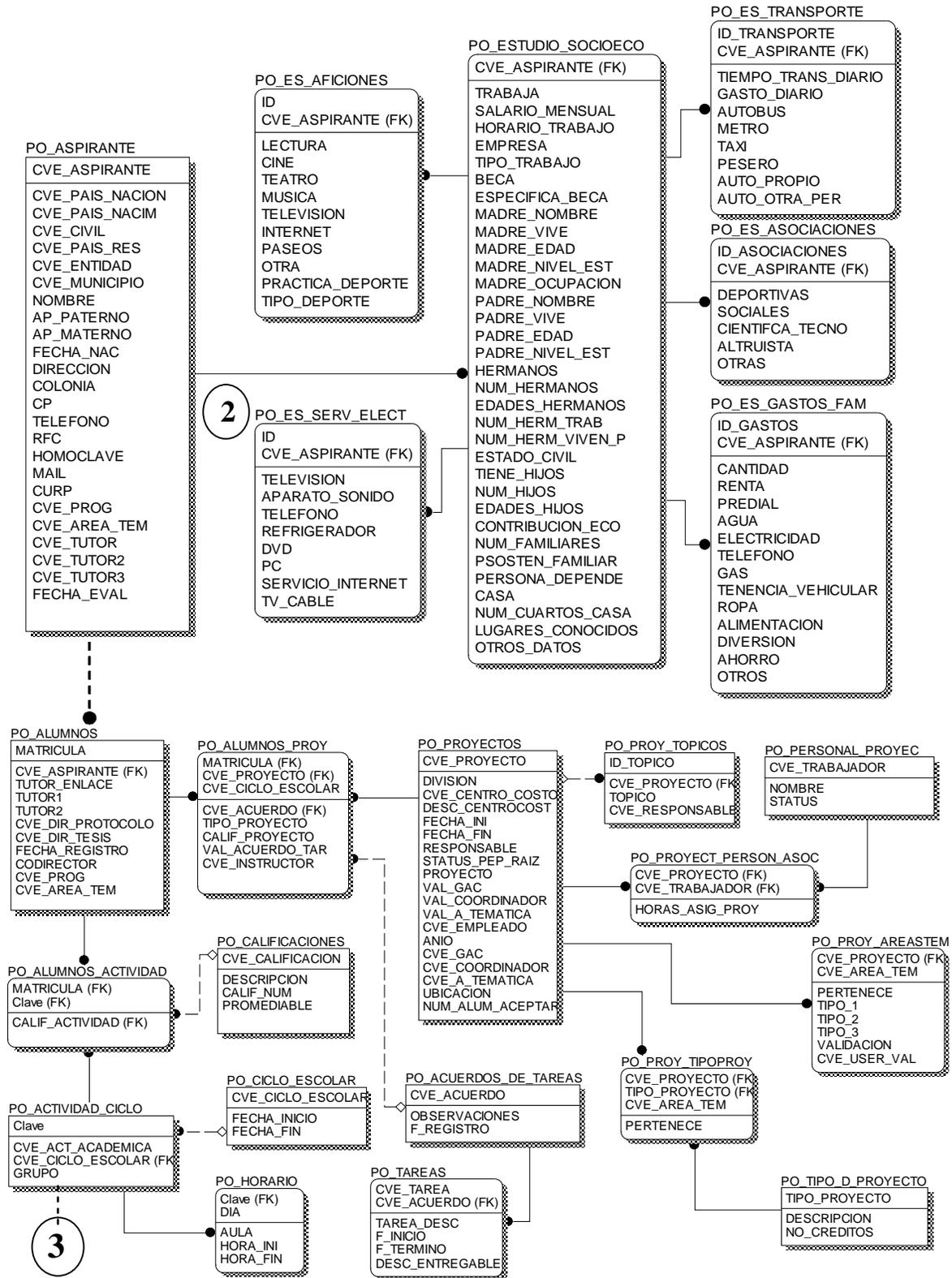


Fig 4.11c Esquema Entidad Relación.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

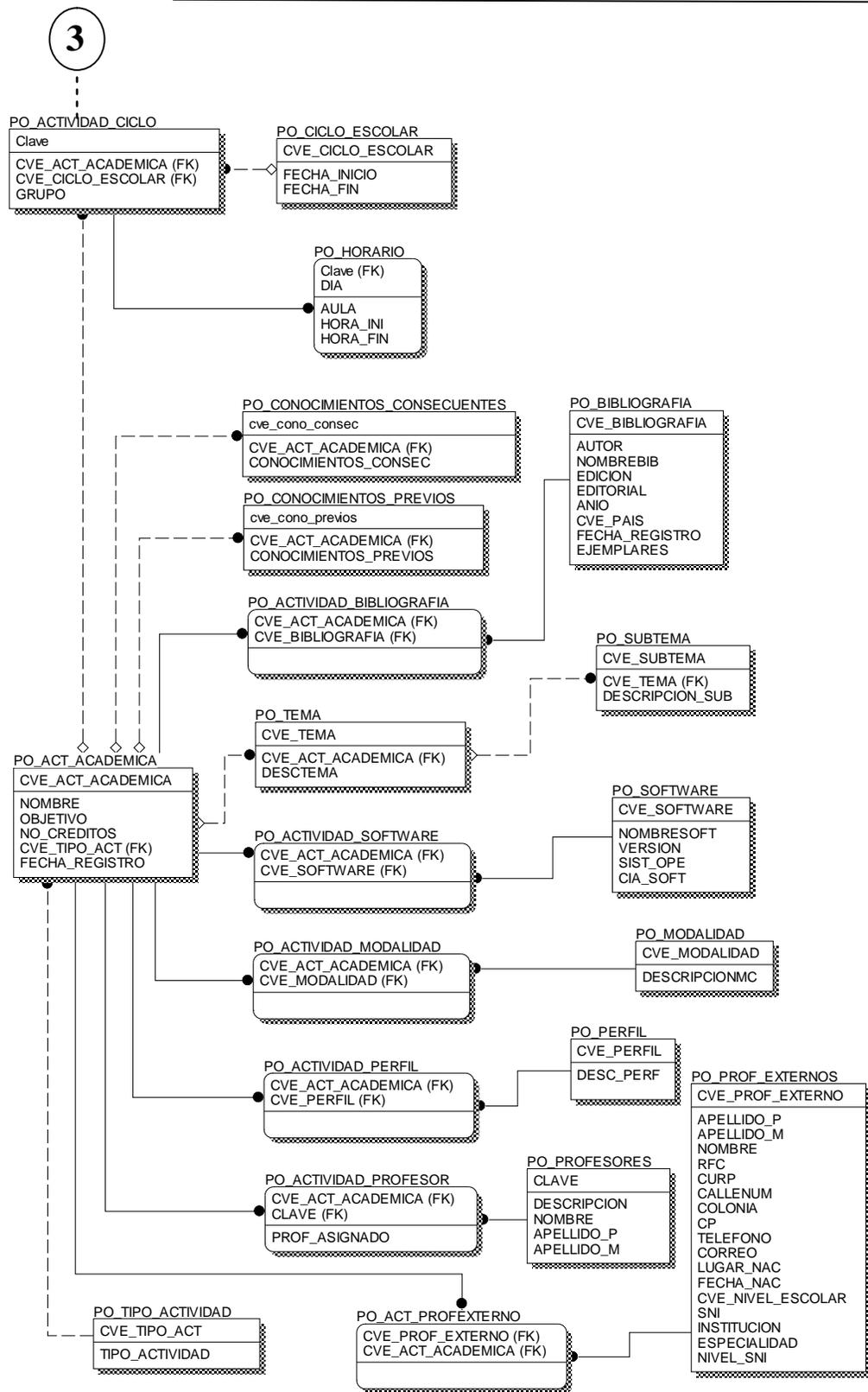


Fig 4.11d Esquema Entidad Relación.

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

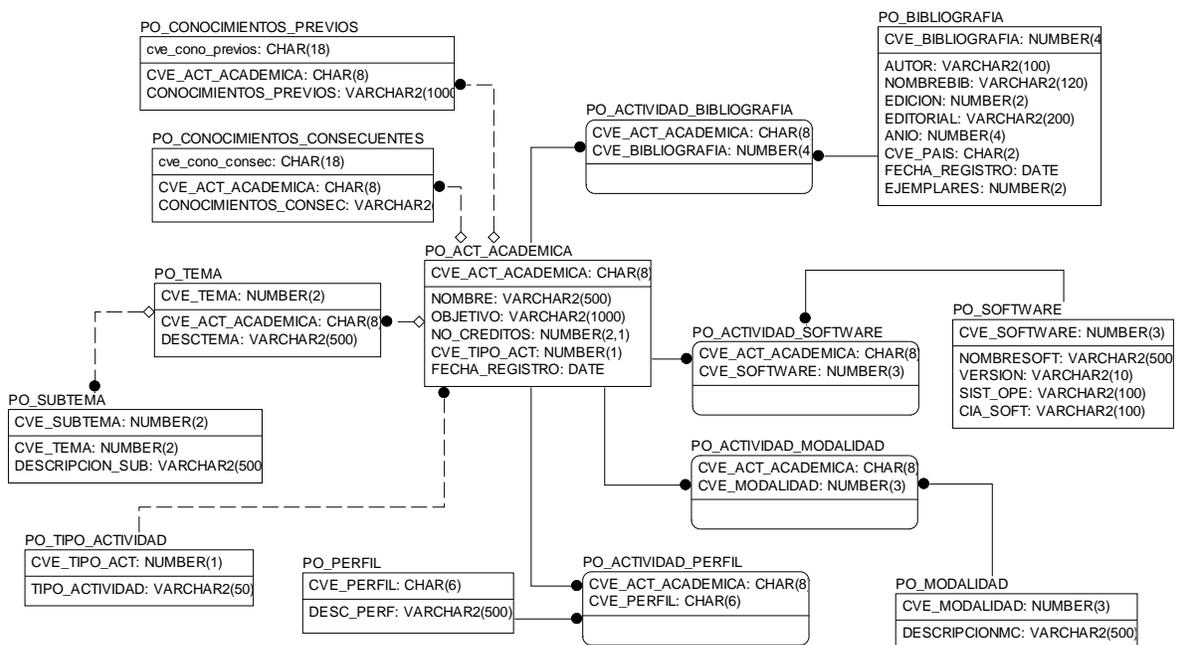
Una vez que se tiene el diseño lógico, el siguiente paso es implementarlo en el DBMS que se ha elegido o el que se tiene a disposición del proyecto. El siguiente paso, de acuerdo al ciclo de vida de Bases de Datos es el diseño físico.

4.3 DISEÑO FÍSICO, EJECUCIÓN Y CARGA.

En el diseño físico se definirán los tipos de datos que tendrán los atributos, la creación de las tablas, índices y restricciones de integridad; el espacio en disco necesario para el almacenamiento de la información a mediano y largo plazo; la asignación de memoria y la configuración adecuada del sistema operativo sobre el cual se instalará el DBMS.

La sintaxis de creación de cada objeto dentro del DBMS como tablas, índices, procedimientos almacenados, triggers, usuarios, etc., se generan a través del lenguaje SQL (Lenguaje Estructurado de Consulta), dicho lenguaje permite acceder a las Bases de Datos para recuperar la información, realizar inserciones o actualizaciones. El SQL es un lenguaje estándar, sin embargo, los diferentes DBMS's comerciales han añadido diferentes particularidades al lenguaje.

Al diseño lógico que se obtuvo se le asignaron los tipos de datos a todos los atributos, apoyados en un diccionario de datos, el cual se mostrará mas adelante. En la siguiente figura (4.12) se muestra el módulo de actividades académicas ya con los tipos de datos asignados en todos los atributos.



4.12 Módulo de actividades académicas

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

A continuación se muestra el diccionario de datos de las tablas que conforman la Base de Datos.

Nombre de la Tabla	Nombre del Campo	Tipo
PO_ACTIVIDADES	Cve_aspirante	NOT NULL NUMÉRICO (6)
	NUM_ACT	NOT NULL NUMÉRICO (2)
	ACTIVIDAD	NOT NULL VARCHAR2(100)
	FECHA_INI	DATE
	FECHA_FIN	DATE
PO_ACTIVIDAD_BIBLIOGRAFIA	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_BIBLIOGRAFIA	NOT NULL NUMBER(4)
PO_ACTIVIDAD_CICLO	ID	NOT NULL NUMBER(6)
	CVE_ACT_ACADEMICA	CHAR(8)
	CVE_CICLO_ESCOLAR	CHAR(5)
	GRUPO	NOT NULL CHAR(1)
PO_ACTIVIDAD_MODALIDAD	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_MODALIDAD	NOT NULL NUMBER(3)
PO_ACTIVIDAD_PERFIL	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_PERFIL	NOT NULL CHAR(6)
PO_ACTIVIDAD_PROFESOR	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CLAVE	NOT NULL NUMBER(6)
	PROF_ASIGNADO	NUMBER(1)

Nombre de la Tabla	Nombre del Campo	Tipo
PO_ACTIVIDAD_SOFTWARE	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_SOFTWARE	NOT NULL NUMBER(3)
PO_ACT_ACADEMICA	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	NOMBRE	NOT NULL VARCHAR2(500)
	OBJETIVO	NOT NULL VARCHAR2(1000)
	NO_CREDITOS	NOT NULL NUMBER(3,1)
	CVE_TIPO_ACT	NOT NULL NUMBER(1)
	FECHA_REGISTRO	DATE
PO_ACUERDOS_DE_TAREA	CVE_ACUERDO	NOT NULL NUMBER(4)
	OBSERVACIONES	VARCHAR2(500)
	F_REGISTRO	DATE
PO_ALUMNOS	MATRICULA	NOT NULL NUMBER(6)

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	TUTOR_ENLACE	NUMBER(6)
	TUTOR1	NUMBER(6)
	TUTOR2	NUMBER(6)
	CVE_DIR_PROTOCOLO	NUMBER(6)
	CVE_DIR_TESIS	NUMBER(6)
	FECHA_REGISTRO	DATE
	CODIRECTOR	NUMBER(6)
	CVE_PROG	CHAR(2)
	CVE_AREA_TEM	CHAR(3)
	FEC_INI_PROG	DATE
	FEC_FIN_PROG	DATE
	FEC_GRADUA_PROG	DATE
	NUM_DOC_PROG	NUMBER(3)
	CUMPLE_PAT_PUB_PROG	NUMBER(1)
	NUM_DOC_ACTA_PROG	NUMBER(3)
	FEC_CUMPLE_PROG	DATE
	CERT_TOEFL	NUMBER(3)
	PUNTOS_TOEFL	NUMBER(5)
	FEC_EXAMEN_TOEFL	DATE
	NUM_PROTOCOLO	NUMBER(2)
	NUM_TESIS	NUMBER(2)
	FEC_INI_AREA	DATE
	FEC_FIN_AREA	DATE
	FEC_INI_TUTORE	DATE
	FEC_FIN_TUTORE	DATE
	FEC_INI_TUTOR1	DATE
	FEC_FIN_TUTOR1	DATE
	FEC_INI_TUTOR2	DATE
	FEC_FIN_TUTOR2	DATE
PO_ALUMNOS_ACTIVIDAD	MATRICULA	NOT NULL NUMBER(6)
	ID	NOT NULL NUMBER(6)
	CALIF_ACTIVIDAD	CHAR(2)
	FECHA_INSC	DATE
	CLAVE_INSC	NUMBER(6)
	SITUACION_INSC	NUMBER(1)

Nombre de la Tabla	Nombre del Campo	Tipo
PO_ALUMNOS_PROY	MATRICULA	NOT NULL NUMBER(6)
	CVE_PROYECTO	NOT NULL VARCHAR2(15)
	CVE_ACUERDO	NUMBER(4)
	TIPO_PROYECTO	NOT NULL NUMBER
	CALIF_PROYECTO	CHAR(2)
	VAL_ACUERDO_TAR	NOT NULL NUMBER(1)
	CVE_INSTRUCTOR	NUMBER(6)
	CVE_CICLO_ESCOLAR	NOT NULL CHAR(5)
	FECHA_INSC	DATE

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

	CLAVE_INSC	NUMBER(6)
	SITUACION_INSC	NUMBER(1)
PO_AREAS_TEMATICAS	CVE_AREA_TEM	NOT NULL CHAR(3)
	DESCRIPAREA	NOT NULL CHAR(40)
PO_AREA_SNI	CVE_AREA_SN	NOT NULL NUMBER(2)
	AREA_SNI	NOT NULL VARCHAR2(60)
PO ARTICULOS	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_IDIOMA	NOT NULL CHAR(2)
	CVE_CARRERA_ESP	NOT NULL CHAR(3)
	CVE_STATUS	NOT NULL CHAR(2)
	TITULO	NOT NULL VARCHAR2(500)
	AUTOR	NOT NULL VARCHAR2(500)
	NOM_PUB	NOT NULL VARCHAR2(140)
	VOLUMEN	NOT NULL NUMBER(2)
	NUMERO	NOT NULL VARCHAR2(15)
	PAGINAS	NOT NULL VARCHAR2(15)
	FECHA_PUB	NOT NULL DATE
	INST_IMP	NUMBER(1)
	INST_OTRA	NUMBER(1)
	ARBITRADO	NUMBER(1)
	VALIDADO	NUMBER(1)
NOMBRE_OTRA	VARCHAR2(100)	
ISSN	VARCHAR2(20)	
PO_BIBLIOGRAFIA	CVE_BIBLIOGRAFIA	NOT NULL NUMBER(4)
	AUTOR	NOT NULL VARCHAR2(100)
	NOMBREBIB	NOT NULL VARCHAR2(120)
	EDICION	NOT NULL NUMBER(2)
	EDITORIAL	NOT NULL VARCHAR2(200)
	ANIO	NOT NULL NUMBER(4)
	CVE_PAIS	NOT NULL CHAR(2)
	FECHA_REGISTRO	DATE
	EJEMPLARES	NUMBER(2)
PO_BITACORA	CLAVE	NOT NULL NUMBER(6)
	ENTRADA	NOT NULL VARCHAR2(20)
	SALIDA	VARCHAR2(20)
	ACCION	VARCHAR2(1000)

Nombre de la Tabla	Nombre del Campo	Tipo
PO_ASPIRANTE	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_PAIS_NACION	NOT NULL CHAR(2)
	CVE_PAIS_NACIM	NOT NULL CHAR(2)
	CVE_CIVIL	NOT NULL CHAR(1)

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

	CVE_PAIS_RES	NOT NULL CHAR(2)
	CVE_ENTIDAD	NOT NULL CHAR(2)
	CVE_MUNICIPIO	NOT NULL CHAR(3)
	CVE_STATUS_ASP	CHAR(2)
	NOMBRE	NOT NULL VARCHAR2(50)
	AP_PATerno	NOT NULL VARCHAR2(30)
	AP_MATERNO	VARCHAR2(30)
	FECHA_NAC	NOT NULL DATE
	DIRECCION	NOT NULL VARCHAR2(60)
	COLONIA	NOT NULL VARCHAR2(60)
	CP	NOT NULL CHAR(5)
	TELEFONO	VARCHAR2(20)
	RFC	VARCHAR2(14)
	HOMOCLAVE	VARCHAR2(4)
	MAIL	VARCHAR2(30)
	CURP	VARCHAR2(25)
	CVE_PROG	CHAR(2)
	CVE_AREA_TEM	CHAR(3)
	CVE_TUTOR	NOT NULL NUMBER(6)
	CVE_TUTOR2	NUMBER(6)
	CVE_TUTOR3	NUMBER(6)
	FECHA_EVAL	DATE
PO_BITACORA_PROY	CLAVE	NOT NULL NUMBER(6)
	ENTRADA	NOT NULL VARCHAR2(20)
	SALIDA	VARCHAR2(20)
	ACCION	VARCHAR2(1000)
PO_CALIFICACIONES	CVE_CALIFICACION	NOT NULL CHAR(2)
	DESCRIPCION	VARCHAR2(100)
	CALIF_NUM	NUMBER(2)
	PROMEDIABLE	NUMBER(1)
PO_CICLO_ESCOLAR	CVE_CICLO_ESCOLAR	NOT NULL CHAR(5)
	FECHA_INICIO	DATE
	FECHA_FIN	DATE
PO_CONOCIMIENTOS _CONSECVENTES	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CONOCIMIENTOS_CONS EC	VARCHAR2(1000)
PO_CONOCIMIENTOS _PREVIOS	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CONOCIMIENTOS_PREVI OS	VARCHAR2(1000)
PO_CREDITOS_PROG_TIPO ACT	CVE_PROG	NOT NULL CHAR(2)
	ORIGEN	NOT NULL CHAR(2)

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

	CLAVE	NOT NULL NUMBER(1)
	CREDITOS	NOT NULL NUMBER(3)
PO_DOM_IDIOMA	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_IDIOMA	NOT NULL CHAR(2)
	HABLA	NOT NULL NUMBER(3)
	TRADUCE	NOT NULL NUMBER(3)
	ESCRIBE	NOT NULL NUMBER(3)
	VALIDADO	NUMBER(1)

Nombre de la Tabla	Nombre del Campo	Tipo
PO_EXP_DOCENCIA	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_AREA	NOT NULL CHAR(2)
	CVE_ESCUELA_IES	NOT NULL VARCHAR2(5)
	CVE_NIVEL_ESCOLAR	NOT NULL CHAR(2)
	MPARTE	NUMBER(1)
	FECHA_INI	NOT NULL DATE
	FECHA_FIN	NOT NULL DATE
	NUMERO	NOT NULL NUMBER(2)
	VALIDADO	NUMBER(1)
	HRS_SEMANA	NUMBER(2)
PO_EXP_LABORAL	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_NIVEL_PUESTO	NOT NULL CHAR(2)
	INSTITUCION	NOT NULL VARCHAR2(100)
	ACTIVIDAD	NOT NULL VARCHAR2(100)
	FECHA_INI	NOT NULL DATE
	FECHA_FIN	NOT NULL DATE
	SECTOR	NUMBER(1)
	VALIDADO	NUMBER(1)
PO_FORM_ACADEMICA	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_NIVEL_ESCOLAR	NOT NULL CHAR(3)
	CVE_CARRERA_ESP	NOT NULL CHAR(3)
	CVE_ESCUELA_IES	NOT NULL VARCHAR2(5)
	CVE_ESCUELA_FAC	NOT NULL VARCHAR2(5)
	GRADUADO	NUMBER(1)
	FECHA_INI	NOT NULL DATE
	VALIDADO	NUMBER(1)
	FECHA_FIN	DATE
	FECHA_GRAD	DATE
	NOM_TESIS	VARCHAR2(500)
	DIR_TESIS	VARCHAR2(200)
	PORCENTAJE	NUMBER(3)
	TIPO_DOC	VARCHAR2(25)
PROMEDIO	NUMBER(5,2)	

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

PO_GRADO_INVESTIGA	CVE_GRADO	NOT NULL NUMBER(1)
	GRADO_INVESTIGA	NOT NULL VARCHAR2(40)
PO_MODALIDAD	CVE_MODALIDAD	NOT NULL NUMBER(3)
	DESCRIPCIONMC	NOT NULL VARCHAR2(500)
PO_NOM_SNI	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_AREA_SNI	NOT NULL NUMBER(2)
	NUM_REGISTRO	NOT NULL NUMBER(10)
	NIVEL	NUMBER(1)
	FECHA_INI	NOT NULL DATE
	FECHA_FI	NOT NULL DATE
	VALIDADO	NUMBER(1)

Nombre de la Tabla	Nombre del Campo	Tipo
PO_REQUISITOS	CVE_REQ	NOT NULL NUMBER(2)
	NOMBRE	NOT NULL VARCHAR2(100)
	COMENTARIO	VARCHAR2(120)
PO_REQ_ASPIRANTE	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_REQ	NOT NULL NUMBER(2)
	CUMPLE	NUMBER(1)
	OBSERVACIONES	VARCHAR2(100)
PO_SOFTWARE	CVE_SOFTWARE	NOT NULL NUMBER(3)
	NOMBRESOFT	NOT NULL VARCHAR2(500)
	VERSION	NOT NULL VARCHAR2(10)
	SIST_OPE	NOT NULL VARCHAR2(100)
	CIA_SOFT	NOT NULL VARCHAR2(100)
PO_STATUS_ASP	CVE_SOFTWARE	NOT NULL NUMBER(3)
	NOMBRESOFT	NOT NULL VARCHAR2(500)
	VERSION	NOT NULL VARCHAR2(10)
	SIST_OPE	NOT NULL VARCHAR2(100)
	CIA_SOFT	NOT NULL VARCHAR2(100)
PO_SUBTEMA	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_TEMA	NOT NULL NUMBER(2)
	CVE_SUBTEMA	NOT NULL NUMBER(2)
	DESCRIPCION_SUB	NOT NULL VARCHAR2(500)

PO_TEMA	CVE_ACT_ACADEMICA	NOT NULL CHAR(8)
	CVE_TEMA	NOT NULL NUMBER(2)
	DESCTEMA	NOT NULL VARCHAR2(500)

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

PO_TIPO_ACTIVIDAD	CVE_TIPO_ACT	NOT NULL NUMBER(1)
	TIPO_ACTIVIDAD	VARCHAR2(50)
PO_USUARIOPRIV	CLAVE	NOT NULL NUMBER(6)
	USUARIO	NOT NULL VARCHAR2(30)
	NIVELACCESO	NOT NULL NUMBER(1)
	ACCESOCALIF	NUMBER(1)
	CONECTADO	NUMBER(1)
PO_CERTIFICADO_ALUMNO	MATRICULA	NUMBER(6)
	ANIO	NOT NULL NUMBER(4)
	NUM_ACTA	NOT NULL NUMBER(3)
	NOMBRE	VARCHAR2(150)
	PROGRAMA	VARCHAR2(50)
	AREA_TEMATICA	VARCHAR2(100)
	CICLO_INGRESO	VARCHAR2(20)
	NOTA_AL_PIE	VARCHAR2(500)
	NOTA_TOEFL	VARCHAR2(500)
	NOTA_CREDITOS	VARCHAR2(500)
	FECHA_REGISTRO	DATE
	FIRMA1	VARCHAR2(100)
	CARGO1	VARCHAR2(100)
	FIRMA2	VARCHAR2(100)
CARGO2	VARCHAR2(100)	
PO_CERTIFICADO_CALIF	ANIO	NOT NULL NUMBER(4)
	NUM_ACTA	NOT NULL NUMBER(3)
	CONSEC_CALIF	NOT NULL NUMBER(3)
	CLAVE_ACT	VARCHAR2(10)
	NOMBRE_ACT	VARCHAR2(500)
	CALIFICACION	VARCHAR2(50)
	CREDITOS	NUMBER(3,1)
	PERIODO_ESCOLAR	VARCHAR2(20)
	FECHA_INICIA_PER	DATE
PO_CERTIFICADO_CREDITOS	ANIO	NOT NULL NUMBER(4)
	NUM_ACTA	NOT NULL NUMBER(3)
	CONSEC_CREDITOS	NOT NULL NUMBER(3)
	ORIGEN_CLAVE_TIPO_ACT	CHAR(3)
	ACTIVIDAD_ACAD	VARCHAR2(100)
	CREDITOS_REQUERIDOS	NUMBER(4,1)
	CREDITOS_OBTENIDOS	NUMBER(4,1)
	PORCENTAJE_ACREDITADO	NUMBER(4,1)

**CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL
DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.**

PO_ESTANCIA_POSDOC	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	INSTITUCION	NOT NULL VARCHAR2(100)
	LINEA	NOT NULL VARCHAR2(100)
	FECHA_INI	NOT NULL DATE
	FECHA_FIN	NOT NULL DATE
	ASESOR	NOT NULL VARCHAR2(100)
	VALIDADO	NUMBER(1)
PO_EXP_INV	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	LINEA	NOT NULL VARCHAR2(50)
	TEMA	NOT NULL VARCHAR2(50)
	INSTITUCION	NOT NULL VARCHAR2(100)
	GRADO_INV	NUMBER(1)
	FECHA_INI	NOT NULL DATE
	FECHA_FIN	NOT NULL DATE
	VALIDADO	NUMBER(1)
PO_PROGRAMAS	CVE_PROG	NOT NULL CHAR(2)
	NOMBRE	NOT NULL CHAR(30)
	CREDITOS	NUMBER(3)
PO_SEG_STATUS	CVE_ASPIRANTE	NOT NULL NUMBER(6)
	CVE_STATUS_ASP	CHAR(2)
	CVE_REALIZO_CAMBIO	NOT NULL NUMBER(6)
	FECHAREGISTRO	NOT NULL DATE
PO_PROYECTOS	CVE_PROYECTO	NOT NULL VARCHAR2(15)
	DIVISION	NOT NULL NUMBER(2)
	CVE_CENTRO_COSTO	CHAR(8)
	DESC_CENTROCOST	VARCHAR2(300)
	FECHA_INI	NOT NULL DATE
	FECHA_FIN	NOT NULL DATE
	RESPONSABLE	NOT NULL VARCHAR2(150)
	STATUS_PEP_RAIZ	NOT NULL VARCHAR2(100)
	PROYECTO	NOT NULL VARCHAR2(300)
	VAL_GAC	NOT NULL NUMBER(1)
	VAL_COORDINADOR	NOT NULL NUMBER(1)
	VAL_A_TEMATICA	NOT NULL NUMBER(1)
	CVE_EMPLEADO	NOT NULL NUMBER(6)
	ANIO	VARCHAR2(4)
	CVE_GAC	NUMBER(6)
	CVE_COORDINADOR	NUMBER(6)
	CVE_A_TEMATICA	NUMBER(6)
	UBICACION	VARCHAR2(60)
NUM_ALUM_ACEPTAR	NUMBER(2)	

CAPITULO IV. SISTEMA DE SEGUIMIENTO Y CONTROL DE LOS ESTUDIANTES DEL POSGRADO DEL IMP.

Para la Base de Datos de Posgrado no hubo necesidad de realizar ninguna instalación o configuración de Oracle 8i o del sistema operativo donde reside debido a que ya se encuentra en funcionamiento dando soporte a otros sistemas dentro del Instituto Mexicano del Petróleo y con un DBA a cargo que realiza todas las configuraciones y ajustes necesarios al DBMS, administración de los tablespaces, la configuración de la memoria usada por Oracle (SGA), la administración de índices, respaldos, etc., de acuerdo a las necesidades que requieran los diferentes proyectos que tiene el Instituto dentro de los cuales se encuentra nuestro sistema para el área de Posgrado. Solo se requirió de tener un usuario con posibilidad de escritura y lectura para poder crear las tablas obtenidas en nuestro diseño lógico.

La información sobre los catálogos a usarse en el sistema, como bibliografía, software, estatus, perfiles, actividades académicas, etc, se nos proporcionó en access y una vez que la Base de Datos fue creada solo fue necesario subir dicha información a Oracle a través de una conexión ODBC (Conectividad Abierta de Base de Datos) desde access.

Hasta éste punto ya se cuenta con una Base de Datos creada de acuerdo a las necesidades del área de Posgrado y con la información necesaria para poder iniciar con las pruebas que verifiquen que la administración de los aspirantes, alumnos, profesores, actividades académicas y proyectos que conforman el área se lleve acabo de manera correcta. En el siguiente tema se mencionan las pruebas realizadas a la Base de Datos creada y al DBMS que la contiene, así como de la aplicación que se desarrolló para la captura, actualización y consulta de la Base de Datos.

Capitulo V
**IMPLEMENTACIÓN Y MANTENIMIENTO DEL
SISTEMA.**

5.1 Diseño de Formas.

El diseño, la programación se desarrollaron mediante Oracle Developer, lenguaje SQL y lenguaje PL.

La figura 5.1 nos muestra la forma principal por medio de ella que se tiene acceso a todo el sistema, de acuerdo a los privilegios que tenga cada usuario, entre los componentes que la conforman tenemos: versión, código de colores, manual, status del usuario e identificación, que forman el sistema Posgrado-IMP, a continuación se describe cada uno de sus componentes.



Fig.5.1 Forma principal del sistema.



Botón Conexión

Permite establecer una sesión dentro del Sistema, el usuario registrado debe proporcionar su *nick* y *clave* para tener acceso a los módulos, si no es un usuario registrado necesitara ponerse en contacto con el ejecutivo correspondiente y posteriormente con el Administrador del Sistema.

Al dar clic sobre el botón este nos muestra la figura 5.2

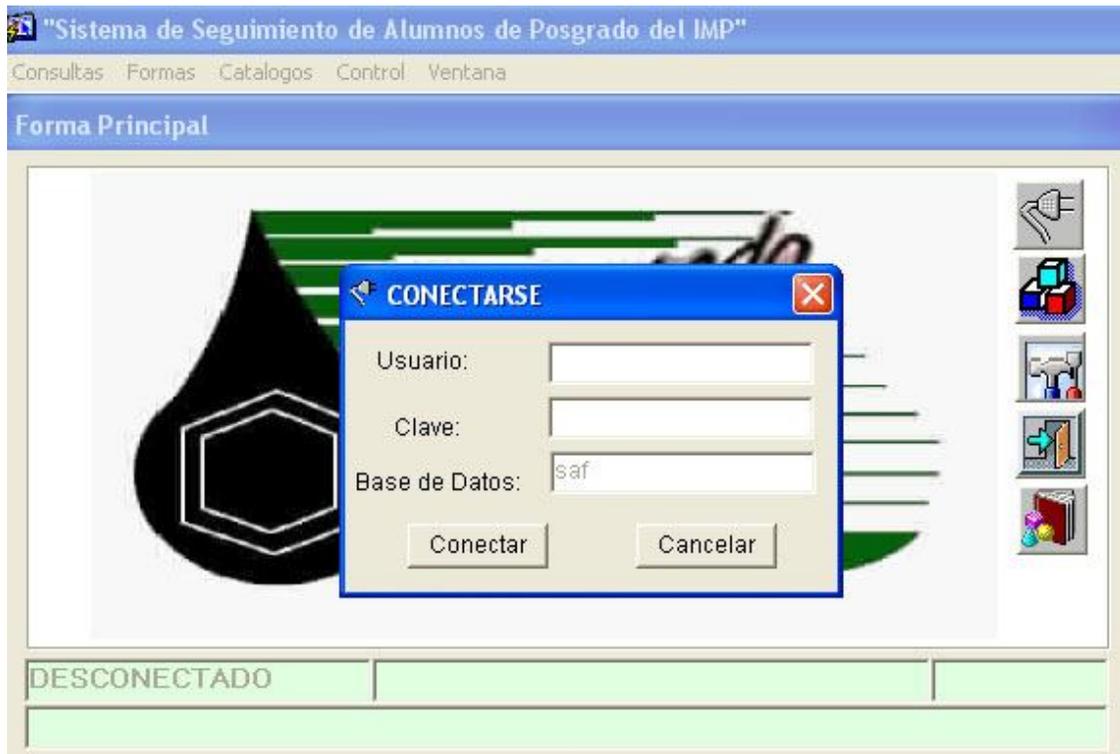


Fig. 5.2. Forma para validar al usuario.

En esta ventana se valida al usuario que entablara una conexión con la Base de Datos; en ella el sistema verifica que el usuario tenga permiso de acceso al sistema, de lo contrario no le permitirá el acceso. A continuación se describe cada uno de los campos y botones.

Usuario.- este campo hace referencia a su clave de empleado que el Instituto del Petróleo le asigno dentro de su padrón de empleados.

Clave.- este campo hace referencia a su password o numero confidencial que UD proporciono en el área de personal del Instituto Mexicano del Petróleo

Base de Datos.- este campo es llenado en automático por el sistema y no debe de introducir ningún dato, solo nos indica el nombre de la Base de Datos a la cual vamos a acceder.

Una vez que el usuario introduce sus datos para establecer la conexión debe de dar clic en el botón *Conectar*, de lo contrario en *Cancelar* para regresar a la forma principal.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Código PL/SQL usado en este botón.

```
declare
    v_fecha_salida char(20) := to_char(SYSDATE,'DD-MON-YYYY HH24:MI:SS');
begin
    IF( :ESTATUS = 'DESCONECTADO')
    THEN
        set_window_property('conectar_window', position, 130, 100);
        :global.return_item := :system.cursor_item;
        go_item('conectar.clave');

    ELSE
        :ESTATUS := 'DESCONECTADO';
        :NIVEL := "";
        :USUARIO := "";
        :CONECTAR.NOMBRE := '';
        UPDATE PO_USUARIOPRIV SET CONECTADO = 0 WHERE CLAVE =
:conectar.clave_empleado;
        COMMIT;
        if(:conectar.usr_accesocalif_si_no = 1) then
Set_Menu_Item_Property('FORMAS_MENU.F_CALIFICACIONES1',VISIBLE,PROPERTY_F
ALSE);
        end if;

        SET_ITEM_PROPERTY('CONTROL.CONECTAR',ICON_NAME, 'CONNECT' );
        SET_ITEM_PROPERTY('CONTROL.CONECTAR',TOOLTIP_TEXT, 'CONECTAR' );

Set_Menu_Item_Property('CONSULTAS_MENU.F_HISTORIALES_ACADEMICOS',ENABL
ED,PROPERTY_FALSE);
Set_Menu_Item_Property('CONSULTAS_MENU.F_ASPIRANTESREGISTRADOS',ENABLE
D,PROPERTY_FALSE);
Set_Menu_Item_Property('CONSULTAS_MENU.F_FORM_ACADEMICA',ENABLED,PROP
ERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.F_ALUMNOS',ENABLED,PROPERTY_FALSE)
;
Set_Menu_Item_Property('FORMAS_MENU.F_BIBLIOGRAFIA',ENABLED,PROPERTY_FA
LSE);
Set_Menu_Item_Property('FORMAS_MENU.F_CAPTURA_ACTIVIDADES',ENABLED,PRO
PERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.F_HORARIOS',ENABLED,PROPERTY_FALSE)
;
Set_Menu_Item_Property('FORMAS_MENU.F_INSCRIPCION_ACTIVIDADES',ENABLED,P
ROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.F_MODALIDAD',ENABLED,PROPERTY_FAL
SE);
Set_Menu_Item_Property('FORMAS_MENU.F_PERFIL',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.F_PROFESORES',ENABLED,PROPERTY_FAL
SE);
Set_Menu_Item_Property('FORMAS_MENU.F_PROF_EXT',ENABLED,PROPERTY_FALSE)
;
;
```

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

```
Set_Menu_Item_Property('FORMAS_MENU.F_SOFTWARE',ENABLED,PROPERTY_FALSE); --
Set_Menu_Item_Property('FORMAS_MENU.F_TUTORES',VISIBLE,PROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.FPROYECTOS1',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('CATALOGOS_MENU.F_CATALOGOS',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('CATALOGOS_MENU.F_REQ_ASPIRANTES',ENABLED,PROPERTY_FALSE);

Set_Menu_Item_Property('CATALOGOS_MENU.F_REQUISITOS',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('CONTROL_MENU.F_USUARIOSPRIV',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('CONTROL_MENU.F_BITACORA',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.F_MATRICULACION',ENABLED,PROPERTY_FALSE);
-- pl_id := Get_Parameter_List('tempdata');
--IF NOT Id_Null(pl_id) THEN
-- Destroy_Parameter_List(pl_id);
-- END IF;
update PO_bitacora
set salida = v_fecha_salida, ACCION = 'ENTRADA/SALIDA DEL SISTEMA DE POSGRADO'
where clave = :conectar.clave_employed AND ACCION = 'ENTRADA';
commit;
:CLAVE := ";
:conectar.PASS :=";
END IF;
end; -- and entrada = v_hora_entrada;
```



Código de colores

Muestra el código de colores que se utiliza en el sistema, figura 5.3, y el cual tiene un significado para los usuarios, a continuación se describe cada uno.



Fig. 5.3. Código de colores.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Color Ámbar (Inmodificable).- los datos contenidos en estos campos no podrán ser modificados por el usuario, solo permite introducir datos para su consulta.

Color Verde (Catalogo Asociado).- los datos contenidos en estos campos nos indican que están asociados a un catalogo del sistema, para ver el catalogo solo se da doble clic sobre el campo, en color verde, estos campos pueden ser modificados únicamente por el administrador del sistema y con el consentimiento de la Coordinación Académica, por ningún motivo acepta un valor que no este dentro del catalogo que el sistema muestra.

Código PL/SQL usado para los catálogos asociados.

```
DECLARE
a_value_chosen BOOLEAN;
BEGIN
a_value_chosen := Show_Lov('LOV_CICLOESC',5,5);
END;
```

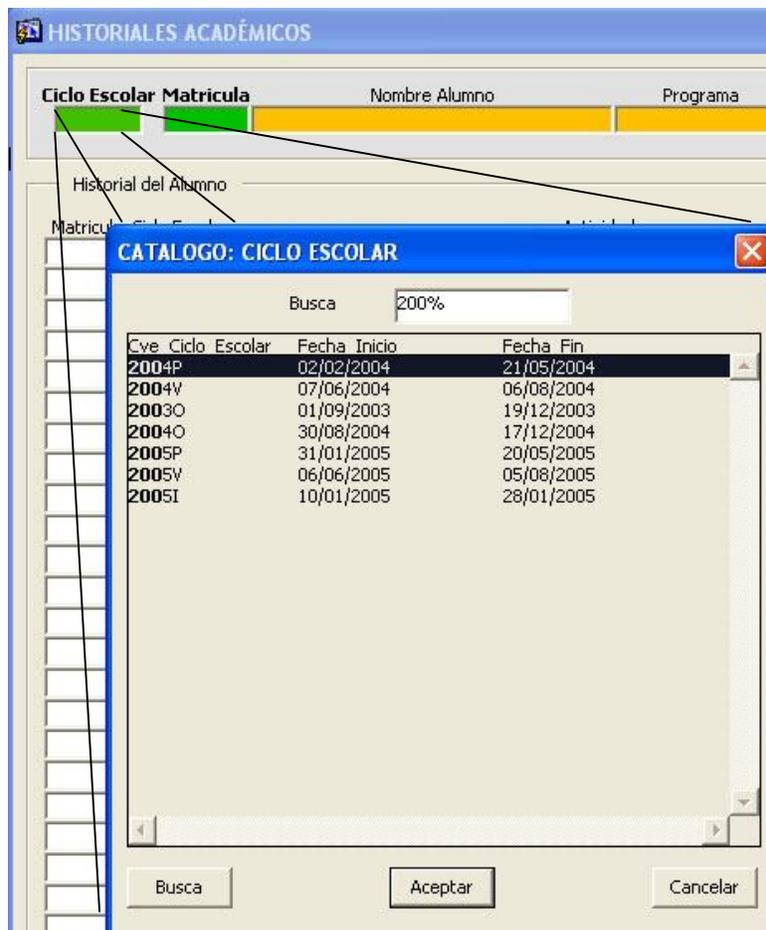


Fig. 5.4. Ejemplo de un catalogo asociado a un campo.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Color rosa (Irrepetible).- los datos contenidos en estos campos son únicos en todo el sistema y no son susceptibles de repetirse por parte del usuario.

Color Azul (Campo obligatorio).- los campos con este color deben de ser llenados en su totalidad para que el sistema le permita actualizar la información, de lo contrario el sistema no le permitirá hacer movimiento en la Base de Datos.

Código PL/SQL utilizado para el botón

```
begin
  set_window_property('colores_window', position, 100, 80);
  :global.return_item := :system.cursor_item;
  go_item('conectar.ok');
end;
```



Botón "Acerca de..."

Este botón muestra la versión del sistema que esta instalado en su equipo de cómputo.



Fig. 5.5. Versión del sistema.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA



Botón Manual de Usuario.

Nos muestra el manual de usuario del sistema de Posgrado y es visible para todos los usuarios en todo momento y se encuentra en la forma principal.



Botón Manual Instalación.

Muestra el manual de instalación del Sistema Posgrado y solo es visible para el administrador del sistema.



Botón Salir.

Nos permite cerrar la sesión entablada con la Base de Datos y cierra todas las ventanas del sistema. **IMPORTANTE** para poder salir del sistema debe desconectarse primero de la sesión con la Base de Datos y después de clic en el botón salir.



Fig. 5.6. Ventana principal.

Código PL /SQL utilizado para en el botón desconectar.

```
DECLARE
    V_BUTTON NUMBER;
BEGIN
    IF( :ESTATUS = 'CONECTADO') THEN
        go_item(':control.conectar');
        synchronize;
        --set_alert_property('desconectese',alert_message_text,'Favor de desconectarse antes de salir del
Sistema');
        v_button := show_alert('desconectar');

        /*      :ESTATUS := 'DESCONECTADO';
                :NIVEL := "";
                :USUARIO := "";
                SET_ITEM_PROPERTY('CONTROL.CONECTAR',ICON_NAME, 'CONNECT' );
                SET_ITEM_PROPERTY('CONTROL.CONECTAR',TOOLTIP_TEXT, 'CONECTAR' );
Set_Menu_Item_Property('REPORTES_MENU.ACTA',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('REPORTES_MENU.ORDEN_DEL_DIA',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('REPORTES_MENU.MINUTA',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('REPORTES_MENU.TARJETA',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('REPORTES_MENU.DATOS_GENERALES',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.BECARIOS_INTERNOS',ENABLED,PROPERTY_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.SEG_DE_SITUACIONES_INTERNAS',ENABLED,PROPERTY
_FALSE);
Set_Menu_Item_Property('FORMAS_MENU.SEG_DE_PAGOS_INTERNOS',ENABLED,PROPERTY_FALS
E);
Set_Menu_Item_Property('FORMAS_MENU.SEG_REP_TRIM_INTERNOS',ENABLED,PROPERTY_FALSE
);
Set_Menu_Item_Property('FORMAS_MENU.SEG_PROMEDIO_INTERNO',ENABLED,PROPERTY_FALSE)
;
Set_Menu_Item_Property('CONTROL_MENU.PROGRAMAS_DE_BECAS',ENABLED,PROPERTY_FALSE)
;
Set_Menu_Item_Property('CONTROL_MENU.USUARIOS_DEL_SISTEMA',ENABLED,PROPERTY_FALSE
);
Set_Menu_Item_Property('CONTROL_MENU.INTEGRANTES_DEL_COMITE',ENABLED,PROPERTY_FA
LSE);
        update bitacora
        set salida = :system.current_datetime
        where clave = :CONECTAR.clave
        and entrada = :dentrada;
        commit;
        EXIT_FORM;*/
    ELSE
        EXIT_FORM;
    END IF;
END;
```

5.2 Menús.

El sistema tiene cinco menús los cuales conforman todo el sistema de seguimiento de alumnos estos se activaran solo si es un usuario con permisos y dependiendo del *status* que tenga podrá consultar la información de cada uno de los submenús. A continuación se describe cada uno de ellos.

5.3 Consultas

En ella se encuentran las formas que dan acceso a las formas de Historiales Académicos, Aspirantes Registrados, Formación Académica. Son formas de carácter informativo que no pueden ser modificadas, el sistema proporciona la opción de poder descargar la información en un archivo Excel para su manipulación.



Fig. 5.7. Menú Consultas.

5.4 Formas.

En ella se encuentran las formas de Alumnos, calificaciones, captura actividades, horarios, inscripción actividades, matriculación, proyectos, profesores, profesores externos, bibliografía, modalidad, perfil y software.

Son formas de carácter informativo que pueden ser modificadas, por el usuario que tenga el *privilegio*, se puede borrar, insertar, se debe ser cuidadoso de los cambios que se hagan de la manipulación de la información, por que una vez hecha la transacción no se puede recuperar lo borrado.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA



Fig. 5.8 Menú de Formas.

5.5 Catálogos.

En ella se encuentran las formas de catálogos y requisitos del aspirante. Son formas de carácter informativo que no pueden ser modificadas, salvo la forma de requisitos del aspirante, por el usuario que tenga el *privilegio*.



Fig. 5.9 Menú Catálogos.

5.6 Control.

En ella se encuentran las formas de bitácora y usuarios del sistema. Son formas de uso exclusivo del administrador del sistema que pueden ser modificadas. Son visibles para los usuarios pero no tienen acceso a ellas por lo que se encuentran desactivadas por lo que el usuario solo las ve en color gris.



Fig. 5.10 Menú Control

5.7 Ventana.

En ella se puede modificar las posiciones de las ventanas del sistema de Posgrado, si se esta trabajando con varias ventanas. Fig. 5.11

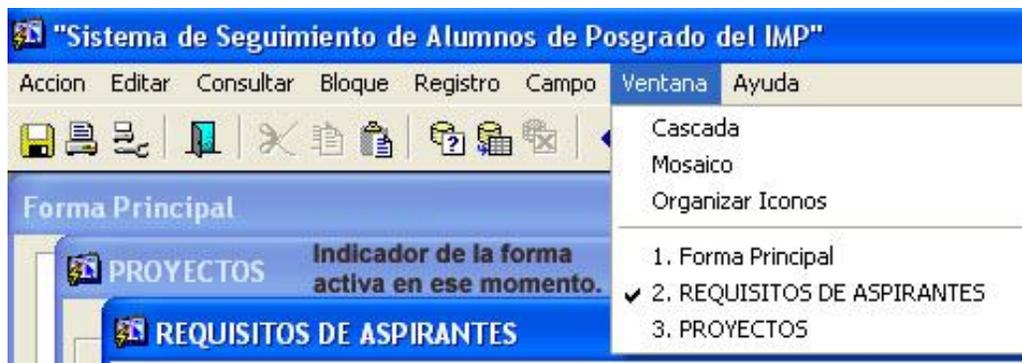


Fig. 5.11. Menú de ventana.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Cascada.- coloca las ventanas que se tengan abiertas en forma de cascada.

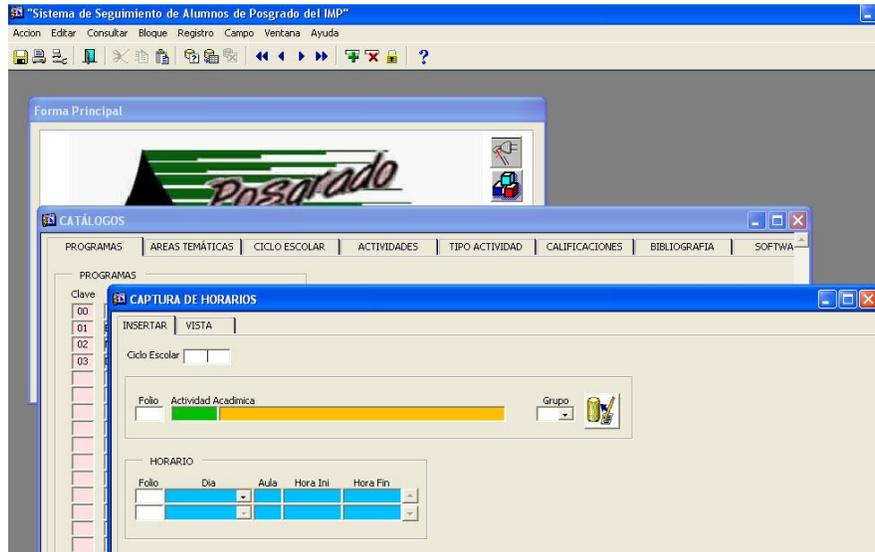


Fig. 5.12 Cascada.

Mosaico.- nos presenta todas las ventanas abiertas ocupando todo el espacio del escritorio de la PC.

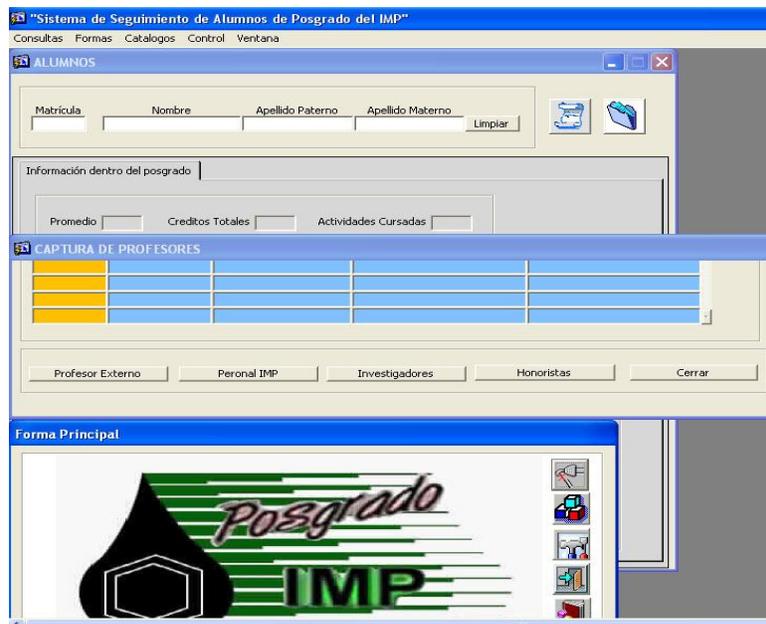


Fig. 5.13. Mosaico

El menú ventana nos muestra que forma es la que esta activa en ese momento y podemos navegar entre las formas, con solo dar clic en el nombre la forma para ponerla activa, siempre y cuando la forma este abierta como lo muestra la figura 5.11, la palomita indica cual es la forma que se esta utilizando en ese momento.

Código PL/ SQL utilizado en el menú.

```
PROCEDURE CALL_REPORT(rptname VARCHAR2,paramval VARCHAR2) IS

plid paramlist;
the_param varchar2(15) := 'tmpdata';

BEGIN

/* check if 'tmpdata' exists */
plid := get_parameter_list(the_param);

/* if it does exist, destroy it */
if not id_null(plid)
then
destroy_parameter_list(plid);
end if;

/* create it fresh */
plid := create_parameter_list(the_param);
/* conditionally set parameter list based on 'rptname' */
if rptname = 'INDSTRY'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'OLEX'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
add_parameter(plid, 'P_FILENAME', TEXT_PARAMETER, PARAMVAL);
elsif rptname = 'RFUND'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
add_parameter(plid, 'P_LASTNAME', TEXT_PARAMETER, PARAMVAL);
elsif rptname = 'MGRAPH'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
add_parameter(plid, 'P_LASTNAME', TEXT_PARAMETER, PARAMVAL);
elsif rptname = 'MAILMRG'
then
add_parameter(plid, 'P_LASTNAME', TEXT_PARAMETER, 'Abers');
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'YES');
elsif rptname = 'MAILASC'
then
add_parameter(plid, 'P_LASTNAME', TEXT_PARAMETER, 'Abers');
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'YES');
elsif rptname = 'INVDEMO'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'c:\oracle\investiga\alma\repopago'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'c:\oracle\investiga\alma\Remgasto'
then
add_parameter(plid, 'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'c:\oracle\investiga\alma\liquidoinvest'
```

```
then
  add_parameter(plid,'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'c:\oracle\investiga\alma\resrembo'

then
  add_parameter(plid,'PARAMFORM', TEXT_PARAMETER, 'NO');
elsif rptname = 'c:\oracle\investiga\alma\recibos_inv'
then
  add_parameter(plid,'PARAMFORM', TEXT_PARAMETER, 'NO');
else
  null;
end if;

/* call run_product to run report using Reports Server */
run_product(REPORTS,
            rptname,
            SYNCHRONOUS,
            RUNTIME,
            FILESYSTEM,
            plid);

--      ASYNCHRONOUS
/* Return to Intro screen after report is run */
-- execute_trigger('when_intro');
END;
```

5.8 Barra de Menú de Oracle Developer.

Esta barra es visible en todas las formas del sistema, excepto en la principal, el sistema la genera por defecto. A continuación se describe cada uno de sus componentes.



Fig. 5.14. Barra de Menús de Oracle Developer

-  Guardar permite guardar los cambios en la información que se hayan generado en la sesión del sistema.
-  Imprimir imprime el documento en la impresora predeterminada en su equipo.
-  Especificar Impresora nos permite ajustar los valores de la impresora, tipo de papel, orientación , buscar una impresora en red.
-  Salir nos permite cerrar la forma activa que estemos utilizando.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA



Cortar al igual que cualquier editor de texto nos permite cortar la información que seleccionemos. Copiar permite copiar la información seleccionada. Pegar nos permite pegar la información seleccionada o cortada.



Introducir consulta nos permite activar el campo para introducir el texto con el cual desea hacer la búsqueda en la Base de Datos. Se usa en algunas formas del sistema a menos que se indique lo contrario, por lo general se usa como campo de búsqueda por las claves, nombres o apellidos del alumno, materia, ciclo, programa, etc.



Ejecutar consulta este botón ejecuta la función del botón aceptar del Mouse. Puede dar clic sobre la tecla Intro para realizar la búsqueda o directamente sobre el botón.



Cancelar consulta permite detener la consulta.



Flechas de navegación permiten navegar entre los registros o bloques de la forma, si esta tuviera más de un bloque de datos.



Insertar registro permite ingresar nueva información a la Base de Datos.



Eliminar registro permite borrar los datos.



Bloquear registro al activarlo no deja introducir ni borrar datos del campo al que se le aplique.



Ayuda.- nos despliega la ayuda sobre Oracle Developer.

5.9 Pantallas de usuario.

Se define forma a las ventanas que el usuario ve y por medio de las cuales hace transferencia de información, actualización, borrado y consultas. A continuación se describe cada uno de los componentes, así como su uso.

Formas de Consulta

Estas formas como lo indican solo permiten hacer consultas de la información, nos permiten hacer búsquedas definidas e incluso permite bajar la información en un archivo Excel para su manipulación. Estas formas son:

- Historiales Académicos
- Aspirantes Registrados
- Formación Académica

Código PL / SQL usado para el botón.

```
DECLARE
v_matricula number(6) := :BLOCK_ELEMENTOS.TEXT_MATRICULA;
v_ciclo char(7) := ""||:BLOCK_ELEMENTOS.TEXT_CICLO_ESCOLAR||"";

begin
  if(:BLOCK_ELEMENTOS.TEXT_CICLO_ESCOLAR IS NULL ) then
    set_block_property('VPO_HISTORIAL_ACT_ACAD',DEFAULT_WHERE,'MATRICULA =
'||v_matricula);
    GO_BLOCK('VPO_HISTORIAL_ACT_ACAD');
    EXECUTE_QUERY;
  ELSE
    set_block_property('VPO_HISTORIAL_ACT_ACAD',DEFAULT_WHERE,'MATRICULA =
'||v_matricula||' and ciclo = '||v_ciclo);
    GO_BLOCK('VPO_HISTORIAL_ACT_ACAD');
    EXECUTE_QUERY;
  end if;
end;
-- if(:BLOCK_ELEMENTOS.TEXT_CICLO_ESCOLAR IS NOT NULL AND
:BLOCK_ELEMENTOS.TEXT_MATRICULA IS NOT NULL) then --
  set_block_property('PO_ALUMNOS_PROY',DEFAULT_WHERE,'MATRICULA = '||v_matricula||'and
cve_ciclo_escolar = '||v_ciclo)
```



Si hubiese un error en la búsqueda o desea realizar otra de clic en el boto para limpiar los campos, después introduzca la nueva consulta.

En el bloque de campos blancos que se encuentran bajo los datos personales se mostrara la información concerniente a ese alumno en particular como son su matrícula, el ciclo escolar , la actividad cursada o en curso, los créditos , y la calificación obtenida.

Código PL / SQL usado para el botón.

```
begin
go_block('VPO_HISTORIAL_ACT_ACAD');
CLEAR_BLOCK;
go_block('BLOCK_ELEMENTOS');
CLEAR_BLOCK;
end;
```

Aspirantes Registrados.

Muestra los aspirantes registrado en el sistema de Posgrado como nombre apellidos , RFC, CURP, homoclavé, dirección, programa solicitado, área temática, asesor , fecha de nacimiento, lugar de nacimiento, nacionalidad.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

La información puede ser filtrada por varios campos, todos aquellos que tengan un botón gris con una flecha, son los campos que nos permiten filtrar la información con información específica ya asignada por el sistema, esto se hace al dar clic sobre el botón lo que abre una lista de valores la cual nos despliega las opciones a escoger predefinidas, como lo muestra la Fig. 5.17.

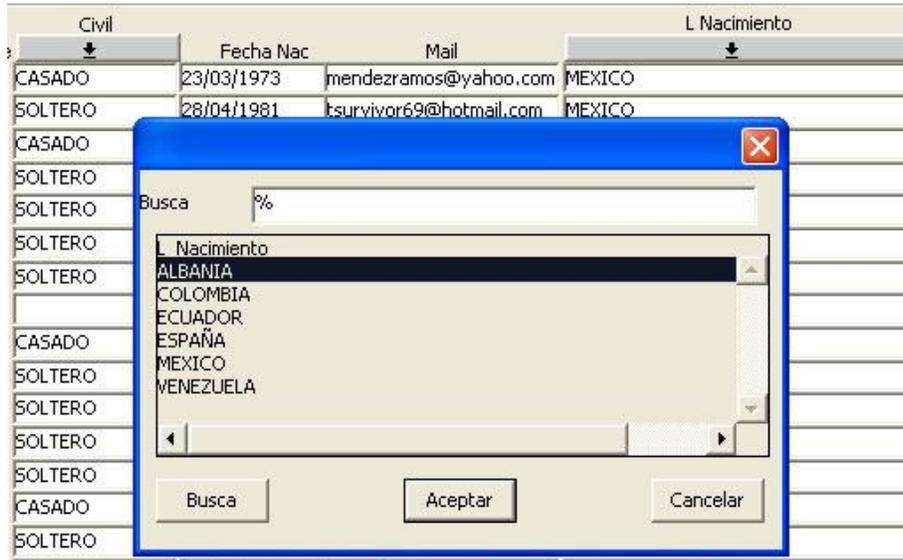


Fig. 5.17. Catalogo asociado al filtro LNacimiento.

Código PL / SQL utilizado para los filtros.

```
DECLARE
    v_detalle number(6);
    value_chosen BOOLEAN;
    v_civil char(40);
    V_BUTTON NUMBER;

BEGIN
    if(:guardar.TEXT_consulta IS NOT NULL) then
        value_chosen := Show_Lov('LOV_civil');
        v_detalle := :guardar.text_consulta;
        v_civil := ':lovs.text_item120';
        set_block_property('VDATOSASPIRANTES', default_where,'clave >=||v_detalle||' and civil=||v_civil);
        go_block('VDATOSASPIRANTES');
        execute_query;
        set_block_property('VDATOSASPIRANTES', default_where,'clave >= 1');

    else
        Set_Alert_Property('LOV_FALLO',alert_message_text,'NO HAY NINGUN VALOR EN EL
CAMPO "Consulta" DEBE INTRODUCIR LA CLAVE PARA PODER REALIZAR LA CONSULTA');
        V_button := Show_Alert('LOV_FALLO');
        go_item('guardar.TEXT_consulta');
        --go_record(to_number(:system.cursor_record));

    END IF;
```



Una vez que la información es obtenida y si el usuario así lo requiere puede guardarla, tal como es solicitada o por medio de los filtros necesarios, en un archivo Excel solo basta dar clic en el icono de Excel.

Código PL / SQL utilizado.

```
DECLARE
    al_noconsulta number;
begin
    IF (:VDATOSASPIRANTES.CLAVE IS NOT NULL) THEN
go_item('text_control.directorio');
    ELSE
        al_noconsulta := show_alert('ALERTA1');
    END IF;
END;
```

Abre una ventana en la cual da por default una ruta definida por el sistema en la cual se guardara el archivo.

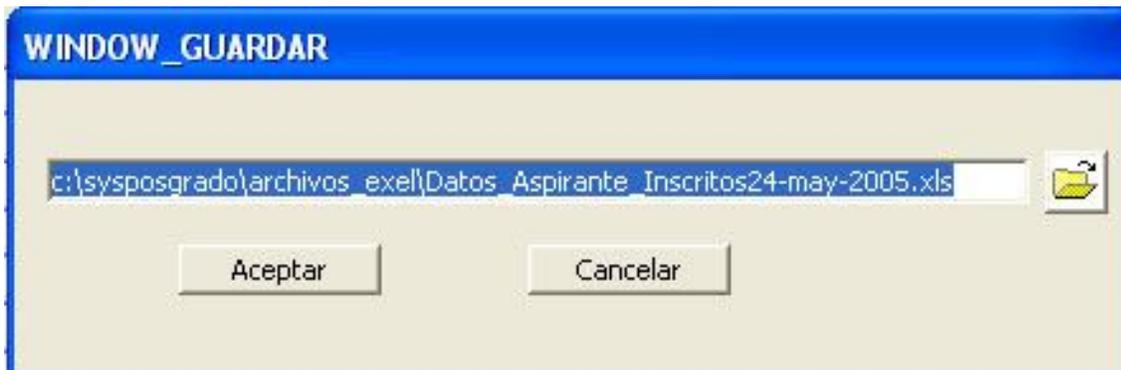


Fig. 5.18. Ruta de almacenamiento de archivo Excel.

Si desea otra ubicación para guardar el archivo debe de dar clic sobre el icono con lo que nos dará la opción de buscar un nuevo directorio. Si lo desea puede guardar con el nombre que el sistema da por default al archivo en la localidad que el usuario elija de su disco duro. Como lo muestra la Fig. 5.19

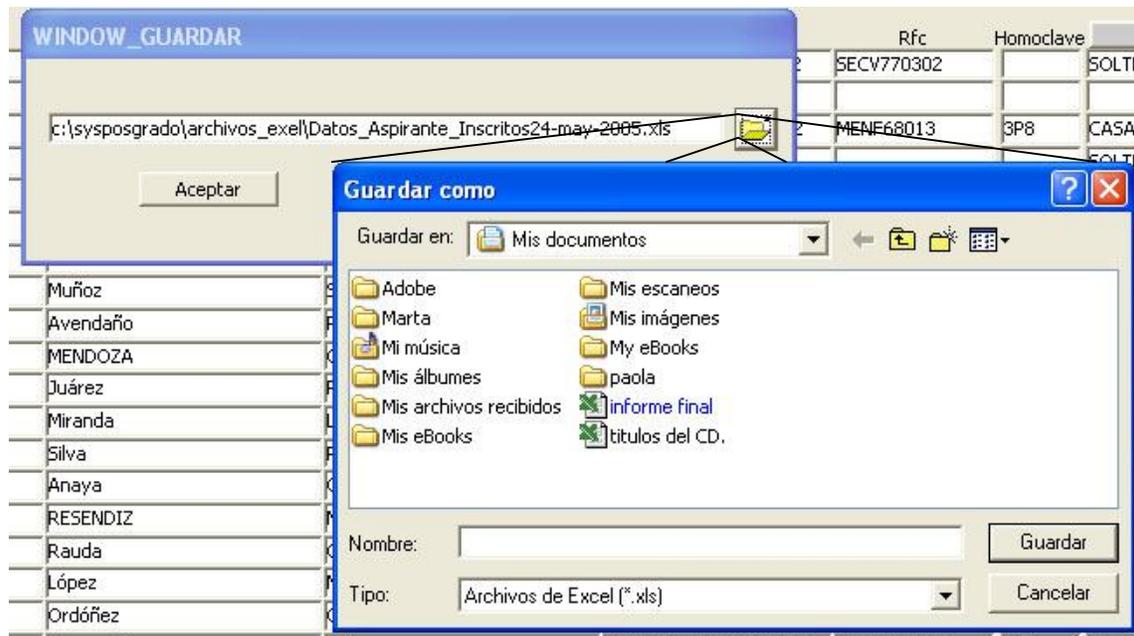


Fig. 5.19. Búsqueda de directorio.

Código PL / SQL utilizado

```
declare
  v_filename varchar2(255);
begin
  v_filename := get_file_name(NULL, NULL,
    'Archivos de Excel (*.xls)*.xls' ||
    'All Files (*.*)*.*)', NULL, SAVE_FILE );
  :text_control.directorio:= v_filename;
end;
```

Formación Académica.

Muestra la trayectoria académica del aspirante y posee las mismas funciones que la consulta anterior, permite guardar la información en archivos Excel y filtrarlos.

Forma Alumnos.

Esta forma presenta la información del alumno dentro del Posgrado del IMP. Formada por dos bloques en uno se muestra información del alumno como sus matriculas, nombres, apellidos, área temática, programa, nombre de los tutores, fecha de registro, calificación, etc. Es una forma básicamente de consulta.

Para efectuar una consulta general solo de clic sobre el botón “ejecutar consulta”, si lo que desea es hacer una consulta especifica de clic sobre el botón “introducir consulta” e introduzca la matricula o nombre del alumno después de clic en el botón “ejecutar consulta”.

Fig. 5.21. Forma de Alumnos.



Para ver el historial del alumno solo de clic sobre el botón, como lo muestra la siguiente figura.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Código PL/SQL usado.

```
DECLARE
alerta number;

BEGIN
If(:vpo_alumnos.matricula is not null) then
  SHOW_WINDOW('WIN_DATOS');
  SET_BLOCK_PROPERTY('PO_ASPIRANTE',DEFAULT_WHERE,'CVE_ASPIRANTE =
'||:VPO_ALUMNOS.CVE_ASPIRANTE);
  GO_BLOCK('PO_aspirante');
  EXECUTE_QUERY;
ELSE
  alerta := show_alert('alerta_datosper');
  if alerta = ALERT_BUTTON1 THEN
    GO_BLOCK('VPO_ALUMNOS');
  END IF;
end if;
END;
```

The screenshot shows a software window titled 'ALUMNOS'. At the top, there is a search bar with fields for 'Matricula' (203005), 'Nombre' (XÓCHITL), 'Apellido Paterno' (DOMÍNGUEZ), and 'Apellido Materno' (BENETTON), along with a 'Limpiar' button. Below this is a blue header for 'Datos Personales del Alumno de Posgrado'. The main area contains several sections of data:

- Estatus del Alumno:** VI VIGENTE
- Curp:** BOBX800723MDFMNC04
- Rfc:** DOBX800723
- Homoclave:** (empty)
- Fecha Nacimiento:** 23/07/1980
- Mail:** xmantis@prodigy.net.mx
- Estado Civil:** S SOLTERO
- Nacionalidad:** 00 MEXICANA
- Pais de Nacimiento:** 00 MEXICO
- Pais de Residencia:** 00 MEXICO
- Municipio:** 121 ZAPOTILTIC
- Entidad:** 15 EDO. DE MEXICO
- Telefono:** (empty)
- Direccion:** Plutón # 29
- Colonia:** Arcos de la Hacienda
- Cp:** 54730

At the bottom, there is a section for 'Datos seleccionados al momento del registro':

- Programa:** 03 Doctorado
- Area Temática:** FQH Físicoquímica de los hidrocarburos.
- Tutor:** 3361 JOSE MANUEL DOMINGUEZ ESQUIVEL
- Tutor2:** 13741 RODOLFO QUINTERO Y RAMIREZ
- Tutor3:** 0
- Fecha de Evaluación:** (empty)

Fig.5.23 Datos personales del alumno.

Forma Calificaciones

Forma que consiste en dos bloques de información los datos generales y el bloque de actividades y proyectos. Muestra las calificaciones obtenidas por el alumno en sus actividades en el ciclo que corresponde así como en los proyectos en los que ha participado. Es una forma de consulta e inalterable. De ser necesaria una rectificación deberá de ponerse en contacto con el ejecutivo correspondiente y obtenido el permiso solo el administrador del sistema podrá realizar el cambio.

Fig. 5.24. Forma de calificaciones.



Una vez seleccionado el ciclo escolar y la matricula los demás campos se llenan automáticamente para ver la información del alumno solo de clic en el botón ejecutar consulta que esta a la derecha.



Si hubiese un error en la búsqueda o desea realizar otra de clic en el boto para limpiar los campos, después introduzca la nueva consulta.



Solo es visible por el administrador del sistema, para alguna corrección de información.

Código PL/SQL usado.

```
begin
/* PERMITE ACTIVAR LA PROPIEDAD DE "ACTUALIZACION PERMITIDA DEL BLOQUE DE DATOS"
TANTO DE PROYECTOS COMO DE ACTIVIDADES. */

set_block_property('ACTIVIDADES',UPDATE_ALLOWED,PROPERTY_TRUE);
set_block_property('PO_ALUMNOS_PROY',UPDATE_ALLOWED,PROPERTY_TRUE);
end;
```

Forma Captura Actividades Académicas.

Esta es la forma mas compleja de todo el sistema, en ella se puede actualizar la mayor parte de los datos y hacer todas las relaciones entre todas la entidades de la Base de Datos. A continuación se describe cada bloque.

Fig. 5.25. Forma captura de actividades académicas.

El primer bloque muestra la actividad su nombre, clave, créditos, el tipo al que pertenece, el objetivo que persigue y la fecha en la que se captura. Muestra la información general, se puede modificar la información, actualizar y borrar.

Fig. 5.26. Bloque de encabezado de actividad académica.

El segundo bloque compuesto por cinco pestañas ó secciones permite la asociación de la actividad con la bibliografía, software, temario, profesores, conocimientos.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

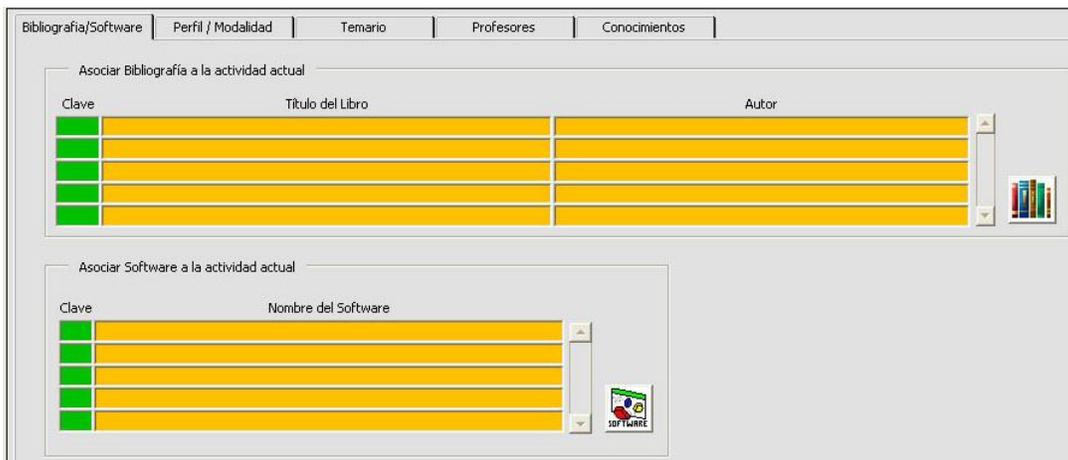


Fig. 5.27 Bloque de asociaciones de la actividad académica.

A continuación se describe cada una de las secciones.

Sección bibliografía/software.

Esta forma muestra la bibliografía y software asociado a la actividad académica que se consulte en ese momento. Se puede agregar nuevo software y nueva bibliografía por medio de los botones de cada uno de las secciones.



Agregar bibliografía.- este botón permite abrir una nueva forma en la cual se puede agregar nuevos libros asociados a la actividad académica. No necesariamente se debe de asociar el nuevo libro a la actividad, esto lo hace al desplegar el catalogo asociado y elegir el libro que se encuentre en esa relación.

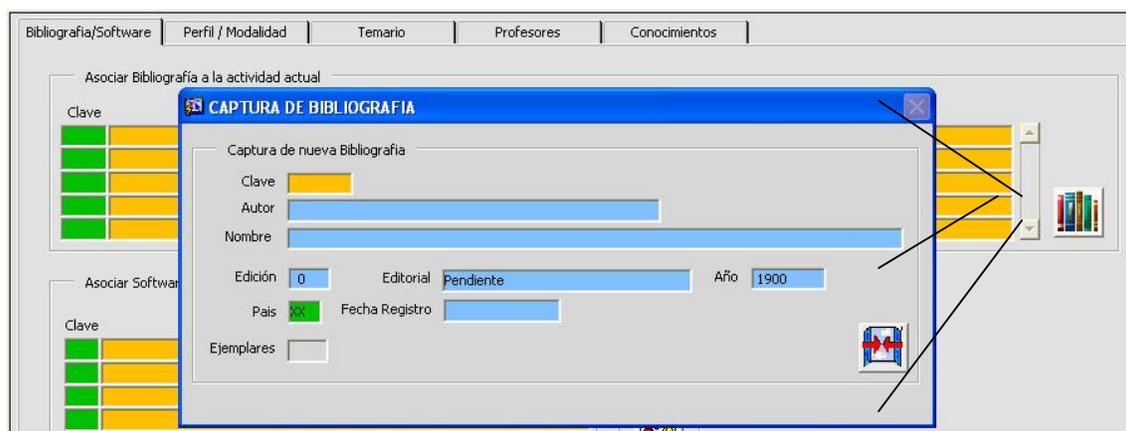


Fig. 5.28 Forma de alta de bibliografía.



Agregar software.- este botón permite abrir una nueva forma en la cual se puede dar de alta nuevos software asociado a la actividad académica.

Fig. 5.29. Forma de alta para nuevo software

Sección Perfil/Modalidad.

Esta forma nos muestra los perfiles en los que se encuentra la actividad académica, así como las diferentes modalidades a las que esta sometida, permite dar de alta nuevas modalidades y perfiles.

Fig. 5.30 Forma de Perfil/Modalidad.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA



Para dar de alta, modificar o eliminar el perfil. Muestra la forma de captura de perfiles.

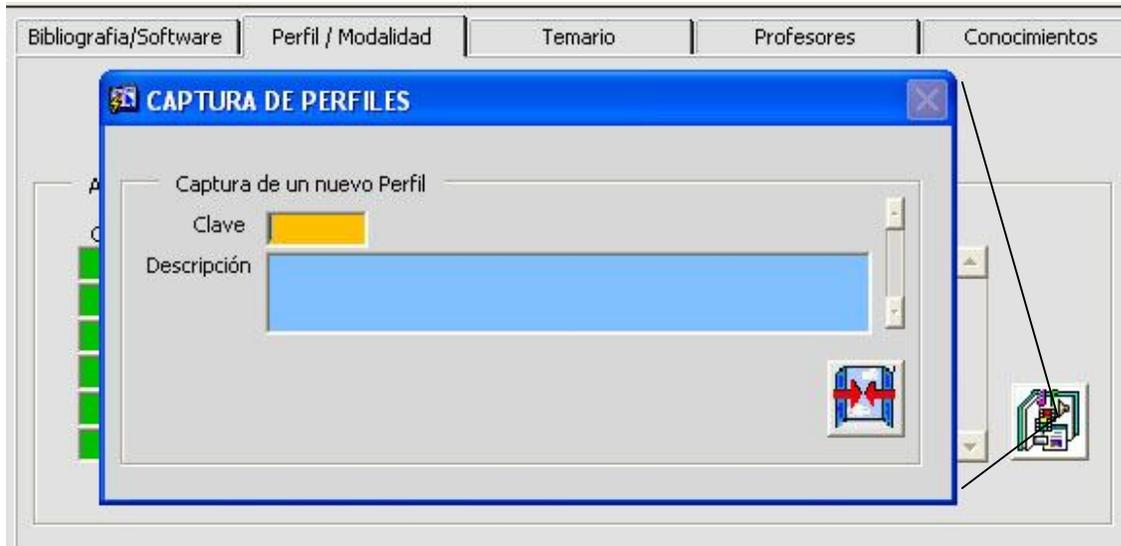


Fig. 5.31. Captura de perfiles.



Para dar de alta, modificar o eliminar la modalidad. Muestra la forma de captura de perfiles.

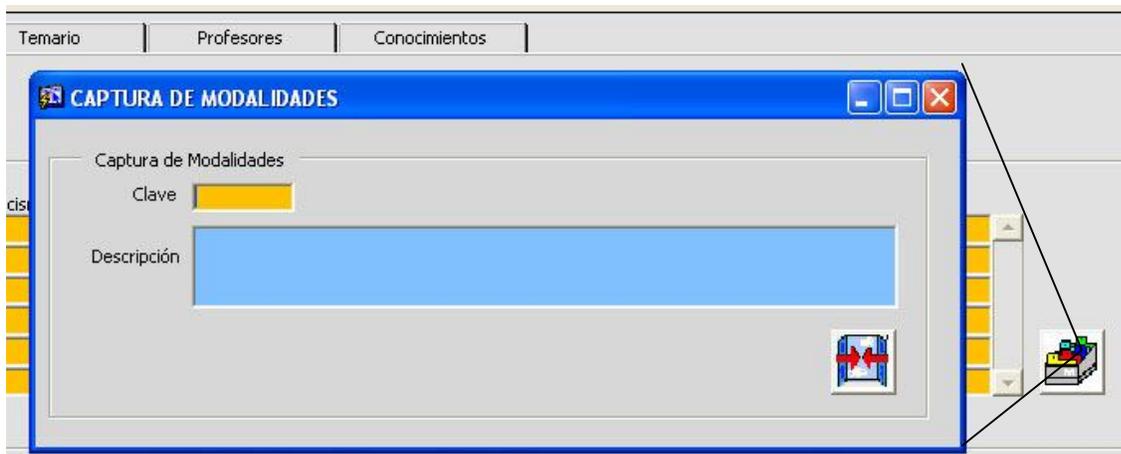


Fig. 5.32. Captura de Modalidad.



Permite abrir la forma de captura de profesores.

Clave	Descripción	Nombre	Apellido Paterno	Apellido Materno

Profesor Externo

Peronal IMP Investigadores Honoristas Cerrar

Fig. 5.35 .Forma de captura de profesores.

- *Externo*. Profesores de alguna institución superior que presta sus servicios al Posgrado IMP.
- *Personal IMP*. Personal que por su alta experiencia y conocimientos imparte sus conocimientos dentro del Posgrado IMP.
- *Investigadores*. Personal que pertenece al SNI y se encuentra inscrito dentro del Programa de Especialistas del IMP.
- *Honorista*. Personal que se encuentra contrato bajo esta partida y que por su alto conocimiento, trayectoria académica y profesional cumple con el perfil.

Al dar clic en alguno de los botones descritos el sistema muestra un catalogo con la información de la persona que impartirá cátedra en el posgrado IMP. Solo selecciónelo y de clic en aceptar para agregarlo al catalogo de profesores disponibles.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

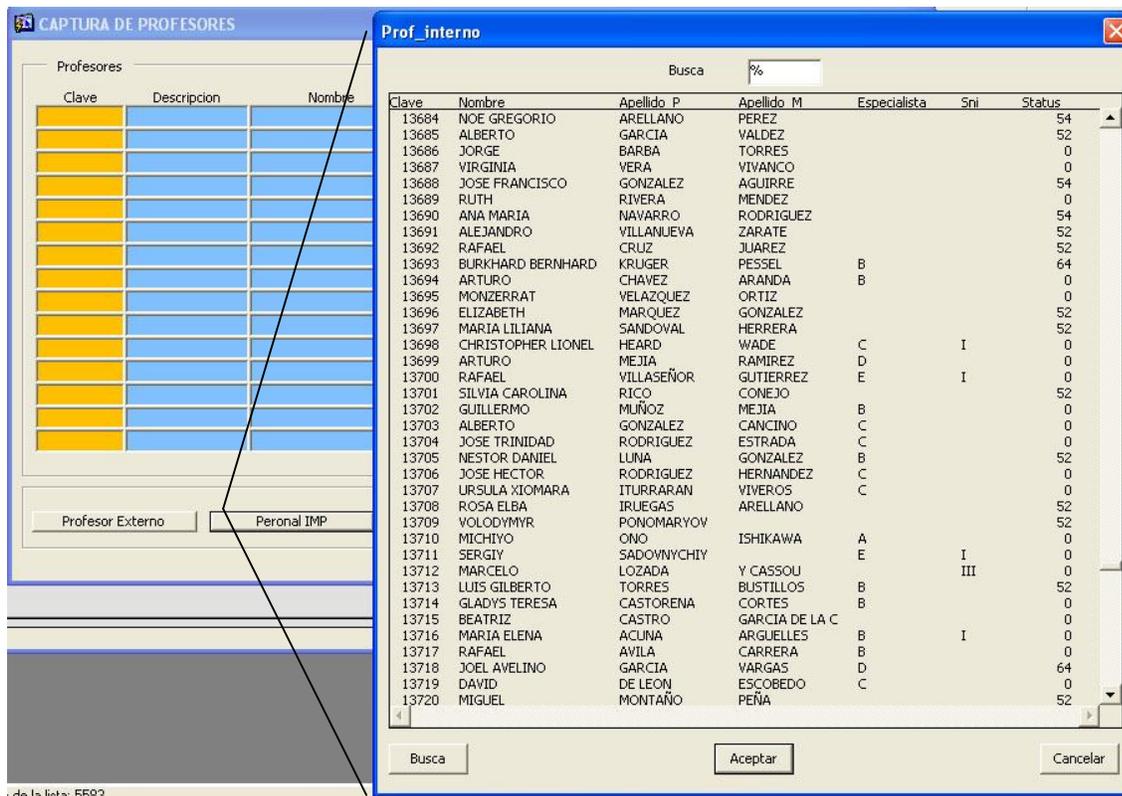


Fig.5.36.Catálogo de profesores del personal IMP.

Sección Conocimientos.

Muestra los antecedentes académicos que se deben de poseer para poder cursar la actividad, así como los adquiridos al final del curso. La información es susceptible de ser modificada.

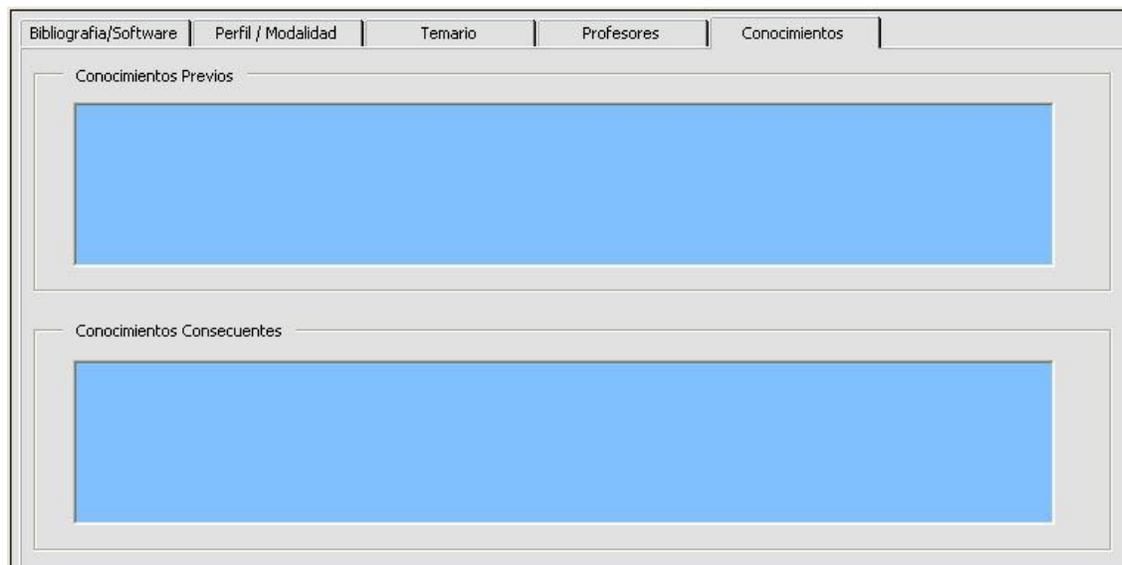


Fig. 5.37. Forma de captura de conocimientos.

Forma Horarios.

Permite capturar los horarios de las actividades académicas, el ciclo escolar, grupo, aquí se asignan a los profesores que estaban propuestos en el catalogo de Profesores, ya explicado.

Fig. 5.38. Forma de horarios.



Nos permite crear un nuevo registro.



Permite ver el catalogo de profesores propuestos.



Al activar este campo el profesor deja de ser propuesto y se convierte en definitivo para el ciclo escolar que se esta dando de alta o modificando.

Código PL/SQL usado en este botón.

```
DECLARE
    v_actividad          po_historico_act_profesor.cve_act_academica%type      :=
:PO_ACTIVIDAD_CICLO.CVE_ACT_ACADEMICA;
    v_clave              po_historico_act_profesor.clave%type                :=
:PO_ACTIVIDAD_PROFESOR.CLAVE;
    v_ciclo              po_historico_act_profesor.cve_ciclo_escolar%type    :=
:BLOCK_ELEMENTOS.TEXT_CICLOESC;
    v_fecha              date := SYSDATE;
    v_id                 po_historico_act_profesor.id%type := :PO_ACTIVIDAD_CICLO.ID;
    v_alerta             NUMBER;
    e_regexiste          EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_regexiste,-0001);

BEGIN

    IF(:PO_ACTIVIDAD_PROFESOR.PROF_ASIGNADO = 1) THEN
        INSERT INTO po_historico_act_profesor
VALUES(v_actividad,v_clave,v_ciclo,v_fecha,v_id);
        commit;
    elsif (:PO_ACTIVIDAD_PROFESOR.PROF_ASIGNADO = 0) THEN

        delete from po_historico_act_profesor where
        clave = v_clave
        and cve_ciclo_escolar = v_ciclo
        and cve_act_academica = v_actividad
        and id = v_id;

        COMMIT;
        If SQL%NOTFOUND THEN
            MESSAGE('NO EXISTE EL REGISTRO EN LA TABLA
PO_HISTORICO_ACT_PROFESOR');
        END IF;
    END IF;

    EXCEPTION
        WHEN e_regexiste THEN
            :PO_ACTIVIDAD_PROFESOR.PROF_ASIGNADO := 0;
            bell;
            v_alerta:=show_alert ('alert_notas2');

END;
```

Forma Inscripción Actividades.

Permite capturar los proyectos, las actividades académicas, el ciclo escolar y los datos personales del alumno. Es una forma de asociación de las actividades y proyectos todo se lleva a cabo mediante las listas de valores que se le presentan al usuario.

Id	Actividad Académica	Tipo actividad

Fig. 5.39. Forma Inscripción Actividades.



Borra los datos de los lienzos.



Ejecuta la búsqueda.



Guarda los cambios

Forma de Matriculación.

En ella se le asigna al aspirante su numero de matrícula que usara en su estancia en el Posgrado, esta forma muestra los datos que el aspirante capturo en el sistema de la Web como el programa que solicita, los tutores, el área temática y los datos personales de cada aspirante, como lo muestra las figuras siguientes.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

The screenshot shows a web application window titled "REGISTRO DE ALUMNOS (MATRICULACIÓN)". At the top, there are four input fields: "Clave", "Nombre", "Apellido Paterno", and "Apellido Materno". Below these is a tabbed interface with two tabs: "Asignar Matricula" (selected) and "Datos Personales". The "Asignar Matricula" section contains several rows of data, each with a green selection box and a yellow input field. The rows are labeled: "Matricula", "Fecha de Registro", "Programa", "Area Tematica", "Tutor deEnlace", "Tutor1", "Tutor2", "Director de Protocolo", "Director de Tesis", and "Codirector".

Fig. 5.40. Forma de matriculación.

The screenshot shows the same web application window, but with the "Datos Personales" tab selected. This section contains a grid of input fields for personal data. The fields are: "Status del Alumno", "Rfc", "Homoclave", "Curp", "Fecha de Nacimiento", "Estado Civil", "Pais de Nacimiento", "Nacionalidad", "Pais de Residencia", "Municipio", "Entidad", "Dirección", "Colonia", and "Cp". Below this grid is a section titled "Datos al momento del registro" which includes fields for "Programa", "Area Temática", "Tutor", "Tutor2", and "Tutor3". Each field has a green selection box and a yellow input field.

Fig. 5.41. Forma de matriculación sección datos personales del aspirante.

Forma de captura de profesores.

Esta forma permite seleccionar a los profesores sugeridos para impartir cátedra en el Posgrado esta conformada por cuatro categorías que son:

- Profesores Externos.- son aquellos profesionales que ofrecen sus servicios al IMP en particular al Posgrado y pertenece a alguna institución educativa pública o privada del país.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

- Personal IMP.- son aquellos profesionales que laboran en el Instituto y que por su trayectoria y experiencia laboral son idóneos para impartir cátedra dentro del Posgrado.
- Investigadores.- son aquellos profesionistas nacionales o extranjero que laboran en el Instituto o se encuentran dentro de alguno de los programas del Programa del Desarrollo del Factor Humano.
- Honoristas.- son los profesionales que prestan sus servicios en el Instituto por alguna de las partidas presupuétales que se manejan.

The screenshot shows a software window titled "CAPTURA DE PROFESORES". Inside, there is a table with the following columns: "Clave", "Descripcion", "Nombre", "Apellido Paterno", and "Apellido Materno". The table contains 12 empty rows. Below the table, there is a dropdown menu currently showing "Profesor Externo". To the right of the dropdown are four buttons: "Peronal IMP", "Investigadores", "Honoristas", and "Cerrar".

Fig. 5.42. Catalogo de profesores sugeridos.

El uso de la forma es bastante simple solo de clic en el botón de alguna de las categorías esto le mostrara una nueva ventana con la información de los profesionistas solo elija uno de ellos y de clic en aceptar con ello lo habrá agregado a su catalogo de profesores sugeridos.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

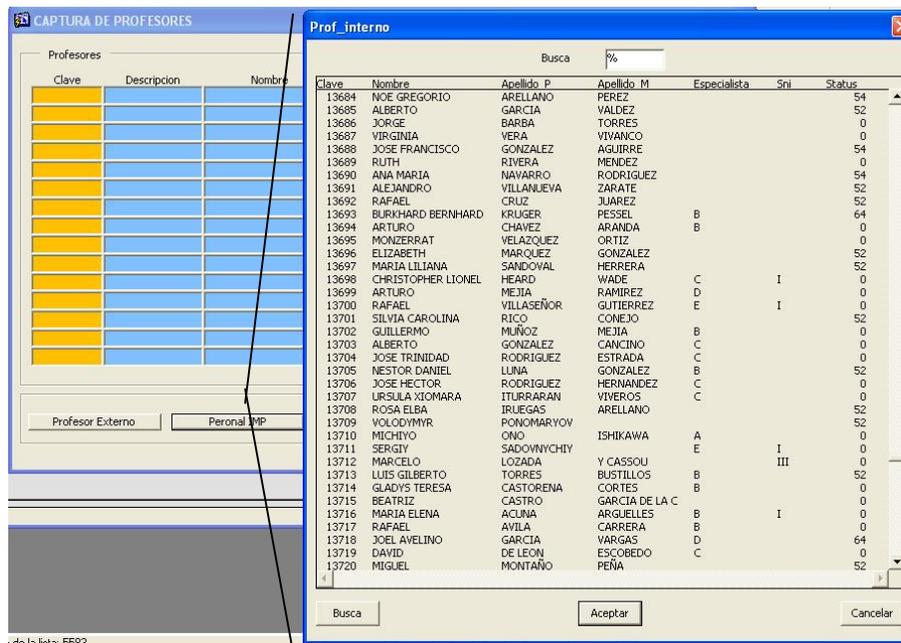


Fig. 5.43. Ejemplo de carga de profesores.

Forma captura de Profesores Externos

Esta forma sirve para tener almacenados los datos del personal externo que imparte clases en el Posgrado, se almacena la información personal, profesional de la persona.

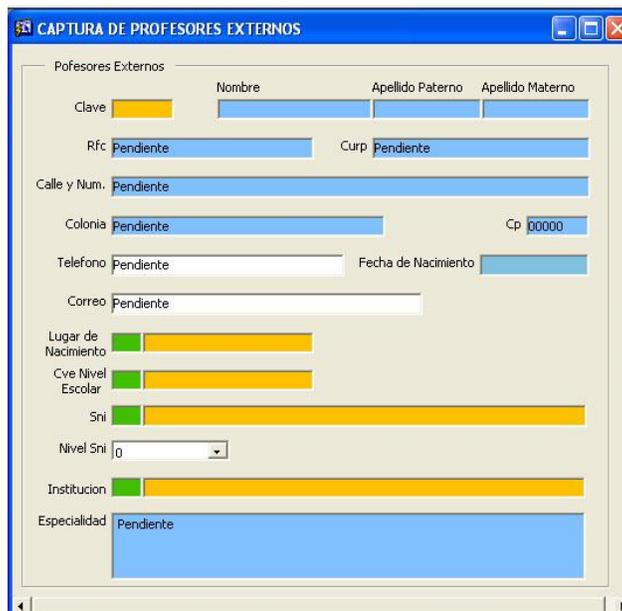
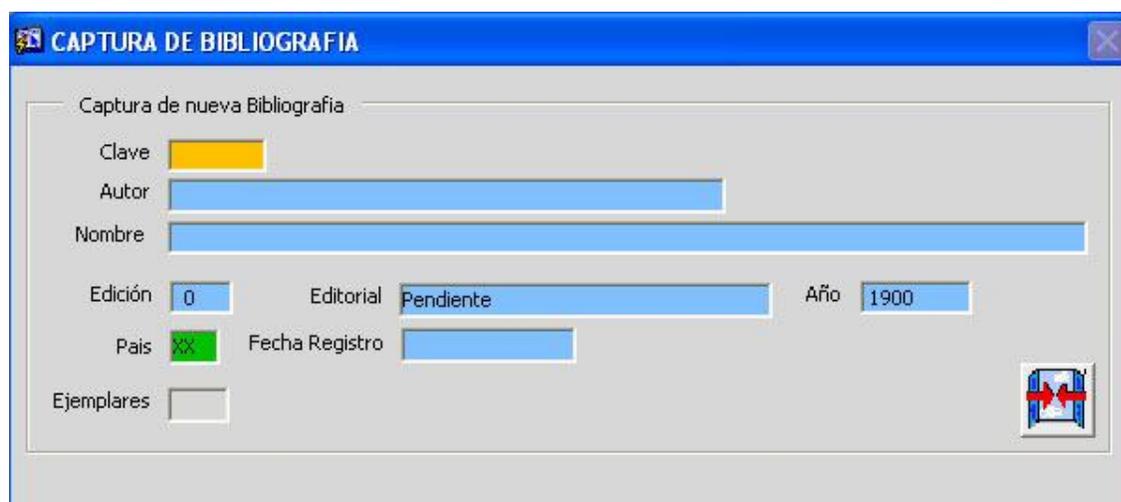


Fig. 5.44. Captura de información del Profesor Externo.

Forma de captura de bibliografía

Esta forma sirve para cargar información a los catálogos de bibliografía que se utilizan en el sistema, se puede modificar, borrar, actualizar. Todos los campos son obligatorios.



The screenshot shows a Windows-style window titled "CAPTURA DE BIBLIOGRAFIA". Inside the window, there is a section titled "Captura de nueva Bibliografía". Below this title, there are several input fields: "Clave" (a yellow box), "Autor" (a long blue box), "Nombre" (a very long blue box), "Edición" (a blue box containing "0"), "Editorial" (a blue box containing "Pendiente"), "Año" (a blue box containing "1900"), "Pais" (a green box containing "XX"), "Fecha Registro" (a blue box), and "Ejemplares" (a white box). In the bottom right corner of the form area, there is a small icon of a book with red arrows pointing outwards.

Fig. 5.45. Captura de Bibliografía.

Forma de captura de Modalidad

Esta forma permite cargar información en los catálogos de modalidad, al igual que la anterior se puede modificar, borrar, actualizar, todos los campos son obligatorios.

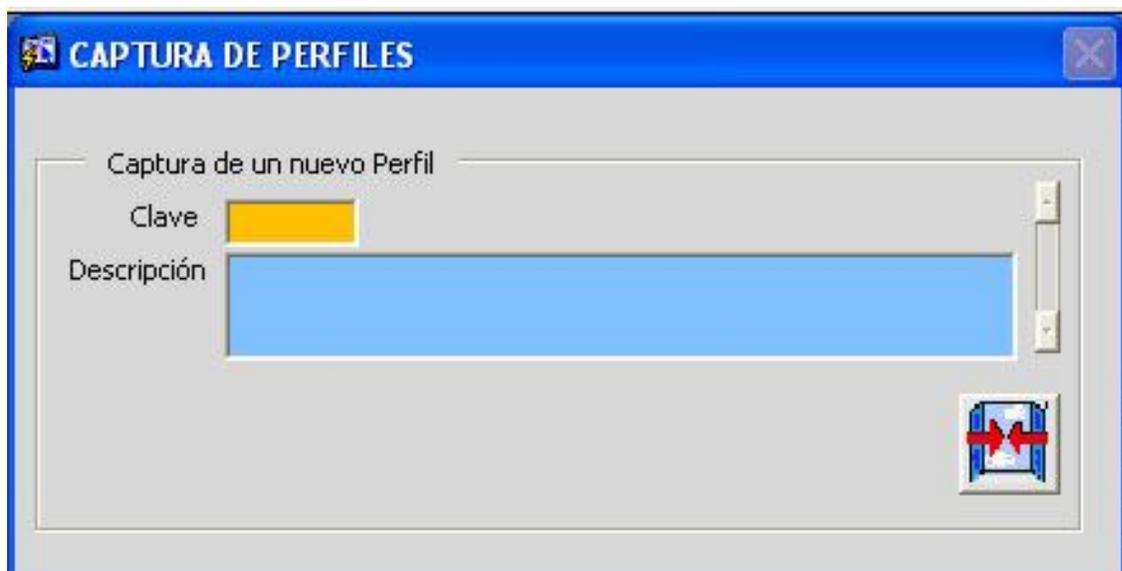


The screenshot shows a Windows-style window titled "CAPTURA DE MODALIDADES". Inside the window, there is a section titled "Captura de Modalidades". Below this title, there are two input fields: "Clave" (a yellow box) and "Descripción" (a long blue box). In the bottom right corner of the form area, there is a small icon of a book with red arrows pointing outwards.

Fig. 5.46. Captura de Modalidad.

Forma de captura de perfiles.

Esta forma permite capturar los diferentes perfiles de las áreas temáticas, se puede modificar la información, borrar, actualizar. Todos los campos son obligatorios.



The screenshot shows a software window titled "CAPTURA DE PERFILES". Inside the window, there is a section titled "Captura de un nuevo Perfil". Below this title, there are two input fields: "Clave" with a yellow background and "Descripción" with a blue background. To the right of the "Descripción" field is a vertical scrollbar. In the bottom right corner of the form area, there is a small icon of a window with red arrows pointing outwards, indicating a help or information function.

Fig. 5.47. Captura de Perfiles.

Forma captura de Software.

Esta forma permite capturar y alimentar el catalogo del software que se utiliza en el Posgrado, la información se puede modificar, borrar, actualizar. Los campos son obligatorios.



The screenshot shows a software window titled "CAPTURA DE SOFTWARE". Inside the window, there is a section titled "Captura de un nuevo Software". Below this title, there are four input fields: "Clave" with a yellow background, "Nombre del Software" with a blue background, "Sistema Operativo" with a blue background and the text "PENDIENTE", and "Versión" with a blue background and the text "0.0". Below the "Sistema Operativo" field, there is another blue input field with the text "PENDIENTE". In the bottom right corner of the form area, there is a small icon of a window with red arrows pointing outwards, indicating a help or information function.

Fig. 5.48. Captura de Software.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

5.10. PROGRAMA DE TRABAJO

Para la realización del sistema el Instituto Mexicano del Petróleo se nos dio un plazo máximo de un año, dentro del cual debemos de recabar la información, los requerimientos, el desarrollo y la puesta en funcionamiento del sistema. A continuación se presenta el programa de trabajo para la realización del proyecto.

Actividades	2004					2005							
	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago
Recopilación y Análisis de la Información													
Diseño de la Base de Datos													
Selección del SGBD													
Diseño de la Interfaz													
Prototipado													
Implementación													
Carga de datos													
Pruebas													
Liberación													

La recopilación de Información se planeo durante medio año, debido a que el Posgrado estaba en su primera etapa de desarrollo, por lo que hasta que estuvieran definidos los objetivos y valores, se recopilaría la información con la que iniciaríamos el análisis de la base de datos.

Diseño de la base de Datos se le dedico nueve meses en los cuales se hicieron pruebas, ajustes y mejoras a los requerimientos que durante el proceso se enriquecían en las reuniones de los ejecutivos del Área Académica.

La selección del sistema manejador de base de datos (DBMS) fue asignado por la Competencia de Tecnologías de Información. Una vez definido el DBMS se acordó utilizar la herramienta Oracle Developer para la creación de la Interfaz.

El desarrollo de los prototipos fue a la par del desarrollo del la Interfaz, durante el cual se realizaban pruebas de funcionamiento y conectividad con la Base de Datos. La implementación se realizo primero con el personal del Área Académica para la realización de pruebas, a su vez se fue cargando la Base de Datos con la información real con la que se contaba hasta el momento, se llevaron acabo las pruebas finales y por fin en Agosto del 2005 se libero el sistema.

5.11 REQUISITOS DE HARDWARE.

Todo sistema requiere de ciertas características de hardware para poder ejecutarse de una manera más eficiente, si el servidor o PC de escritorio donde se va a montar la aplicación de no cumplir con los requerimientos mínimos, aún y cuando el sistema creado sea programado de una manera excelsa, no podrá funcionar de manera correcta provocando insatisfacción por parte del usuario y lo primero que diría es: “el sistema no funciona”.

En nuestro caso los puntos más importantes a tomar en cuenta para la instalación del sistema para el área de Postgrado son: la memoria RAM, la capacidad del disco duro, y el procesador,

La PC donde se ha de instalar el sistema requiere de al menos 128MB de RAM para poder ejecutar las pantallas de Developer, pero se recomiendan 256 MB para un funcionamiento óptimo. La instalación del entorno de ejecución de Oracle Developer requiere de 16 MB de espacio en disco y 7MB para el sistema de Postgrado. En cuanto al procesador deberá de contar mínimo con un Pentium 2 de 333 Mhz, aunque al igual que la memoria se sugiere tener un Pentium 3 o similar para un mejor desempeño.

Esos son los requerimientos de hardware necesarios para que el sistema pueda operar sin ninguna dificultad, aunque como todo sistema, mientras mejor hardware, tenga su funcionamiento será mejor, y hoy en día se puede contar con hardware muy superior al que mencionamos, pero desde luego esto dependerá también al presupuesto que tenga el cliente tenga destinado para el desarrollo del sistema y los equipos donde será instalado.

5.12 REALIZACIÓN DE PRUEBAS.

La realización de pruebas, como cada una de las etapas de desarrollo de software, es de suma importancia ya que a través de ellas se pueden prevenir problemas al momento de poner en funcionamiento un sistema.

La etapa de pruebas debe ser realizada con el objetivo de encontrar cualquier fallo dentro del sistema, por lo cual no deben ser ejecutadas solo por los desarrolladores, debido a que tienen conocimiento total sobre el funcionamiento de cada módulo creado e inconcientemente se pueden realizar pruebas con ejemplos ideales en las que se sigue la secuencia normal del programa y se llega a suponer que el usuario hará lo mismo; en cambio un usuario llevará al sistema a situaciones que quizá no fueron contempladas por los desarrolladores, con ejemplos que se asemejen a las tareas operativas cotidianas de los usuarios, las cuales se quieren automatizar o hacer más eficientes a través del sistema construido.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Las pruebas realizadas al Sistema de Postgrado se enlistan a continuación.

- **Interfaz:** se revisaron las pantallas de manera individual (módulos) para determinar si su diseño era ordenado, entendible y amigable para el usuario, también para verificar que la información fluí de forma adecuada hacia y desde el módulo a prueba.
- **Validaciones:** Se pusieron a prueba las validaciones de las pantallas para la introducción correcta de la información así como la claridad de los mensajes o advertencias para los usuarios en caso de que violaran alguna de las validación o restricciones del módulo.
- **Información:** Se verificó que la información mostrada por cada módulo fuera la correcta y de utilidad para el usuario. También se verificó que la información capturada por lo usuarios fuera guardada de manera exitosa (en caso de que la pantalla fuera para captura de datos) y la recuperación correcta de la información de la Base de Datos (en caso de que fueran pantalla solo de consulta).
- **Integración:** Después de ejecutar las pruebas de manera individual, se realizaron ya con todos los módulos interactuando entre sí, verificando un correcto flujo de las pantallas, una navegación y acceso entre módulos sencilla y ordenada.

Una vez que se probaron los aspectos mencionados anteriormente y se corrigieron los errores encontrados y se adecuaron los módulos para una mejor interacción con el usuario, se procedió a realizar pruebas de desempeño de la herramienta, las cuales se muestran a continuación.

- **Base de Datos:** Se verificó el esquema de la Base de Datos, normalización; con el DBA se revisaron los Tablespace, tuvieran un tamaño adecuado, índices y recursos de servidor suficientes para una operación adecuada la de Base de Datos.
- **Conexión a la Base de Datos:** Se probó que la conexión fuera de manera correcta, rápida y sin interrupciones, así como la respuesta ante consultas extensas de información fuera en tiempos idóneos.
- **Prueba de estrés:** Se realizaron pruebas de estrés a la base de datos creando querys de inserción, consulta. y actualización ejecutándose simultáneamente.
- **Concurrencia:** Se probó la concurrencia del sistema hacia la Base de Datos, revisando al mismo tiempo que la integridad de los datos se conservara.
- **Instalación:** Se realizaron instalaciones y desinstalaciones en varios equipos para comprobar el procedimiento de instalación para detectar posibles problemas antes de que se implementará ya de manera definitiva.
- **Permisos:** Se realizaron pruebas con los usuarios verificando los permisos de cada nivel de usuario.

Una vez terminado el periodo de pruebas y con el visto bueno de los usuarios se procedió a instalar el sistema en todos los equipos necesarios para su puesta en marcha de manera oficial.

5.13 GUÍA DE INSTALACIÓN DE ORACLE DEVELOPER (CLIENTE).

La información aquí presentada le ayudará a realizar una instalación de Oracle Developer Ver. 6.0 en los siguientes sistemas operativos: Windows 95, Windows 98 y Windows NT. El proceso de instalación es el mismo en los sistemas operativos mencionados.

Para generar un cliente de Oracle Developer realice lo siguiente:

- Introduzca el Disco de Instalación de Oracle Developer Ver. 6.0. La primer pantalla nos pedirá el idioma, elegir *Spanish* y enseguida aceptar (Fig. 5.1).

Fig. 5.1. Pantalla principal



- Deberá introducir el Nombre de la Empresa el cual será *IMP* , y elegir el Directorio donde alojará sus archivos Oracle, por default genera el directorio *C:\ORAWIN95*. Este es el que se usa generalmente en todos los clientes. Terminada esta etapa, dar un click en el botón aceptar (Fig. 5.2).

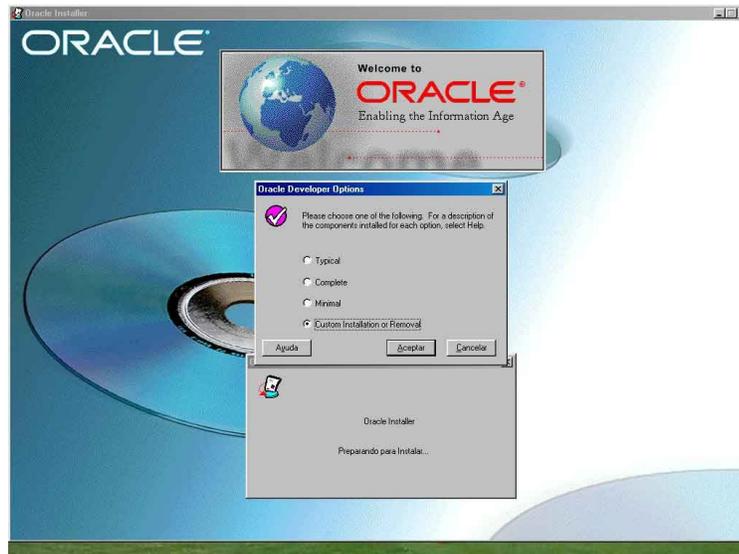
Fig. 5.2 Ruta de instalación.



CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

- Debemos elegir alguno de los tipos de instalación que se nos presenta. Como en este caso lo que se desea es la instalación de un cliente necesitaremos solo algunas aplicaciones, para elegir las aplicaciones seleccionaremos la opción *Custom Instalacion or Removal*. Dar un click en el botón aceptar (Fig. 5.3).

Fig.5.3 Tipo de instalación



- La pantalla nos muestra del lado izquierdo los *Productos Disponibles* en nuestro disco de instalación; si tuviéramos instalados algunos productos de Oracle aparecerían del lado derecho de la pantalla, de lo contrario aparecerá en blanco. De los Productos Disponibles seleccionaremos solo los *runtime* de Forms, Reports y Graphics: Forms Runtime, Graphics Runtime y Reports Runtime. Estos aparecerán en la parte de *Productos*

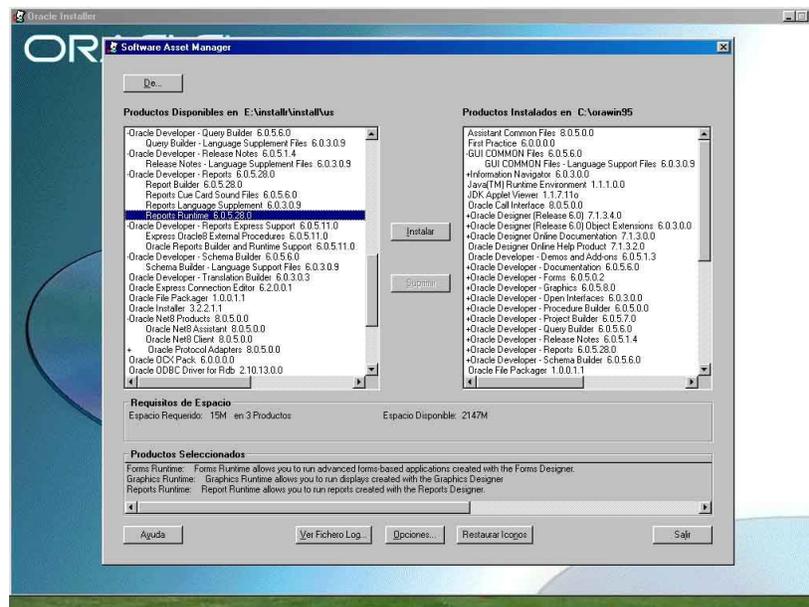


Fig. 5.4 Selección de componentes para la instalación.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

- **Seleccionados.** Además nos muestra el espacio requerido y el espacio disponible. Para estos 3 productos necesitaremos 15 Mb en DD. Para comenzar la instalación, una vez seleccionados los productos le damos un click en el botón instalar (Fig. 5.4). Comenzada la instalación nos aparecerán algunas ventanas preguntando sobre la ruta en que se van a instalar los productos de Forms, Reports y Graphics, solo hay que dar un click en el botón aceptar.
- Para realizar una instalación desde otra máquina le damos un click en el botón *De...* (superior izquierdo) cuando nos encontremos en la pantalla que se muestra en la Fig. 5.5. Aparecerá una ventana en la cual se deberá buscar el archivo win95.prd que se encuentra en el disco de instalación. La ruta es:

X:\Installr\install\us\win95.prd.

Donde X es la unidad donde se encuentra el disco de instalación.

Enseguida elegir los productos runtime antes mencionados.

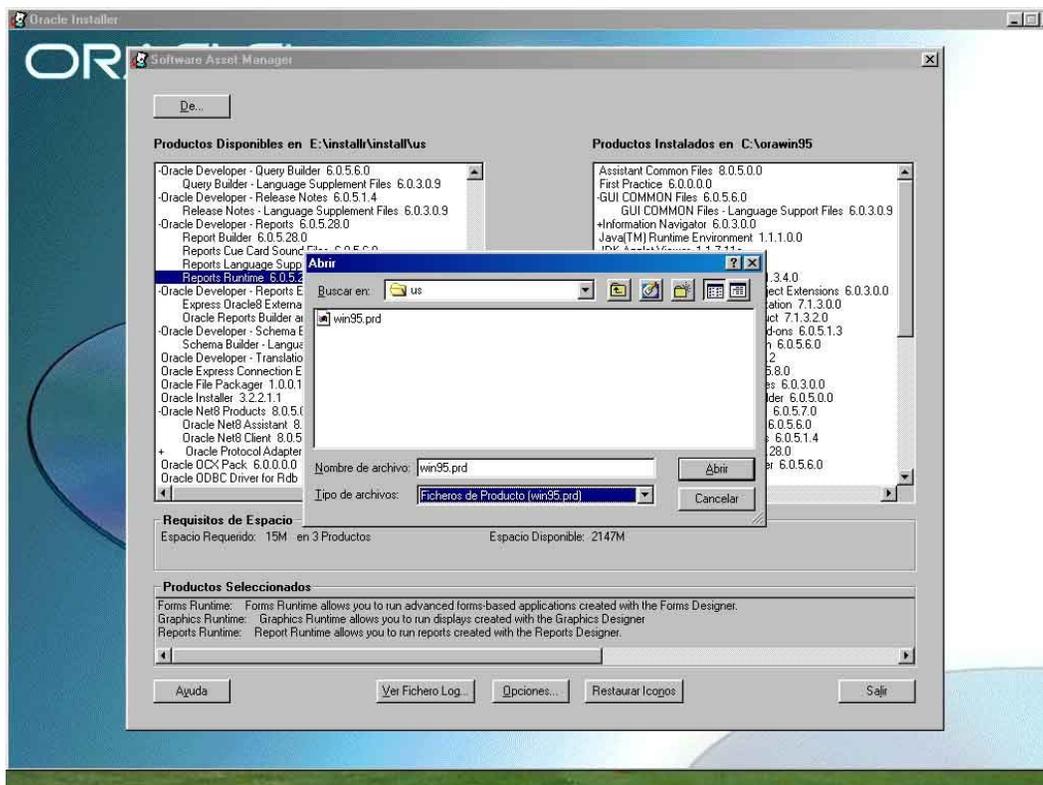


Fig. 5.5. Instalación desde otro equipo.

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

- Terminada la instalación se deberá reiniciar la máquina para que se realicen los cambios en el autoexec.bat.
- Una vez reiniciada la máquina se deberá configurar algunas variables de ambiente en el *regedit* (**inicio -> ejecutar -> regedit**). La ruta donde se encuentran las variables a modificar es HKEY_LOCAL_MACHINE \Software\Oracle, **si las variables de ambiente no se encuentran en el registro, generarla dando click derecho en la lista de variables de ambiente -> nuevo -> valor alfanumérico como una variable de valor de cadena**. Las variables son:
 - **FORMS60_PATH.-** Agregar al valor la cadena C:\SYSPOSGRADO\FORMAS. Si la variable ya contiene alguna cadena, agregar esta anteponiendo un punto y coma: ;C:\SYSPOSGRADO\FORMAS. Esta es la ruta donde el sistema buscará las formas de consulta y actualización del sistema de becas.
 - **GRAPHICS60_PATH.-** Agregar al valor la cadena C:\SYSPOSGRADO\GRAFICAS. Si la variable ya contiene alguna cadena, agregar esta anteponiendo un punto y coma: ;C:\SYSPOSGRADO\GRAFICAS. Esta es la ruta donde el sistema buscará las gráficas del sistema de becas.
 - **REPORTS60_PATH.-** Agregar al valor la cadena C:\SYSPOSGRADO\REPORTES. Si la variable ya contiene alguna cadena, agregar esta anteponiendo un punto y coma; C:\SYSPOSGRADO\REPORTES. Esta es la ruta donde el sistema buscará los reportes.
 - **UI_ICON.-** Agregar la cadena C:\SYSPOSGRADO\ICON. Aquí se encuentran los iconos utilizados por el sistema.
 - **DEVELOPER.-** Esta variable se tendrá primero que generar como una variable de valor de cadena y agregar C:\SYSPOSGRADO\MANUAL. Esta es la ruta donde se encuentra el manual en html.
 - **NLS_LANG.-** Cambiar la cadena actual por SPANISH_MEXICO.US7ASCII. Dentro del directorio Oracle puede haber varios NLS_LANG y en los cuales hay que realizar dicho cambio.
 - **NLS_DATE_FORMAT.-** En algunos casos ésta variable de ambiente no existe en el registro y se tiene que generar como una variable de valor de cadena. Cambiar el valor de la cadena actual por DD/MM/YYYY.

En el archivo llamado Tnsnames que se encuentra en la ruta:

Sistema operativo	Ruta
Windows 95	C:\ORAWIN95\NET80\ADMIN
Windows 98	C:\ORAWIN95\NET80\ADMIN
Windows NT	C:\ORANT\NET80\ADMIN
Windows 2000	C:\ORANT\NET80\ADMIN
Windows XP	C:\ORANT\NET80\ADMIN

Deberá agregarse las siguientes líneas:

```
GTIE.IMP.MX =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.100.181.4)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = gtie)
    )
  )

GTIE.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (COMMUNITY = tcp.world)(PROTOCOL = TCP)(Host = gtie.imp.mx)(Port = 1521))
      (ADDRESS = (COMMUNITY = tcp.world)(PROTOCOL = TCP)(Host = gtie.imp.mx)(Port = 1526))
    )
    (CONNECT_DATA =
      (SID = gtie)
    )
  )
```

Realizados las configuraciones en el regedit reiniciar de nuevo la máquina para que tome los cambios.

Posteriormente se creará una carpeta, en el directorio raíz, con el nombre de Develop, dentro de ella otra subcarpetas llamadas Formas, Reportes, Graficas, Icon y Manual, donde estarán alojados las formas, reportes, gráficas, iconos del sistema y los archivos html del manual, respectivamente. Por último se creará un acceso directo a la Forma F_principal, ubicada precisamente en la carpeta de Formas, llamado Sysposgrado.

Nota: Verificar que el archivo d2kwut32.dll se encuentre dentro de la carpeta de formas, de lo contrario pueden ocurrir algunos errores cuando se este haciendo uso del sistema.

5.14 .Mantenimiento del Sistema

EL mantenimiento es una etapa de las más importantes en el sistema se desarrollan mecanismos de evaluación, control e implementación de nuevas modificaciones, incluso nuevas formas; por lo que dependiendo de la magnitud de la modificación serán necesarios cambios en el análisis y el diseño.

Se describen las siguientes actividades realizadas durante el mantenimiento:

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Mantenimiento preventivo y correctivo, se detecto y corrigió errores que durante la fase de prueba se encontraron. Se modificaron algunas tablas en las que se necesitaron campos que no fueron contemplados en un principio, se generaron nuevas vistas y modificaron tablas, lo que implicó un rediseño del sistema. Continuamente se realizan una depuración en la base de datos, eliminando los registros que ya no son útiles para el usuario, además se hacen respaldos a la información, esto último lo realiza el Área de Tecnologías de la Información todos los fines de semana por medio de los responsables a cargo.

Mantenimiento adaptativo, consistió en adaptar el sistema a los diferentes sistemas operativos utilizados por el usuario final, siendo estos: Windows 95, Windows 98, Windows 2000, Windows XP. Así también se tuvo que adaptar a los distintos tipos de Pc's que se tenían a disposición, encontrando problemas principalmente en el espacio del disco duro y en la capacidad de la memoria RAM. Lo anterior va encaminado a mejorar la facilidad de mantenimiento y acrecentar la fiabilidad del sistema.

5.15. Pruebas de satisfacción del cliente.

Como parte de las políticas de calidad que el Instituto emplea en todos sus servicios se realizó una encuesta al Gerente de la Coordinación Académica del Posgrado , Lic. Gerardo de Alba Mora. , cuyos resultados se pueden ver en la Fig. 5.6.

Etapa de Ejecución del Proyecto		(-) Nivel de Satisfacción (+)										
		1	2	3	4	5	6	7	8	9	10	
1	El personal tuvo una actitud orientada a comprender mis necesidades											X
2	La planeación es satisfactoria y se actualiza con la dinámica del proyecto											X
3	El avance del proyecto es acorde con los compromisos contraídos											X
4	El personal tiene la capacidad necesaria para la realización del servicio											X
5	La actitud del personal es propositiva y contribuye en la solución de los problemas que se presentan											X
6	El personal utiliza procedimientos, equipos y herramientas adecuados											X
7	El personal IMP asignado al proyecto resuelve oportunamente los problemas que se presentan											X
8	Los entregables o informes del proyecto cumplen con los requisitos de calidad y alcance acordados											X
9	Se tomaron en cuenta mis opiniones, comentarios y sugerencias											X
10	El tiempo para la realización del servicio fue adecuado											X

Fig. 5.6 Encuesta de calidad

CAPITULO V. IMPLEMENTACIÓN Y MANTENIMIENTO DEL SISTEMA

Al término del trabajo el sistema de “Seguimiento y Control de Estudiantes del Posgrado del IMP”, quedó a cargo de la Lic. Lilia Santiago Monzón administradora del Sistema y soporte informático de la Coordinación Académica del Posgrado en coordinación directa con la M en C. Esperanza Sánchez Martínez designada por el Área de Tecnologías de Información como administradora de la Base de Datos.

El software debe desarrollarse en base a una metodología, de ésta manera se logrará que sea de una mayor calidad y cumpla con los requerimientos de los usuarios finales. Sin embargo, el tener una buena metodología, herramientas de desarrollo de primer nivel o hardware lo suficientemente amplio, no sirve de nada si como desarrolladores no mantenemos una amplia comunicación con los usuarios, en sus diferentes niveles; una falta de comunicación puede provocar el construir un sistema informático que no solucione los problemas que tenga el usuario o no se automatice las actividades o procesos dentro de la empresa o institución que permitan al cliente aumentar su productividad.

Con el desarrollo del “Sistema de seguimiento y control de los estudiantes del Posgrado del IMP” el área de la coordinación académica del Posgrado cuenta con una herramienta que le permite tener de manera rápida información, manejarla y administrarla de una manera más sencilla. Se han agilizado las labores administrativas, se han disminuido costos y tiempos, por lo que el sistema de Seguimiento y Control de Estudiantes del Posgrado del IMP cumple con su objetivo, al proveer de información de una manera rápida y concisa, su fácil manejo lo vuelven un sistema amigable para el usuario, se agiliza la captura al vincular la base de datos con la pagina de registro vía Web, por la cual el aspirante registra sus datos, lo que permite a la Coordinación Académica enfocarse solo en aquellos aspirantes que tienen alta probabilidad de ser aceptados, lo que reduce el tiempo de operación del personal del área, y planear mejor las entrevistas a las que son sometidos dichos aspirantes, reduciendo el tiempo y/o costo del personal que las realiza. Se han dejado las bases para que el sistema en un futuro sea capaz de generar los reportes que requieren los ejecutivos del Área de Investigación y Posgrado, paso siguiente en el desarrollo del sistema, pero el cual no forma parte de este trabajo de tesis.

ABAMSON, Ian; Michael Abbey, Michael J. Corey
ORACLE8i Guía de Aprendizaje
McGraw-Hill, España, 2000.
744 pp.

BROWN, Bradley D.
Oracle8i : desarrollo de soluciones Web
Madrid : McGraw-Hill Interamericana, 2001 Descr.
1015 p.

ENSOR, Dave, Ian Stevenson
Oracle Design
O'REILLY, E.U., 1997
526 pp.

GARDARIN, Georges, Mateo García F.
Bases de Datos
Paraninfo, Madrid , España, 1990
295 pp.

GONZÁLEZ PÉREZ, Alfonso
SQL Server Programación y Administración
Alfaomega, México, 1999
368 pp.

HIPSLEY, Paul
Developing Client / server Applications Whith Oracle Developer/2000
Sams publishing, Indianapolis, 1996
640 pp.

K. BURLESON, Donald.
Oracle Databases Applications
Coriolis, E.U. , 1996
461 pp.

KOCH, George
Oracle
McGraw-Hill, E.U., 1990
1045 pp.

KOLETZKE, Peter
Oracle Developer (Manual avanzado de forms y reports)
McGraw-Hill, España, 2000
776 pp.

LONEY, Kevin, Marlene Therianet.
Oracle 9i. Manual del Administrador
McGraw-Hill, Madrid, 2002
1024 pp

LOPEZ , Antonio

Metodologías de desarrollo : Producción automática de software con herramientas case

Mexico : Macrobit : Ra-Ma, c1991 Descr. Física
226 pp.

LUCAS Gomez, Ángel.

Diseño y Gestión de Sistemas de Bases de Datos

Paraninfo, Madrid , Espana, 1993
472 pp.

LUQUE Ruíz, Irene; Gomez-Nieto,Miguel; López, Enrique; Cerruela, Gonzalo

Bases de datos : desde Chen hasta Codd con Oracle

Alfaomega : ra-ma, México c2001
422 pp.

LUQUE Ruíz Irene

Diseño y uso de Bases Datos

RA-MA, Madrid, España, 1997
449 pp.

MCCLURE, Carma L.

Case la automatizacion del software

Madrid : Ra-Ma, 1992 Descr. Física 381 p.

PIATTIANI MARIO, Adoración de Miguel

Diseño de Bases de Datos Relacionales

Alfaomega Rama, España, 2000
549 pp.

PIATTIANI MARIO, Adoración de Miguel

Fundamentos y modelos de Bases de Datos

Alfaomega Rama, España, 1998
515 pp

ROB, Peter

Sistemas de bases de datos : diseño, implementación y administración

Traducción Rodolfo Navarro Salas
Mexico : International Thomson, c2004
838 pp.

S. PRESSMAN, Roger

Ingeniería del Software un enfoque práctico.

McGraw-Hill, España , 1993
824 pp.