

24021 3a 24j



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
"ACATLAN"

"PROYECTO DE MODERNIZACION DEL SISTEMA ESCOLAR DE LA DIVISION DE INGENIERIA MECANICA E INDUSTRIAL (DIMEI) DE LA FACULTAD DE INGENIERIA DE LA UNAM, HACIENDO USO DEL SISTEMA MANEJADOR DE BASE DE DATOS RELACIONAL (RDBMS) INFORMIX.

**T E S I S**  
QUE PARA OBTENER EL TITULO DE LICENCIADO EN MATEMATICAS APLICADAS Y COMPUTACION  
P R E S E N T A  
**JORGE CHAVEZ GARCIA**



ASESOR: ING. PERLA FERNANDEZ RUIZ



TESIS CON FALLA DE ORIGEN

277740



Universidad Nacional  
Autónoma de México



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

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

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

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

*A nuestra máxima casa de estudios,*

*la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO :*

Por abrigarnos dentro de su seno y encausarnos a nuestra superación profesional.

*A la Escuela Nacional de Estudios Profesionales Campus ACAPULCAN :*

Con respeto y admiración en agradecimiento de sus enseñanzas brindadas y sus constantes motivaciones.

*A la División de Ingeniería Mecánica e Industrial (DIMEI) :*

Por las facilidades brindadas para la realización de este trabajo.

*A la Ing. Perla Julieta Fernández Reyna :*

Por sus consejos, enseñanzas y estímulos que favorecieron notablemente la realización de este trabajo. ¡ Mil gracias !.

*A mis padres :*

Por todo el apoyo y motivación incondicional que de ellos he recibido con el cual he logrado alcanzar esta preciada meta.

“ Su mejor herencia ... GRACIAS”

*A mis hermanos :*

Por esa unión familiar que siempre nos ha caracterizado, brindándome sus consejos y celebrando juntos nuestros triunfos.

*A mis cuñados, tíos y primos :*

Por ser ellos un estímulo para alcanzar este objetivo y por transmitir sus experiencias y consejos invaluable en mi vida.

*A mis sobrinos :*

Deseando ser parte de la motivación para que alcancen todas sus metas día a día y consideren que todo es posible con un poco de esfuerzo.

A todos ellos, este triunfo también es suyo.

*A mi gran amor Carolina :*

Por todo el amor y comprensión que me has brindado en todos estos años, en los cuales hemos compartido triunfos y realizado nuestros sueños siendo este uno de los más importantes en nuestra vida profesional.

Gracias por la motivación que me inspiras para ser mejor en lo personal y en lo profesional, superarme día a día y saber que siempre estarás conmigo para disfrutar los frutos del éxito.

¡ Gracias por todo Amor !

*JORGE.*

## ÍNDICE.

<b>INTRODUCCIÓN.....</b>	<b>2</b>
<b>CAPITULO 1. CONCEPTOS BÁSICOS.....</b>	<b>5</b>
<b>1.1 ANTECEDENTES.....</b>	<b>5</b>
<b>1.2 BASE DE DATOS. ....</b>	<b>11</b>
1.2.1 SISTEMAS MANEJADORES DE BASES DE DATOS. ....	14
1.2.2 CARACTERÍSTICAS Y OBJETIVOS.....	17
<b>1.3 TIPOS DE RDBMS (Sistema Manejador de Base de Datos Relacional).....</b>	<b>22</b>
1.3.1 ORACLE, INFORMIX, INGRES.....	23
1.3.1.1 ORACLE.....	23
1.3.1.2 INFORMIX.....	28
1.3.1.3 INGRES.....	30
<b>CAPITULO 2. INFORMIX.....</b>	<b>38</b>
<b>2.1 DEFINICIÓN.....</b>	<b>38</b>
<b>2.2 CARACTERÍSTICAS.....</b>	<b>38</b>
<b>2.3 INFORMIX COMO RDBMS.....</b>	<b>42</b>
<b>2.4 PROS Y CONTRAS DE INFORMIX.....</b>	<b>46</b>
<b>2.5 ENTORNO DE PROGRAMACIÓN UNIX, DOS.....</b>	<b>48</b>
2.5.1 UNIX.....	48
2.5.2 MS-DOS.....	51
<b>CAPITULO 3. DESARROLLO DEL SISTEMA ESCOLAR DE LA DIMEI DE LA FACULTAD DE INGENIERIA DE LA UNAM.....</b>	<b>57</b>
<b>3.1 SITUACIÓN ACTUAL.....</b>	<b>57</b>
3.1.1 FACULTAD DE INGENIERÍA.....	57
3.1.2 INGENIERÍA INDUSTRIAL.....	61
3.1.3 MANEJO DE LA INFORMACIÓN EN LA DIMEI.....	64
<b>3.2 RECURSOS DISPONIBLES.....</b>	<b>65</b>

<b>3.3 ANÁLISIS.....</b>	<b>66</b>
<b>3.3.1 MODELO AMBIENTAL DE LA DIMEL.....</b>	<b>77</b>
<b>3.4 PROPUESTA DE SOLUCIÓN.....</b>	<b>114</b>
<b>3.5 DISEÑO DEL SISTEMA.....</b>	<b>115</b>
<b>3.5.1 ESTRUCTURAS DE PROGRAMA.....</b>	<b>116</b>
<b>3.5.2 DISEÑO GENERAL DEL SISTEMA.....</b>	<b>119</b>
<b>3.5.3 DISEÑO DE PANTALLAS.....</b>	<b>127</b>
<b>3.6 PRUEBAS E IMPLANTACIÓN.....</b>	<b>137</b>
<b>3.6.1 PRUEBA DE LA CAJA BLANCA.....</b>	<b>139</b>
<b>3.6.2 PRUEBA DE LA CAJA NEGRA.....</b>	<b>139</b>
<b>3.6.3 IMPLANTACIÓN.....</b>	<b>141</b>
 <b>CONCLUSIONES.....</b>	 <b>143</b>
 <b>BIBLIOGRAFÍA.....</b>	 <b>146</b>



## **INTRODUCCIÓN.**

Durante los últimos años, las computadoras y las redes informáticas han producido en nuestra sociedad un impacto de enormes consecuencias. La creciente integración de computadoras y comunicaciones dentro de un sistema único, ha llevado a una industria nueva y de rápido crecimiento : “La industria de comunicación y almacenamiento de datos basada en computadoras”. Con ello se dice que hemos entrado a un mundo automatizado, lo cierto es que estas herramientas han multiplicado la productividad y eficacia de las empresas.

Actualmente los trámites escolares de la DIMEI (División de Ingeniería Mecánica e Industrial) de la Facultad de Ingeniería son realizados en forma manual, lo que trae como consecuencia la pérdida de tiempo tanto para los usuarios como para los prestadores de servicio. Surge con esto la necesidad de integrar a la división a la modernidad de nuestro mundo, es decir, automatizar sus procesos manuales.

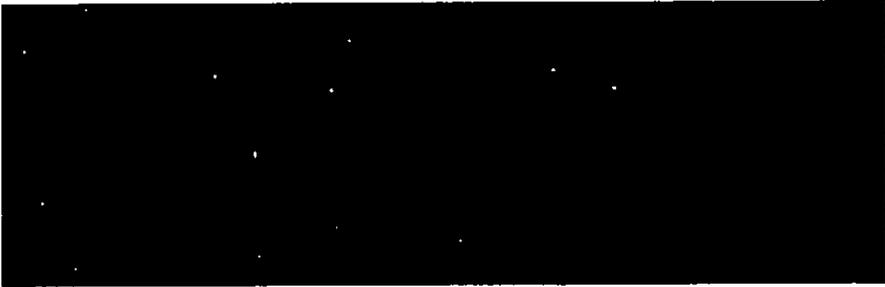
Es en este punto donde los adelantos tecnológicos permiten dar solución al manejo de grandes volúmenes de información. Obteniéndose óptimos resultados haciendo uso de los sistemas manejadores de base de datos relacionales tal como es el caso de Informix, el cuál es una herramienta apropiada para el desarrollo de aplicaciones, ya que por su gran potencial ha ayudado a empresas e instituciones a conseguir sus objetivos de manera más rápida y eficiente.

La estructura del presente trabajo consta de tres capítulos :

En el capítulo 1, se definen cada uno de los conceptos que servirán para el desarrollo del sistema de la División de Ingeniería Mecánica e Industrial de la Facultad de Ingeniería. Se comienza dando los antecedentes de las bases de datos con el fin de presentar la evolución que han tenido, para complementar la información de éstas se definen los manejadores de base de datos de tipo relacional que existen dentro del mercado computacional ; posteriormente para explicar de una manera clara y precisa tanto su utilización como los alcances de estos manejadores, se ejemplifica basándose en los tres principales, los cuáles son : Oracle, Ingres e Informix.

Dentro del capítulo 2 se amplía la información acerca del manejador de base de datos relacional Informix, destacando su importancia en la industria computacional y el por qué de su utilización para desarrollar el sistema que dará solución al problema de información en la DIMEI.

Para finalizar, el capítulo 3, especifica el desarrollo del ciclo del sistema. Se procede a la realización de un análisis detallado y profundo de las funciones del sistema, haciendo uso de la metodología de análisis estructurado. Una vez concluida la fase anterior, se pasará directamente al diseño de su estructura, el cual involucra la utilización de las herramientas de la programación estructurada, tal como es el caso de la modularidad y del diseño top-down. En la última etapa, se presentan las diferentes formas de pruebas que se deben realizar a un sistema para hacerlo confiable y eficiente.



## CAPITULO 1. CONCEPTOS BÁSICOS.

### 1.1 ANTECEDENTES.

El comienzo del almacenamiento de datos fue el archivo secuencial. Se iba leyendo el archivo desde el principio, hasta que se encontraba el dato requerido, y, entonces, se rebobinaba la cinta dejándola preparada para el próximo acceso. A medida que fueron emergiendo dispositivos de almacenamiento más sofisticados, tales como tambores y discos, los lenguajes de programación se fueron extendiendo para incluir sentencias que hicieran posible el acceso directo; en algunos casos se denominaba, por razones incomprensibles, *acceso aleatorio*.

La existencia de los archivos indexados hizo posible conceptualizar una relación estructural entre estos archivos, que capturarían parte de la estructura del mundo real o de la aplicación. Puesto que la pertenencia a una clase es un componente estructural tan evidente del mundo, los primeros productos de bases de datos impusieron una estructura jerárquica en sus archivos. En los sistemas de negocios esto solía producirse en la forma de *apariciones* de un cierto tipo ; por ejemplo, una compañía con varias sucursales. Las bases de datos jerárquicas, tales como Information Management System (IMS, Sistema de Información Gerencial), eran a la vez populares y eficientes. Sin embargo, muchas relaciones de negocios no encajaban en elegantes jerarquías, y era frecuente que emergiesen redes más generales después de las investigaciones

de los analistas de sistemas. Esto dio lugar al desarrollo de base de datos en red, cuyo ejemplo podría ser Total y cuya máxima expresión son los estándares del comité CODASYL (Conference on Data Systems Languages : conferencia sobre lenguajes de sistemas de información). Ambos tipos de sistemas, sin embargo, dependían de punteros fijos, y eran muy difíciles de modificar o de extender como respuesta a la reorganización de los negocios.

En la Figura 1.1, se muestra una jerarquía de tipos de registro, representativa de la forma de estructura que se encuentra en un Sistema Manejador de Bases de Datos (DBMS) jerárquico. Los cuatro tipos de registros de la Figura 1.1, representan las oficinas filiales de una empresa, los automóviles asignados a cada una, los empleados autorizados a conducir un automóvil específico y las fechas en que se les dio mantenimiento. Aunque en la figura no se muestra cada una de estas casillas, representa un conjunto de campos, como debe suceder con cualquier registro. Por ejemplo, cada registro de **oficina** tiene un campo de **número**, uno de **nombre**, uno de **ciudad de la oficina**, y así sucesivamente. Los registros de **automóvil de oficina** y **empleado** están identificados por campos llave únicos (p. ej., **número de oficina**).

Un dogma básico de la estructura jerárquica de datos es que cada enlace entre dos tipos de registros representan una relación uno a muchos, en dirección descendente. De esta forma, la estructura de la Figura 1.1, indica que cada **oficina** tiene asignados algunos automóviles, por cada uno de ellos hay varios

empleados autorizados a conducirlos, y cada automóvil tiene un historial de múltiples fechas de mantenimiento.

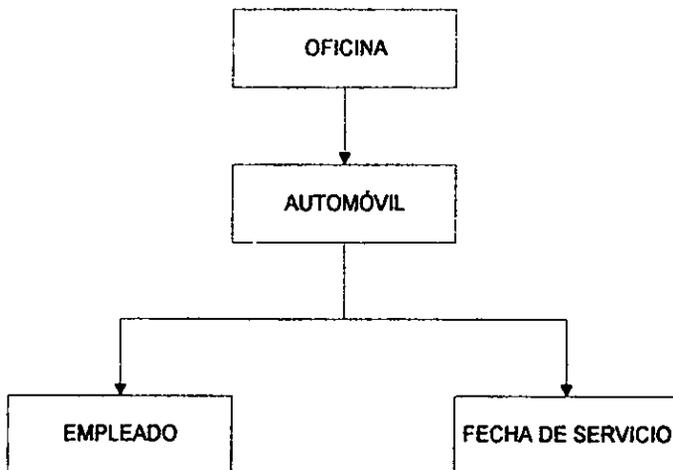


Figura 1.1 Jerarquía de datos.

La disminución de posible redundancias con ese arreglo surge, primero, del hecho de que los detalles de un registro superior específico (p. ej., un registro de **oficina**) se almacena sólo una vez y puede integrarse con los datos de cualquier ocurrencia de sus registros subordinados (p. ej., **automóvil**). En un caso supuesto, se podría combinar la dirección de **oficina** con detalles acerca de cualquiera de los automóviles asignados a ésta.

En la Figura 1.2, se muestra una red de datos. Como en el caso jerárquico, cada enlace entre dos tipos de registro representa una relación uno a muchos entre

ellos; la dirección de ésta se muestra por medio de las puntas dobles de flecha. A diferencia del caso jerárquico en las redes es común considerar que dos tipos de registro conectados y el enlace entre ellos constituyen una subestructura denominada *conjunto*, que tiene ciertas propiedades valiosas.

Los sistemas de manejo de base de datos que se fundamentan en los enfoques jerárquicos o de red, a menudo se denominan sistemas de *navegación*. En ambos casos, los datos se almacenan como registros de distintos tipos que están interconectados por medio de apuntadores de dirección.

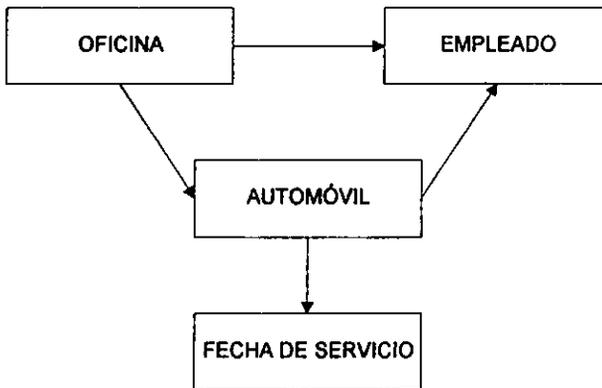


Figura 1.2 Red de datos.

El siguiente desarrollo fue la introducción del primer modelo formal de datos, el modelo relacional de Ted Codd (Codd, 1970). El estándar establecido ahora para la definición y manipulación de datos, SQL, está basado en el cálculo

relacional. Se pueden añadir atributos, y también es posible eliminarlos, sin necesidad de reconstruir sistemas complejos de punteros, y las bases de datos relacionales hacen posible redefinir los sistemas de negocios y las estructuras organizativas para conseguir mayores ventajas competitivas en un mercado que cada vez es más internacional y complejo.

El modelo relacional de datos fue producido con varios objetivos entre los cuales se cuenta el deseo de utilizar métodos formales en el diseño de base de datos, en las consultas y en las actualizaciones; por el deseo de ser capaz de probar la corrección de los programas basados en descripciones no procedimentales. La idea básica es que los datos se representan como una serie de tablas o de *archivos planos*. No deben permitirse grupos repetitivos ni jerarquías implícitas, y ningún enlace estructural fijo debe formar parte de la base de datos. Las relaciones lógicas entre los datos se construyen en el momento de la ejecución, o bien se almacenan en las propias tablas. Por tanto el mismo tipo de objeto se utiliza para representar tanto las entidades como las relaciones. Además, estas tablas de referencias cruzadas pueden ser reconstruidas sin necesidad de reorganizar los datos básicos. Ésto es una gran ventaja en las bases de datos que modelan a empresas sometidas a grandes cambios organizativos.

En la Figura 1.3, se muestra una base de datos relacional; a primera vista parece que consiste en unos cuantos archivos lineales independientes, y en realidad, esto es cierto desde un punto de vista estructural. Obsérvese también que ya que los campos **oficina**, **automóvil**, **empleado** y **fecha de**

únicos, existen sólo un registro para una oficina específica y (está en el archivo de oficina), uno para un cierto automóvil (en el archivo automóvil), y así progresivamente. Esto, desde luego, significa que no hay redundancia entre los campos no llave; ya que existe sólo un registro para una oficina específica, todos los campos que dependen de él, como **gerente**, **dirección** y así sucesivamente aparecen sólo una vez.

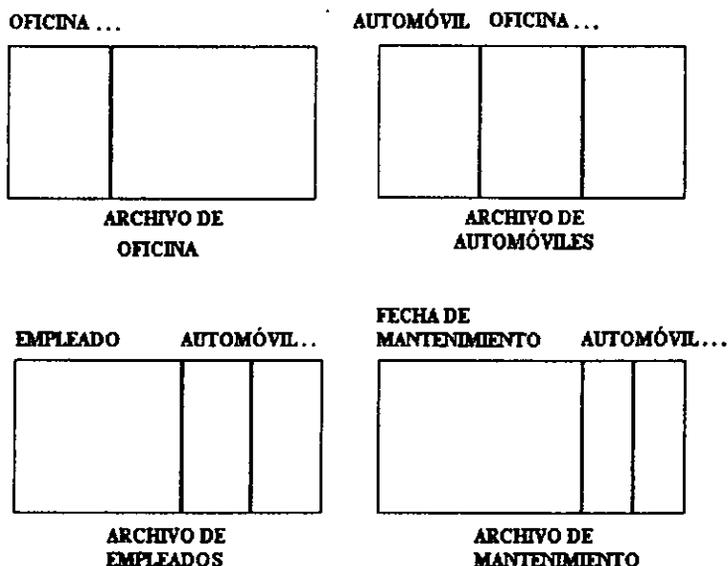


Figura 1.3 Estructura relacional.

## 1.2 BASE DE DATOS.

Aunque administración de base de datos es un término informático, también se puede aplicar a las distintas formas en que se clasifica la información. En todo sistema de administración de información existe una base de datos.

*“Una base de datos es una colección organizada de información que guarda relaciones entre sí, se agrupa integrando conjuntos y sirve para diferentes propósitos. Concretamente, la base de datos es la colección completa de datos, punteros, tablas, índices, etc”.<sup>1</sup>*

La base de datos es un modelo representado por un conjunto de datos, los cuales tienen las siguientes propiedades : están interrelacionados, son capaces de evolucionar, son accesibles a múltiples aplicaciones y su redundancia es mínima y controlada.

El que estén interrelacionados significa que se puede tomar cualquier elemento de información y cruzarla con cualquier otro elemento de información de la base de datos.

El ser capaces de evolucionar significa que en cualquier momento se pueden modificar las características de los datos, su estructura o bien agregar nuevos elementos de información, siendo este cambio transparente para los usuarios.

---

<sup>1</sup> Notas tomadas de la clase de base de datos impartida por la profesora Judith Jaramillo, UNAM, ENEP “ACATLAN” [95-1]

Al ser accesibles a múltiples aplicaciones tendremos en la misma base de datos la información de toda la empresa y esta información será accesada por todas las aplicaciones que se desarrollan, por ejemplo: nómina, facturación, inventarios, contabilidad, etc.

La redundancia se refiere a la repetición de los datos. Idealmente en los sistemas de datos se pretende que cada elemento de información se almacene una sola vez, esto es cero redundancia. Sin embargo por cuestiones del modelo o de eficiencia esto no siempre es posible, en la realidad se han llegado a repetir los datos el menor número de veces posible y se ha logrado controlar que cada una de estas repeticiones tenga el mismo valor.

En una base de datos la información se organiza y se mantiene en una tabla compuesta por filas y columnas. En la Figura 1.4, por ejemplo, se observa una lista de correos en forma de base de datos, cada fila conteniendo un **nombre**, una **dirección**, un **código postal** y un **número de teléfono**. Cada fila esta relacionada con las otras porque todas ellas contienen el mismo tipo de información establecida en un orden determinado ( una columna de **nombres**, una columna de **direcciones**, una columna de **códigos postales** ), por lo que se trata de una base de datos.

Las filas en un archivo de base de datos se llaman *registros* y las columnas se llaman *campos*.

Un *registro* es un grupo de campos afines de información considerados como una unidad.

Un *campo* identifica una posición en un registro en donde un dato elemental es almacenado. Este campo tiene ciertas características, tales como el largo y tipo (una cadena o un tipo numérico).

NOMBRE	DIRECCION	C.P.	TELÉFONO
J. BARRERA	ORO # 2345	94350	333-56-78
A. FLORES	LAGO JUI # 34	54970	569-45-67
M. MONTES	CHICONCUAC # 9	85432	882-46-90
B. OROZCO	TELLOCAN # 345	94525	725-25-32
L. ORTEGA	DUARTE # 25	25210	325-25-25
J. RINCÓN	AZTECA # 22	67899	845-89-65
M. SÁNCHEZ	BOLIVIA #35	57800	714-73-56
U. HERRERA	ERMITA #56	54185	891-32-14

Figura 1.4 Base de datos simple. <sup>2</sup>

Cuando la base de datos se implementa en una computadora, muchos problemas desaparecen, una base de datos informatizada proporciona velocidad: buscar un número telefónico entre un millar de entradas lleva menos

<sup>2</sup> Notas tomadas de la clase de base de datos impartida por la profesora Judith Jaramillo, UNAM, ENEP "ACATLAN" [95-1]

de dos segundos y ordenar una base de datos con cien mil elementos requiere menos de 1 minuto. Una base de datos informatizada es compacta : una base de datos con unos cien mil elementos puede ser almacenada en un pequeño disco, además es flexible y eficiente.

En la actualidad el manejo de información debe hacerse de una manera eficaz y rápida, por lo cual, el uso de base de datos relacionales tiene su mayor aceptación en las empresas.

Una base de datos relacional es aquella que establece relaciones entre dos o más entidades de datos evitando con ello la duplicación de la información.

### **1. 2.1 SISTEMAS MANEJADORES DE BASES DE DATOS.**

Un sistema manejador de bases de datos DBMS (Database Management System) es un conjunto de programas especializados y diseñados para describir, proteger, almacenar y acceder una base de datos superando las limitaciones del procesamiento tradicional de archivos, garantizando la seguridad, integridad y protección de los datos, así como sincronizando el acceso de múltiples aplicaciones.

Por seguridad se entenderá que sólo podrán acceder la base de datos los usuarios autorizados y cada usuario autorizado verá sólo la parte que le corresponde. La integridad garantiza que los valores almacenados son correctos

y que la base de datos es consistente. La protección asegura que en caso de *corrupción* de la base de datos, es posible recuperar los datos correctos. Finalmente la sincronización controla que varias aplicaciones accedan en forma concurrente la base de datos.

Si una base de datos se diseña, implementa y mantiene en forma correcta, un DBMS puede ayudar a una organización a aumentar su habilidad para responder a las necesidades cambiantes de información. Con la base de datos y el DBMS, el sistema de información se convierte en un sistema de base de datos esquematizado como se observa en la Figura 1.5.

El sistema manejador de la base de datos se encarga de :

- La creación original de la descripción de la estructura de la base de datos y la forma en que ésta es reflejada por los archivos de la base de datos física.
- Establecer y mantener las trayectorias de acceso a la base de datos, de tal manera que los datos en cualquier parte de la base se puedan acceder rápidamente.
- Manejar los datos de acuerdo con las peticiones de los usuarios.
- Mantener la integridad y seguridad de los datos.

- Modificar la descripción de la base de datos, así como realizar copias de la misma.

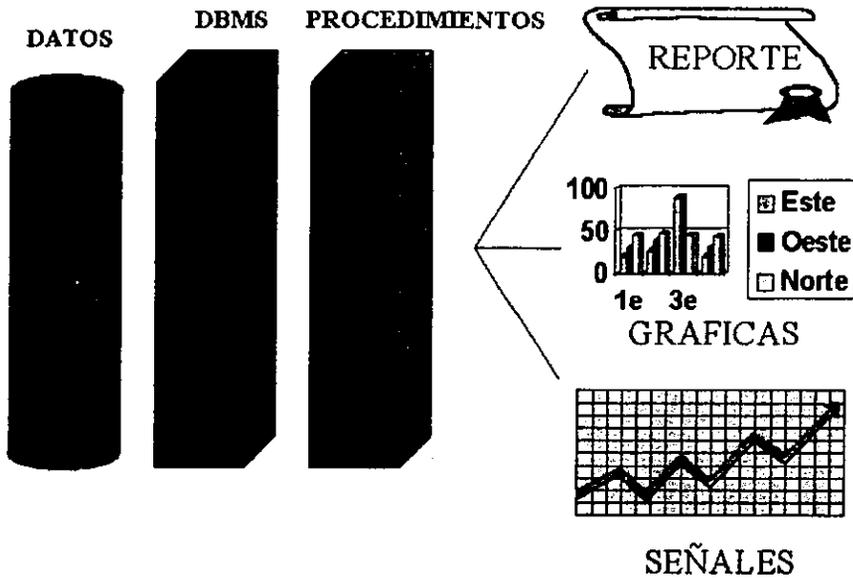


Figura 1.5 Sistema de base de datos.

El DBMS opera bajo tres niveles los cuales son : DDL, DML y DQL.

El lenguaje de definición de datos (DDL), es utilizado por el administrador de la base de datos (DBA) para la creación de la base de datos, tablas, vistas y para establecer las medidas de seguridad en el acceso a las mismas.

El lenguaje de manipulación de datos (DML), permite al programador de aplicaciones consultar, insertar, actualizar la información y eliminar registros dentro de la base de datos.

Y por último el lenguaje de consulta de datos (DQL), es utilizado por el usuario final o casual para realizar consultas a la base de datos.

### **1.2.2 CARACTERISTICAS Y OBJETIVOS.**

Un DBMS proporciona una visión abstracta de la información al usuario. Es decir, el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan y mantienen.

La búsqueda de la eficiencia conduce al diseño de estructuras de datos complejas para representar la información en la base de datos. Pero como los sistemas de bases de datos muchas veces son utilizados por personal que no cuenta con conocimientos de computación, esta complejidad debe permanecer escondida para los usuarios.

Dentro de las características que un DBMS debe incorporar se encuentran :

- ♦ Independencia de los programas respecto a los cambios efectuados a la estructura de los datos.

- ♦ Programas de utilería para la administración de la base de datos.
- ♦ Mecanismos de seguridad para imponer límites de acceso.
- ♦ Recuperación en caso de fallas.
- ♦ Facilidades para afinación (Tuning) de la base de datos.
- ♦ Un lenguaje de consulta propio.
- ♦ Capacidad para proceso de transacciones en línea (DLTP).
- ♦ Diccionario de datos.
- ♦ Control de concurrencia.
- ♦ Facilidad de acceso.
- ♦ Protección de los datos.

El objetivo primordial de un DBMS es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma conveniente y eficiente. Es decir el DBMS debe cumplir con los siguientes objetivos :

1. Organizar los datos con sencillez.

La estructura de los datos en un DBMS debe ser tan simple como sea posible. Cuanto más sencillo sea la estructura de la base de datos, tanto más fácil será manipular los datos.

2. Reducir los costos.

La mayoría de los DBMS disminuye la necesidad de almacenamiento en disco y de grupos de programadores. Aunque un DBMS reduce la redundancia de datos, podría pasarle inadvertido el espacio de disco extra.

El ahorro real radica en el tiempo de programación. Con un DBMS, la solución de problemas simples tarda apenas unas horas y no días. Aplicaciones complejas requieren sólo de un programador en lugar de un equipo.

3. Minimizar la redundancia.

Una de las principales diferencias entre un sistema de información convencional y un sistema de base de datos es que en este último, ningún programa tiene sus propios archivos privados. Todos los datos de los distintos usuarios se unen e integran en la base para ser compartidos por todos. Así se minimiza la redundancia de los datos.

4. Garantizar la consistencia.

El DBMS permite obtener la misma información por peticiones similares en un momento dado.

5. Integridad de los datos.

Las áreas importantes en lo que concierne al reforzamiento de la integridad de los datos son: la validación de los datos de entrada, la integridad referencial, la interferencia debido a la concurrencia y los bloqueos. Cada DBMS tiene diferentes enfoques y grados de sofisticación para reforzar la integridad de los datos. Es decir, la integridad se refiere a las reglas dictadas por políticas o normas de la empresa y que los datos deben cumplir.

6. Seguridad de los datos.

Debido a que la base se comparte extensamente, el DBMS protege los datos individuales de cada usuario contra la intrusión o destrucción no autorizada. El administrador de la base de datos (DBA), controla los derechos de acceso de los programadores a ciertas porciones de la base, declarando los elementos de datos apropiados a un esquema determinado.

FALTA PAGINA

No.

27

aquellos que utilizan nuevos elementos. La mayoría de las demandas en un sistema de explotación son solicitudes de nuevos informes".<sup>4</sup>

### 11. Permitir la utilización compartida de datos:

El DBMS debe permitir que los datos sean compartidos. En una organización, un DBMS debe permitir a personas diferentes usar los mismos datos. Quizás el personal de compras y el encargado de las cuentas por pagar puedan compartir la misma base de datos de proveedores.

### **1.3 TIPOS DE RDBMS (Sistema Manejador de Base de Datos Relacional).**

Los modelos de redes y jerárquicos establecen las conexiones entre la información a través de apuntadores, minimizando la redundancia. El uso de apuntadores reduce el tiempo de acceso a la información, pero al introducir o eliminar datos consumen mucha memoria. Por otro lado, el modelo relacional utiliza archivos para almacenar los identificadores (o llaves) de los elementos (o entidades) relacionados, duplicando la información de las llaves en la base de datos. La filosofía relacional ha sido la más difundida por su facilidad de implantación y desarrollo, y por no requerir demasiados recursos (en comparación con los otros modelos). Lo anterior justifica el uso extensivo de este modelo en el mercado de las microcomputadoras.

---

<sup>4</sup> IBIDEM, Pg. 104.

En la actualidad existen diversos sistemas manejadores de base de datos relacionales entre los que se encuentran ORACLE, INFORMIX e INGRES, los cuales utilizan para el acceso a la base de datos un lenguaje llamado *Structured Query Language*, conocido también como SQL.

SQL se ha convertido en un estándar y la mayoría de los DBMS relacionales lo utilizan, en algunos casos como lenguaje principal y en algunos otros como lenguaje alternativo o para comunicación con otros DBMS.

SQL contiene tanto instrucciones de DDL como DML y se puede utilizar a nivel intérprete o bien inmerso en un lenguaje anfitrión. Las operaciones de SQL funcionan casi siempre en términos de conjuntos y no de registros individuales, y tampoco incluyen ninguna referencia a rutas de acceso explícitas, lo que lo hace bastante amigable para los usuarios.

En la siguiente sección se describirá de forma general cada uno de los DBMS relacionales mencionados anteriormente.

### **1.3.1 ORACLE, INFORMIX, INGRES**

#### **1.3.1.1 ORACLE**

**a) Descripción general.** Oracle es un producto registrado por Oracle Corporation de Belmont, California. Fue el primer RDBMS comercial

disponible que ofrecía SQL (el llamado SEQUEL) como lenguaje de consulta. El DBMS corre sobre una gran variedad de hardware y sistemas operativos tales como UNIX, MS-DOS y VMS. El producto básico de Oracle incluye RDBMS (el kernel), SQL\*PLUS lenguaje de consulta, SQL\*Forms diseñador de pantallas, SQL\*Report diseñador de reportes, utilerías para la importación/exportación de la base de datos en respaldos y recuperaciones de información e interfaces con lenguajes de tercera generación, en los que se incluyen COBOL, C, FORTRAN y PASCAL. Utilerías adicionales son SQL\*Menu diseñador de menús, SQL\*Calc software de hoja de cálculo, SQL\*Net herramienta de red, SQL\*Design la cual es una herramienta CASE y SQL\* Graphs paquete manejador de gráficas.

Todos los títulos precedidos con "SQL" son interactivos y manejan comandos avanzados para usuarios con experiencia. Por el contrario, títulos con prefijo *Easy* (p. ej. *Easy\*SQL*) indica los productos que guían al usuario a través de su trabajo, ofreciendo ayuda detallada en línea.

**b) Definición de datos.** Las tablas en una base de datos son creadas usando el comando CREATE TABLE. Esto incluye la siguiente información :

1. Nombre de la tabla.
2. Nombre de cada columna.
3. Tipo de datos a ser almacenados en cada columna.
4. Ancho de cada columna.

5. Información adicional, tal como NOT NULL, para indicar que el diseño de las columnas no pueden aceptar valores nulos.

El comando contiene tres cláusulas opcionales :

1. SPACE: Especifica la definición del espacio de almacenamiento.
2. PCTFREE : Especifica el porcentaje de espacio libre para actualizaciones. Cuando un renglón es insertado en una tabla, Oracle crea un nuevo bloque para almacenar el renglón, si la inserción es en un bloque existente debe ser menor al porcentaje de espacio libre en el bloque.
3. CLUSTER: Especifica que la tabla esta incluida en el diseñador de agrupamiento.

**c) Manipulación de datos.** Permite la manipulación de datos (recuperación y actualización) en tres formas:

1. Interfaz manejador de menú (Easy\*SQL)
2. Interfaz manejador de comandos (SQL\*Plus)
3. Interfaz manejador de formas (SQL\*Forms)

En el método 1 el usuario selecciona el menú de opciones y responde a las proposiciones dadas por el software. En el método 2 el usuario utiliza comandos de SQL estándar, tales como SELECT, INSERT, DELETE y UPDATE. En este método son posibles realizar consultas y recuperaciones

simples como complejas. INSERT adiciona un registro a la vez, mientras que INSERT AS SELECT adiciona múltiples registros retornados por la sentencia SELECT, la sentencia DELETE y UPDATE trabaja sobre un grupo de renglones basados en un criterio especificado por el usuario.

En el método 3 el usuario recupera datos por medio de una consulta basada en una forma de una o más tablas donde se encuentra el dato. Triggers pueden ser incluidos a la forma para la validación de los datos. Así la integridad referencial puede forzarse a través de éstos. Oracle no hace uso del comando JOIN, aunque esta es una propiedad de la sentencia SELECT.

Las vistas son definidas de una o más tablas con el comando CREATE VIEW y son manejadas de la misma manera que las tablas. Oracle permite la actualización de tablas a través de vistas en los siguientes casos :

1. Los registros pueden ser borrados de una tabla a través de una vista, si la consulta usada para definir la vista contempla estas restricciones :

Si los registros seleccionados forman parte de una tabla.

Si no contiene las cláusulas GROUP BY y DISTINCT o hace referencia a la pseudo-columna ROWNUM.

2. Los registros pueden ser insertados en una tabla a través de una vista, si la consulta utilizada para definirla contempla las restricciones antes mencionadas y lo siguiente :

Si existe en la vista una columna de la tabla definida como NOT NULL.

Los reportes pueden ser generados de tres formas :

1. Usando el comando de reportes SQL\*Plus.
2. Usando la utileria SQL\*Report, la cual es completamente procedural, o SQL\*Report Write, el cual está basado en el manejo de menú y pantalla.
3. Usando una Interfaz 3GL tal como Pro\*Cobol o Pro\*C (precompiladores) .

**d) Desarrollo de aplicaciones.** Las aplicaciones basadas en Oracle son mejores si se usa una combinación de sus utilerías. Por ejemplo, menús y pantallas pueden ser desarrollados con SQL\*Forms, y los reportes pueden ser generados con SQL\*Report o SQL\*Report Writer. Alternativamente, el programador puede usar programas consistentes de código de lenguajes de tercera generación e incluir bloques de sentencias de SQL.

Asimismo, Oracle puede operar con base de datos distribuidas, posee facilidades para realizar procesos distribuidos y compartir recursos, integra una interfaz para el uso de redes de área local, así como hojas de cálculo.

Sin embargo, debido a los altos recursos necesarios para su utilización, no es posible instalarlo en equipos PC AT sólo bajo ciertas configuraciones.

### 1.3.1.2 INFORMIX.

a) **Descripción general.** Informix está registrado por Informix Software, Inc., aunque inicialmente fue pensado para ambiente UNIX, el vendedor tiene ahora los soportes para otros computadores, tales como IBM, DEC, AT&T y Data General. Los productos básicos de DBMS provistos por el vendedor son Informix-SQL, Informix-ESQL/COBOL, Informix-4gl, y Informix-ESQL/C. Tanto Informix-4gl como Informix-SQL ofrecen una forma generadora y un lenguaje de programación basado en SQL. Pero Informix-4GL no tiene separado el componente REPORT como Informix-SQL lo tiene. Informix-ESQL (la E representa *empotrar*) permite a un programador escribir programas en un sin número de lenguajes como C o COBOL usando bloques de instrucciones de SQL para corregir o actualizar *propósitos empotrados* dentro de los programas.

b) **Definición de datos.** Informix proporciona tanto manejo de menú como interfaz de comandos orientados para definir las bases de datos. El usuario puede crear, modificar y borrar tablas en una base de datos a través de la interfaz del manejo de menú, por medio de los comandos introducidos interactivamente de RDSQL ( RDSQL es un SQL basado en lenguaje de consulta ), o a través de los comandos incluidos en RDSQL en un programa de 4GL. La interfaz del manejo de menú es útil para un principiante pero es demasiado lenta para un usuario experimentado. Sin embargo una buena característica de la interfaz del manejo de menú es que permite al usuario preguntar a Informix como crear un *archivo de*

*comandos* RDSQL basado en la entrada interactiva del usuario. Este archivo puede entonces ser salvado y modificado usando un editor de texto durante la fase del diseño de la base de datos, con lo cual se interactua en forma natural.

c) **Manipulación de datos.** Informix ofrece dos métodos para la manipulación de los datos : una es el manejo de la forma en la interfaz y un comando orientado a la misma. El manejo de la forma en la interfaz incluye generalmente una forma por default con el generador de Informix y después la procesa con una sentencia PERFORM. Esta despliega un menú sobre la pantalla con opciones tales como QUERY, ADD, UPDATE, y REMOVE, con lo cual entonces puede ser usado para corregir o actualizar los datos. Las interfaz de los comandos orientados usan los comandos estándar de SQL tales como SELECT, INSERT, DELETE, y UPDATE. El usuario introduce estos comandos interactivamente para corregir o actualizar los datos.

Asimismo el usuario puede crear y borrar vistas con los comandos de RDSQL. Una tabla puede ser actualizada a través de una vista excepto en las siguientes dos situaciones :

- La vista dependa de múltiples tablas.
- La definición de la vista involucre JOINS, la cláusula GROUP BY, la palabra clave DISTINCT, o una función agregada tal como SUM, AVG, o COUNT.

**d) Desarrollo de aplicaciones.** Tal como en la definición de datos como en la manipulación de datos, Informix ofrece dos métodos para el desarrollo de aplicaciones. Uno de los métodos utiliza la forma generadora y la ejecuta utilizando el comando PERFORM. La validación apropiada para las rutinas puede ser agregada usando el control de bloques de la forma. Sin embargo, este método tiene las siguientes cuatro limitaciones :

1. Sólo una de las formas puede ser ejecutada en un tiempo, no permitiendo la jerarquía de las formas.
2. Los menús presentados sobre la forma no pueden ser personalizados.
3. Las operaciones complejas y lógicas no pueden ser incluidas.
4. Las instrucciones del acceso a la base de datos no pueden ser incluidas.

El otro método utiliza el lenguaje propio de Informix-4gl complementándose con rutinas de C o COBOL, haciendo este método más completo y eficiente.

### **1.3.1.3 INGRES.**

**a) Descripción general.** Ingres (Interactive Graphics and Retrieval System) esta registrado por Relational Technology, Inc. (RTI) of Berkeley, California. Fue comenzado como un proyecto de investigación en la

Universidad de California en Berkeley en 1973 y utilizado en el sistema operativo UNIX. En el presente Ingres esta disponible para otros sistemas operativos, tal como VMS, VM/CMS, y MVS. Opera como un DBMS multiusuario, además una versión para PC también esta disponible.

Los componentes principales del Kernel incluyen lo siguiente :

RDBMS ( DBMS relacional )

Ingres/Menú (Generador de Menú)

Ingres QBF (Consulta de formas)

Ingres VIFRED (Editor visual de formas interactivas)

Ingres REPORTS (Generador de reportes)

Ingres REPORTS RBF ( Reporte de formas )

Ingres ISQL/ IQUEL (Monitor terminal SQL/QUEL interactivo )

ESQL/EQUEL ( Preprocesador para C).

Entre los componentes agregados podemos mencionar

Ingres /PCLINK

Ingres/GRAPHICS

ESQL/EQUEL Preprocesador para COBOL, BASIC, ADA, PASCAL, FORTRAN, y PL/I

ABF (Aplicación para formas)

b) **Definición de datos.** Una base de datos es creada por el usuario usando el comando `CREATEDB` en el nivel del sistema operativo. Tanto el manejador del menú como las interfases del manejo de comandos están disponibles para definir y borrar relaciones. Una combinación de distintos comandos puede ser utilizada para agregar, borrar, cambiar, o renombrar un atributo. Así las definiciones relacionales no pueden ser modificadas de la interfaz del manejo de menú.

El usuario puede acceder a la interfaz del manejo del menú seleccionando la opción `TABLES` de `Ingres/Menú`. La interfaz del manejo del comando esta disponible en `ISQL`. El usuario puede meter una o más declaraciones en una pantalla en blanco y pedirle a `ISQL` que la ejecute.

c) **Manipulación de datos.** `Ingres` ofrece dos herramientas para la manipulación de los datos : `QBF` la cuál es una forma orientada y `ISQL` el cuál es un comando orientado. El usuario puede trabajar con una relación en un simple formato de archivo, el cuál despliega un registro, o dentro de una tabla con formato de archivo, con la cuál se despliegan varios registros.

Si el usuario selecciona la ruta `QBF` y elige una relación, una forma en blanco correspondiente a la relación aparece sobre la pantalla. El usuario puede insertar nuevos registros para datos de entrada sobre la pantalla. Para borrar o modificar registros existentes él o ella pueden cambiarlos seleccionando las opciones en la pantalla y haciéndolos de acuerdo a como los requieran. La forma usada por `QBF` debe de ser generada con `VIFRED`.

Si el usuario selecciona la ruta ISQL, los comandos estándar de SQL, tal como SELECT, INSERT, DELET y UPDATE son utilizados. Ingres tiene el soporte de dos métodos para el manejo de JOINS. El primero es con la cláusula SELECT (comando orientado) ejecutado a través de ISQL. El segundo método es con entidades llamadas Joindef definidas y manipuladas desde QBF (forma orientada). Un Joindef incluye la definición de un Join, en una forma para consultarla, agregarla, cambiarla, o borrar los datos dentro de las tablas.

El usuario puede definir vistas usando la cláusula SELECT de ISQL. En general, Ingres tiene los soportes para actualizar los datos a través de las vistas solamente si puede ser garantizado (sin buscar en los datos actuales) que el resultado de la actualización de la vista puede ser idéntico a la actualización de las correspondientes relaciones. La actualización de los datos a través de las vistas no están permitidos si cualquiera de estas condiciones se cumplen :

1. La vista tiene más de una relación.
2. La actualización afecta a una columna que esta dentro de la condición de búsqueda de la definición de la vista.
3. La actualización afecta a una columna dentro de la vista que no es una columna de la base de datos. Por ejemplo, la columna podría ser el resultado de una función de conjunto (p.ej. max, min, avg) o de operaciones.

QBF permite al usuario corregir datos a través de las vistas. No es posible agregar, cambiar, o borrar datos dentro de una vista desde QBF ; esto podría realizarse a través de ISQL. Uno no puede construir índices sobre las vistas.

Ingres proporciona tres métodos para generar reportes : reportes por default, reportes generados por RBF, y reportes generados con el escritor de reportes. Un reporte por default incluye todos los registros y columnas de una tabla seleccionada. Para incluir un subconjunto de los datos de la tabla, el usuario crea una vista o una tabla permanente basada en la tabla original y usa la nueva tabla o la vista para el reporte. Se pueden seleccionar el estilo de salida de las columnas, bloques o *wrap*. La salida puede ser dirigida a la terminal, a la impresora, o a un archivo.

RBF trabaja con VIFRED . Un reporte RBF trabaja sobre una tabla o una vista de una manera diferente a la cláusula SELECT. El usuario primero pide a RBF genere el reporte por default y entonces lo edita. El reporte por default contiene cinco elementos formales : título, encabezados de las columnas, mostrar los datos sobre la línea detallada, encabezados y pies de páginas.

Para generar un reporte, el usuario primero lo escribe especificando el archivo que contiene los comandos del escritor, el cual almacena los datos hacer reportados y la manera en que podrán aparecer. Después que el archivo es creado es llamado dentro de la base de datos antes de correrlo.

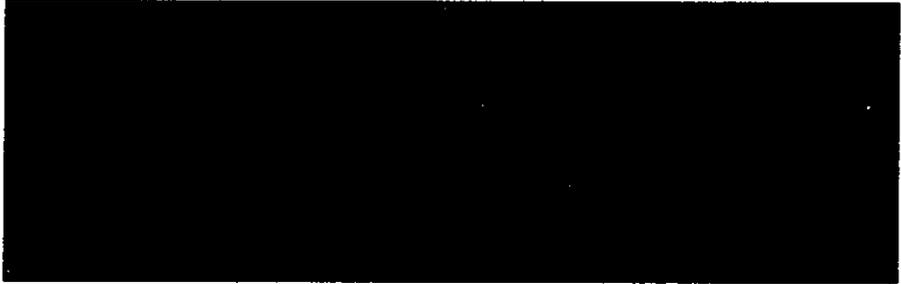
**d) Desarrollo de aplicaciones.** INGRES proporciona dos herramientas para el desarrollo de aplicaciones: ABF y ESQL. ABF es apropiado para aplicaciones de formas orientadas, y ESQL es idóneo para aplicaciones en formas y no formas orientadas.

1. ABF. El mas alto nivel conocido en ABF dentro de una aplicación. ABF mantiene una lista de componentes para cada aplicación. Una aplicación consiste de dos tipos de componentes: *frames* y procedimientos.

ABF soporta cuatro tipos de *frames*: usuario definido, QBF, VIGRAPH (no sobre PC), y reporte. Un usuario definido *frame* consiste de una forma y una especificación de un lenguaje en 4GL. Una especificación de un lenguaje de 4GL es un módulo escrito dentro de un lenguaje de programación de ABF, conocido como lenguaje de especificación de operación (OSL). Un QBF *frame* consiste de una forma. Una VIGRAPH *frame* consiste de una gráfica y, opcionalmente, el nombre de un archivo de salida y un comando de banderas de VIGRAPH. Un Reporte *frame* consiste de un reporte de un archivo específico, y opcionalmente el nombre de un archivo de salida, y una forma opcional para pasar los parámetros al reporte del archivo especificado. Un procedimiento para ABF esta escrito dentro de OSL o dentro de un lenguaje de 3GL tal como C, PASCAL, o FORTRAN, y puede ser llamado de una forma de 4GL o de otro procedimiento.

ABF proporciona un aplicación de menú orientado. Dentro de ABF el usuario puede invocar a VIFRED, VIGRAPH, y designar el editor.

2. ESQL, permite al usuario escribir programas dentro de lenguajes 3GL tal como COBOL, C, PASCAL, FORTRAN, Ada y empotrar el acceso a bases de datos y sentencias de control para formas. A diferencia de las aplicaciones en lenguajes de 4GL, donde los datos pueden ser pasados directamente entre las columnas de la base de datos y las formas, dentro de un programa ESQL el usuario necesita ir vía variables de programas. Los módulos ESQL necesitan ser corridos a través de un preprocesador proporcionado por INGRES. El preprocesador convierte las sentencias empotradas dentro de un lenguaje 3GL en sentencias conocidas. Las formas también tienen que ser preprocesadas. Todos los módulos preprocesados y formas están entonces compilados y ligados para crear un módulo que pueda ser corrido desde el sistema operativo.



## **CAPITULO 2. INFORMIX.**

### **2.1 DEFINICIÓN.**

Informix es un manejador de base datos que integra un SQL estándar y un lenguaje de cuarta generación (Informix-4gl) diseñado específicamente para el desarrollo de aplicaciones; puede ser utilizado en ambientes UNIX y MS-DOS, corriendo de manera natural en equipos HP-9000, PC's y Micro Vax.

Informix-4gl es un lenguaje de cuarta generación. A diferencia de C, Pascal, Fortran, Cobol y otros lenguajes de tercera generación que no están enfocados a aplicaciones, Informix-4gl ha sido diseñado específicamente para aplicaciones de base de datos.

### **2.2 CARACTERÍSTICAS**

Los lenguajes de cuarta generación (4gl) han recortado el tiempo de desarrollo de tal forma que se puede implantar soluciones mas complejas y completas en un tiempo muy breve.

Informix-4gl usa sentencias no procedurales tales como MENU o CONSTRUCT que permite a los programadores describir que es lo que quiere realizar sin tener que especificar los detalles de cada paso. Esto conlleva a

realizar aplicaciones de base de datos en un tiempo significativamente más corto comparado con los lenguajes de tercera generación.

Informix provee de sentencias procedurales o ciclos de control (p. ej. IF - THEN, WHILE, FOR ) para usarse cuando sea necesario. Esto permite al programador tener toda la funcionalidad de los lenguajes de tercera generación para adicionar características específicas dentro de sus aplicaciones.

El mantenimiento de aplicaciones escritas en un 4gl resulta muy sencillo dado el tamaño del código y por las estructuras no procedurales de 4gl.

Algunas características son :

Informix-4gl proporciona todas las herramientas que se necesitan para desarrollar en forma rápida prototipos y aplicaciones, tales como :

- Ambiente de desarrollo integrado y complejo.
- Métodos flexibles para el acceso de datos y complejas interfaces con el usuario final que le permiten concentrarse en la aplicación.
- Sistema de menús que le permiten ir de una pantalla a otra.
- Manejo de formas que le permite codificar rápidamente interfaces fáciles de usar.

- Manejo de ventanas que le permite ver el detalle de cierta información para cambiar de contexto.
- Validaciones de datos que pueden incorporarse rápidamente a un programa a partir de un diccionario de datos.
- Reportes no procedural que automáticamente manejan el formato y aspectos aritméticos de un reporte mientras el usuario se concentra en el contenido del mismo.
- Puede fácilmente construir un sistema de ayuda en línea (help) para que asista al usuario poco familiarizado con la aplicación.

Usando el Informix-4GL se puede fácilmente :

- ◆ Crear menús de anillo.
- ◆ Obtener datos desde formas de pantalla.
- ◆ Usar RDSQL para manipular la base de datos.
- ◆ Llamar pantallas de ayuda.
- ◆ Obtener varios renglones de datos usando una pantalla sencilla llamada scroll (corrimiento) de datos.

- ♦ Provee formas con query by example (ejemplos de consultas rápidas).
- ♦ Crear reportes query by example.
- ♦ Usar funciones de entrada y teclas de control.
- ♦ Llamar a librería de funciones de 4gl y C.

Las herramientas de desarrollo de Informix son totalmente independientes una de otra. En otras palabras, es posible desarrollar una aplicación como por ejemplo, en Informix-SQL sin necesidad de programar absolutamente nada, en Informix-4gl o en alguna otra herramienta.

Permite igualmente la portabilidad de las aplicaciones desarrolladas, a gran variedad de ambientes. Esto ocurre gracias a la disponibilidad de los productos Informix en un amplio rango de computadoras como los sistemas operativos DOS, Windows, UNIX y redes del DOS entre otros.

### 2.3 INFORMIX COMO RDBMS

Informix comienza con la creación del RDS (Relational Database System Inc) en el año 1980, por su fundador Roger Sirpppl, para producir C-ISAM (Indexed Sequential Access Method in C) sobre Unix y DOS. Este tuvo un rápido éxito, después de ser adoptado por Microfocus y de su estandarización por los grupos X/OPEN y SQL Access.

Un par de años más tarde RDS introduce en el mercado a Informix como un DBMS (Sistema Manejador de Base de Datos) relacional basado en C-ISAM para la gestión de datos. Este producto esta compuesto de cinco (5) aplicaciones:

APLICACION	FUNCION
formbuild	Generación y la compilación de pantallas
perform	Ejecución de formatos de pantalla
dbbuild	Creación de base de datos, tablas e índices
acceptred	Generación y compilación de informes
acego	Ejecución de los informes

Con el pasar del tiempo el DBMS de IBM SQL/DS y DB2 se fue convirtiendo en los lenguajes de referencia de los sistemas relacionales modernos.

Esto impulso a RDS a sustituir a Informix por SQL y anunció en el año de 1984 Informix-SQL.

Con el éxito de este producto, RDS terminó por adoptar el nombre de ISI (Informix Software Inc.). Dos años más tarde ISI presenta un lenguaje de cuarta generación, el Informix-4gl, que permite un rápido desarrollo de las aplicaciones, y dos lenguajes integrados ESQ/C y ESQ/COBOL permitiendo así que los programadores puedan formular sentencias SQL dentro de sus programas escritos bajo lenguaje C o en COBOL.

Informix, es un excelente ejemplo de sistema manejador de base de datos relacional, proporciona comandos para crear tablas de información. Después de haber creado una tabla se pueden insertar, borrar, o actualizar el contenido de uno o varios registros con una sola operación.

Existen dos tipos de Informix-4gl:

- Informix-4gl RDS “Rapid Develoment System”
  
- Versión compilada a “C” de Informix-4gl.

La diferencia se encuentra en la fase de compilación. Después de codificar el módulo (fuente), este es compilado y posiblemente linkeado a otro módulo.

Este proceso de compilación es diferente para los dos tipos de Informix-4gl.

### ***Informix-4gl Rapid Development System (RDS)***

El proceso de compilado Informix-4gl RDS toma el fuente y lo lleva a pseudo-código, el cual será interpretado para que el módulo funcione, obteniendo como resultado que el tiempo que va desde el compilado hasta la corrida sea muy corto. Esta es la razón de su nombre, sistema rápido de desarrollo o *Rapid Development System* ya que el tiempo de compilación es reducido dramáticamente.

### ***Informix-4gl C Compiler Version***

Al contrario de la versión RDS la versión compilada, el tiempo de compilación es sustancialmente largo, ya que el compilador toma el código fuente y lo convierte a ESQL/C que es una combinación de lenguaje C con instrucciones SQL, luego este es tomado y es llevado a lenguaje C puro, el cual será compilado para obtener el código objeto y de esta manera ser linkeado para tener como resultado final un módulo ejecutable. (Figura 2.1)

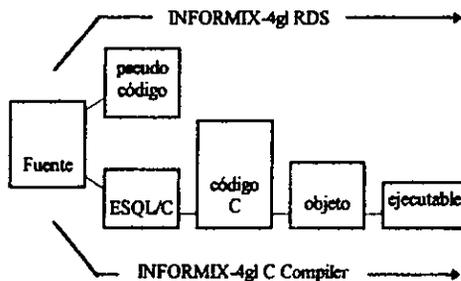


Figura 2.1 Fases de compilación de Informix-4GL

El pseudo-código generado por Informix-4gl RDS no es ejecutado directamente por el computador, éste es ejecutado a través de un interpretador. El interpretador toma las instrucciones Informix-4gl y las traduce en tiempo de ejecución y de esta manera las traslada a código entendible por el computador. Esto con lleva a que la ejecución de un programa en la versión RDS puede ser lenta, dependiendo del tipo de computador que se este utilizando y de la complejidad del fuente que se este desarrollando.

Con la versión compilada el tiempo de ejecución es muy rápido con respecto a la versión RDS, esto es debido a que el proceso de compilación toma el fuente en Informix-4gl y los transforma a instrucciones entendibles por el computador en forma de un programa ejecutable, esto quiere decir, que no se necesita de un interpretador.

Lo anterior queda ilustrado en la Figura 2.2

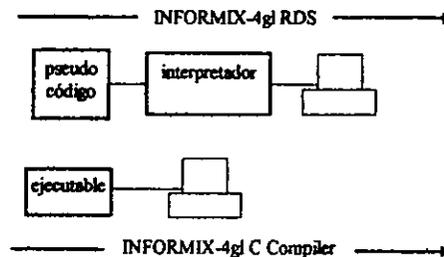


Figura 2.2 Tiempo de ejecución.

Para concluir esta sección de este capítulo podemos decir que la elección de alguno de los tipos disponibles de Informix dependerá sólo del tiempo que se desee aprovechar, si se trata de la compilación o de la ejecución. Muchas empresas adoptan la siguiente normativa, cuando se esta trabajando en un ambiente de desarrollo se usa la versión RDS y al entrar en producción se genera un programa ejecutable con la versión compilada en C.

## **2.4 PROS Y CONTRAS DE INFORMIX.**

En comparación con los manejadores de archivos tales como Dbase, Fox Pro, Clipper, etc, Informix proporciona las siguientes ventajas que lo hacen un RDBMS seguro y confiable:

- ♦ Integridad de la información
  
- ♦ Evita la redundancia de los datos
  
- ♦ Seguridad (Protección de datos contra accesos, modificaciones o pérdidas de manera intencional o no intencional).
  
- ♦ Consistencia de datos (Múltiples usuarios al mismo tiempo comparten la misma información).

- ◆ Recuperación (Capacidad de restaurar integridad y consistencia de datos después de una falla del sistema).
  
- ◆ Manejo de gran volumen de información.

En comparación con otros manejadores de bases de datos :

- Cuenta con un gran número de distribuidores, despachos respaldados y en condiciones de otorgar el soporte requerido a nivel nacional.
  
- El producto Informix se incluye en los planes de estudio a nivel superior del I.T.E.S.M., la UNITEC y como materia optativa en la Dirección General de Servicios de Cómputo Académico de la UNAM, esto origina un beneficio a mediano plazo, al tener personal de nivel licenciatura capacitado en Informix, se reduce el costo de entrenamiento a las empresas que utilizan el citado producto.
  
- El costo de la implementación de un sistema con Informix es más accesible a comparación de Oracle e Ingres.
  
- Es compatible con diferentes plataformas tales como UNIX y MS-DOS lo que permite migrar una aplicación de una plataforma a otra con un mínimo de cambios.

Desventajas del RDBMS Informix :

- ◆ Su interfaz es poco atractiva para el usuario, ya que las pantallas se realizan en modo carácter.
- ◆ En sistemas distribuidos el performance es menor cuando en el servidor se están ejecutando varios procesos.
- ◆ Informix depende de herramientas externas para realizar Front-ends orientados a eventos, tal como es el caso de Visual Basic ó Delphi.

## **2.5 ENTORNO DE PROGRAMACIÓN UNIX, DOS.**

### **2.5.1 UNIX.**

UNIX fue creado por los laboratorios Bell, UNIX es un sistema operativo interactivo multiusuario es decir, permite que varios usuarios ejecuten simultáneamente varios programas en el mismo computador ; interactivo se refiere a la repuesta casi inmediata del computador a la petición de una orden.

Los programas del sistema UNIX están funcionalmente clasificados de la siguiente forma :

- El *núcleo* que planifica tareas y administra el almacenamiento de datos.
- El *shell* es un programa que relaciona e interpreta las órdenes tecleadas por un usuario del sistema. Interpreta las peticiones de los usuarios, llama a los programas de la memoria y los ejecuta al mismo tiempo o en unas series de acoplamientos llamadas *tubo*.
- Los *programas de utilidad* ejecutan una variedad de rutinas y unas funciones especiales de mantenimiento del sistema.

Educadores, administradores, investigadores, ejecutivos y programadores adaptan y emplean sistemas UNIX para satisfacer sus necesidades específicas, entre las que se pueden incluir :

- ◆ Preparación e impresión de textos.
- ◆ Almacenamiento y manejo de documentos.
- ◆ Programación.
- ◆ Correo electrónico.

Puesto que el sistema operativo se diseñó por y para programadores, ofrece varias características fundamentales para el desarrollo de programas. Para

muchas aplicaciones, los programas se pueden escribir sin un lenguaje de programación. Los usuarios seleccionan y unen órdenes del sistema para crear programas específicos dentro del programa shell.

En los últimos años, el sistema operativo UNIX ha pasado de utilizarse únicamente en laboratorios de instituciones académicas y de investigación a tener una seria utilización comercial. UNIX ya no es sólo un entorno para profesores utilizado para el aprendizaje a los estudiantes, ni para *hackers* que deseen demostrar su gran habilidad programando. La aceptación comercial de un sistema operativo se reduce a que sea ampliamente utilizado en aplicaciones de procesamiento de datos. De hecho, el UNIX se está convirtiendo en el entorno operativo estándar de las aplicaciones de procesamiento de datos, que estuvieron muchos años dominadas por sistemas operativos específicos del vendedor. Esta situación a creado una feroz competencia entre vendedores para proveer de sistemas de administración de bases de datos a una gran cantidad de aplicaciones.

Informix ha sido uno de los primeros vendedores en darse cuenta del enorme potencial que existe en el mundo del UNIX para la administración de bases de datos. Combinando las ventajas de ser uno de los primeros en lanzar series de productos innovadores, la compañía tiene la línea de productos más versátil y eficiente del mercado de los DBMS en UNIX.

El mismo UNIX proporciona todos los ingredientes necesarios para construir un sistema de archivos planos con una gran cantidad de herramientas. Las

grandes posibilidades de sus numerosas herramientas que utilizan pipes y redirecciones y la potencia de la shell hacen del UNIX la mejor elección para aquel que desarrolla aplicaciones.

Los datos almacenados en UNIX pueden leerse secuencialmente o acceder a ellos directamente. El acceso secuencial requiere leer los primeros catorce registros para leer el decimoquinto. Es evidente que esto hace más lento el procesamiento. Para evitar este problema, UNIX proporciona acceso directo a cualquier byte especificando su posición en el archivo. Todos los paquetes de DBMS hacen uso de esa facilidad en sus implementaciones y proporcionan un amplio abanico de métodos de acceso superiores al acceso secuencial.

### **2.5.2 MS-DOS.**

DOS es la abreviatura de Disk Operating System (Sistema operativo de disco). Es un sistema operativo desarrollado por Microsoft Corporation para las computadoras (ordenadores) personales IBM y compatibles que emplean la familia de procesadores de 16 y 32 bits de Intel. Posee las instrucciones necesarias para manejar ficheros, controlar las interacciones entre el ordenador y el usuario y gestionar periféricos.

Ofrece una serie de servicios ejecutándose en un ambiente de un solo usuario, y un sencillo intérprete de líneas de instrucción como interfaz de su usuario. Su

mayor característica es un grupo de servicios para manejar los archivos (ficheros) y los dispositivos de entrada y salida.

Una de las mayores ventajas del MS-DOS es su popularidad : al haber sido elegido este sistema operativo por muchos fabricantes de ordenadores personales, los programadores de software han respondido con un verdadero diluvio de programas de aplicación. Esta situación implica que puede disponer de los programas que necesita, sin necesidad de realizar modificación alguna para aplicarlos a su sistema concreto.

El MS-DOS tiene cuatro componentes funcionales principales :

- El intérprete de comandos (COMMAND.COM)
- El DOS shell
- El sistema básico de entrada y salida (BIOS)
- Las utilerías del DOS.

### ***El intérprete de comandos***

El intérprete de comandos es el *anfitrión amistoso* del DOS que interactúa con el usuario a través de teclado y pantalla. Al intérprete de comandos también se le conoce como procesador de comandos, y a menudo se le denomina

COMMAND.COM. Este acepta los comandos que el usuario escribe y vigila que éstos se lleven a cabo.

### ***El DOS Shell.***

Es una interfaz visual entre el usuario y el intérprete de comandos. El DOS shell es una alternativa a la posibilidad de escribir los comandos en el indicador de comandos del DOS. Al usarlo puede realizar funciones del DOS sin tener que memorizar todos los comandos disponibles.

En vez de ofrecerle el indicador de comandos, el DOS shell enlista en la pantalla directorios y archivos, lo que ayuda a controlar los programas y datos de los discos. A través de menús y pantallas de ayudas, el DOS shell le permite manipular los archivos y directorios listados para realizar con facilidad las tareas relacionadas con dichos discos y archivos.

### ***El sistema básico de entradas / salidas.***

Los llamados archivos ocultos, o de sistema, son otra parte del sistema operativo. Estos archivos especiales le definen al software cuál es el hardware existente. Al encender la computadora, los archivos del sistema del DOS son cargados en RAM. Combinados, los archivos proveen un conjunto unificado de rutinas que controlan y dirigen las operaciones de la computadora y son conocidos como sistema de entradas/salidas.

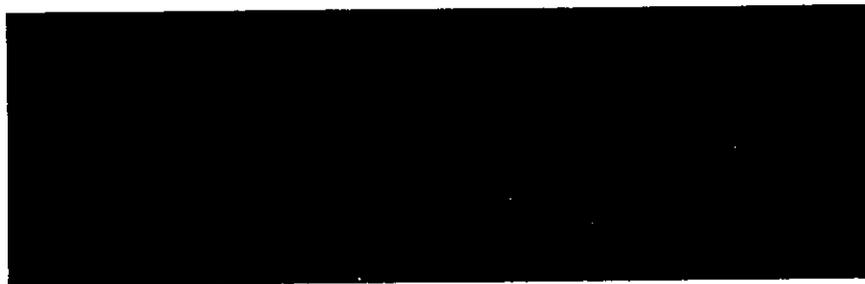
Estos archivos ocultos interactúan con la memoria de sólo lectura o ROM (Read Only Memory), la cual está en la placa principal o de sistema. Esta memoria especial se denomina sistema básico de entrada y salida, o simplemente BIOS, por su siglas en inglés de Basic Input Output System. Como respuesta a los requerimientos de servicio de un programa, los archivos de sistema traducen estos requerimientos y los pasan a ROM BIOS. El BIOS proporciona una traducción posterior de estas peticiones que enlazan éstas con el hardware.

En el trabajo interno de una PC, el DOS es un conjunto de servicios ofrecidos en forma de grupos integrados de instrucciones relacionadas, o rutinas de software. Estas rutinas constituyen el núcleo de los servicios que el DOS proporciona para los programas de aplicación. Por ejemplo el extenso sistema de manejo de archivos proporcionado por el DOS esta formado por estas rutinas. Los programadores acceden a estas rutinas de software para realizar una variedad de operaciones internas con la PC. Por medio del acceso a las rutinas de servicio integradas en el DOS, un programa no necesita del detalle de como funciona el DOS con la PC. Con la interfaz para servicio del DOS ya incorporada, los lenguajes de computación y los programas que se producen con ellos pueden emplear el DOS de manera uniforme.

### ***Las utilerías del DOS.***

Las utilerías del DOS realizan provechosas tareas de rutina, tales como preparar discos, comparar archivos, encontrar el espacio libre de un disco e

imprimir en segundo plano. Algunas utilerías proporcionan estadísticas sobre tamaño del disco y memoria disponible y comparan discos y archivos. Los programas de utilerías son programas que residen en el disco y que son cargados en memoria por el COMMAND.COM.



## **CAPITULO 3. DESARROLLO DEL SISTEMA ESCOLAR DE LA DIMEI DE LA FACULTAD DE INGENIERIA DE LA UNAM.**

### **3.1 SITUACIÓN ACTUAL.**

#### **3.1.1 FACULTAD DE INGENIERÍA.**

Con más de 200 años de existencia, la Facultad de Ingeniería (F.I.) conserva hoy su papel de institución universitaria dinámica, líder en la formación de ingenieros, maestros, doctores y especialistas que demanda el desarrollo del país.

Con el propósito de mejorar su participación en la construcción de un país más competitivo y acorde con las nuevas realidades internacionales, la Facultad de Ingeniería ha reestructurado sus planes de estudio, ha reorganizado sus cuadros docentes y de investigación y ha mejorado sus instalaciones físicas con la finalidad de optimizar su nivel académico. Así mismo, en los últimos años ha desarrollado nuevas formas de vinculación con el sector productivo del país.

La Facultad de Ingeniería cuenta con cuatro divisiones :

- División de Ingeniería civil, topográfica y geodésica
- División de Ingeniería de ciencias de la tierra
- División de Ingeniería mecánica e industrial (DIMEI)
- División de Ingeniería eléctrica, electrónica y en computación.

En las cuales se imparte educación en once licenciaturas : Ingeniero Civil, en Computación, Geofísico, Industrial, Geólogo, Eléctrico y Electrónico, de Minas y Metalurgista, Petrolero, Mecánico, en Telecomunicaciones, Topógrafo y Geodesta.

La Facultad de Ingeniería cuenta también con una División de Estudios de Posgrado que ofrece especialidades, maestrías y doctorados. Además cuenta con una División de Educación Continua, misma que se ha consolidado desde 1971 con intensas actividades de actualización profesional. (Como se muestra en la Figura 3.1).

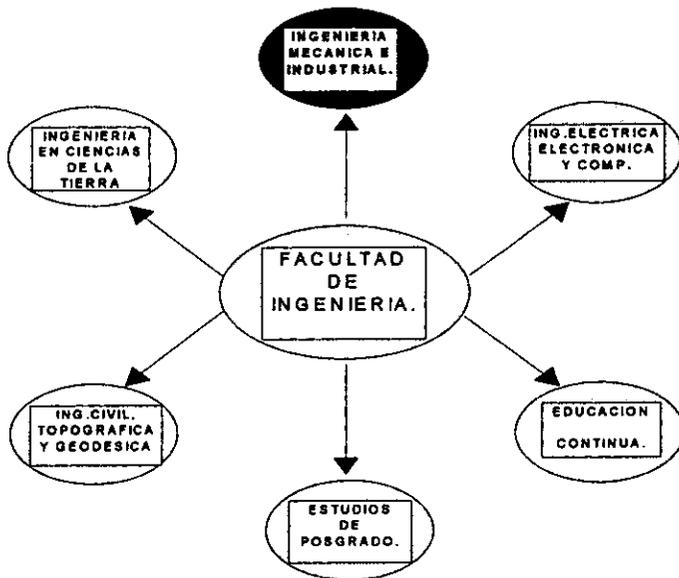


Figura 3.1 Divisiones de la Facultad de Ingeniería.

**a) Profesores.**

En la facultad de Ingeniería dan clases 250 profesores de carrera y los profesionistas más distinguidos de las ingenierías como profesores de asignatura. Los de carrera imparten cursos, materias de su especialidad, y participan en proyectos de investigación y desarrollo tecnológico. Los de asignaturas se desarrollan profesionalmente en empresas públicas o privadas e imparten las materias de su especialidad; en ambos casos dirigen tesis profesionales.

Los profesores de carrera participan en programas de investigación aplicada en ingeniería en los que colaboran con estudiantes. Estos profesores publican los resultados de sus investigaciones en forma regular, elaboran notas y textos para sus cursos y tienen estudios de posgrado. El 8 % de los profesores poseen doctorado, han participado en investigaciones de su especialidad con investigadores de primer nivel en el mundo, son invitados frecuentemente a impartir cátedra en otras universidades del país y del extranjero, y son autores de una producción científica de la más alta calidad.

**b) Alumnos.**

La Facultad de Ingeniería es la escuela más importante en la formación de ingenieros del país. Posee la matrícula más elevada con más de 10,000 estudiantes de licenciatura. En el posgrado cuenta con 550 estudiantes, que representan la mitad del total nacional.

En todas las licenciaturas de la Facultad, el alumno participa en proyectos en conjunto con una empresa o una firma de ingeniería. El alumno aprende así a desenvolverse como miembro de un equipo plural y a evaluar su desempeño profesional. De este modo, los alumnos participan en la labor de extensión universitaria.

***c) Instalaciones.***

La F. I. cuenta con las más amplias y mejor equipadas instalaciones dentro de su campo en el país, mismas que están conformadas por 18 edificios en el campus de Ciudad Universitaria ; un edificio de la División de Posgrado en Juitepec, Morelos ; además de los del Palacio de Minería y el Real Seminario de Minería, estos últimos en el centro histórico de la Ciudad de México.

Los estudiantes de la Facultad tienen acceso a más de 1200 computadoras de diversas capacidades, que van desde computadoras personales, workstation, hasta computadoras que les permiten conectarse con la supercomputadora CRAY más grande del país. Los laboratorios especializados como los de Ingeniería Ambiental, Petrolera, Geología, Ingeniería Eléctrica y Electrónica, Manufactura Avanzada, Mecánica de Fluidos, Materiales, Termoenergía, etc., se encuentran entre los mejor equipados tecnológicamente de México.

***d) Egresados.***

En la actualidad en la F. I. se titulan anualmente alrededor de 690 ingenieros de las diversas áreas. En los últimos años ésta cifra ha ido creciendo y llega ya a 800 ingenieros, 35 especialistas, 70 maestros en ingeniería y de cuatro a siete doctores más cada año.

De los egresados de licenciatura, 97 % obtienen empleo en el sector industrial afín a sus estudios al obtener el título, y los demás inician estudios de posgrado en México o en el extranjero.

La Sociedad de Ex-alumnos de la Facultad de Ingeniería (SEFI) constituye la asociación de su tipo más activa y numerosa en nuestro país. Actualmente agrupa a más de 7,000 ex-alumnos y se plantea entre sus principales objetivos proporcionar a la facultad apoyo moral y material para propiciar el que siga siendo la mejor escuela de ingeniería de México.

**3.1.2 INGENIERÍA INDUSTRIAL.**

Es la rama de la ingeniería que desarrolla sistemas de producción, los diseña, analiza, innova, dirige y controla.

Estos sistemas de producción pueden ser de un bien o un servicio, y así la Ingeniería Industrial se coloca entre las ramas científicas y tecnológicas más interdisciplinarias.

La Ingeniería Industrial aporta a las empresas el estado del arte en procedimientos basados en matemáticas aplicadas y principios generales de la física para contribuir a su éxito en la competitividad mundial.

Dentro de la Facultad de Ingeniería de la UNAM, el Departamento de Ingeniería Industrial se inserta como parte de la División de Ingeniería Mecánica e Industrial (DIMEI).

El departamento se especializa en enseñar, aplicar e investigar las ciencias de la ingeniería industrial.

Está organizado de forma matricial y con un jefe de departamento; el grupo de profesores de carrera que conforma el departamento se aboca a la Investigación de Operaciones, a la Ingeniería Financiera, a la Producción y a la Calidad ; se complementan con profesores de asignatura que enfocan su enseñanza en la ingeniería industrial aplicada.

### ***Funciones y Objetivo:***

La misión del Departamento de Ingeniería Industrial es contribuir a la formación de ingenieros industriales en su parte medular, es decir, en

ingeniería aplicada y en ciencias de la ingeniería industrial; también el departamento coadyuvará en la enseñanza de temas que los estudiantes de ingeniería de otras áreas necesitan para su formación.

El Objetivo del Departamento de Ingeniería Industrial es crear las condiciones necesarias y suficientes para dar servicio a los alumnos, maestros, industria e instituciones vinculadas a la Ingeniería Industrial.

Las funciones del departamento se dividen en administrativas y académicas; las primeras abarcan contratación de personal, gestión y práctica de visitas, control de costos de los proyectos y del propio departamento, programación de horarios de clase y exámenes. Las segundas contemplan la elaboración planes de estudio; elaboración, actualización y revisión de temarios de los cursos que se imparten, y asesorías tanto de apoyo a las asignaturas como para la realización de trabajos de tesis.

Otra de las funciones del departamento es la relación escuela - industria, que se dedica a vincular con la industria, a profesores y alumnos, y se divide en:

- 1.-Realización de proyectos académicos
- 2.-Realización de proyectos externos
- 3.-Realización de prácticas y visitas
- 4.-Realización de pruebas de calidad a productos terminados.

### **3.1.3 MANEJO DE LA INFORMACIÓN EN LA DIMEI.**

Actualmente la División de Ingeniería Mecánica e Industrial depende de la información que le proporciona la F.I. referente a todo lo relacionado con el alumnado que la conforma, es decir, en la F.I. se llevan a cabo sus trámites escolares como son inscripción a semestres, cambios de horarios, inscripción a exámenes extraordinarios, y tanto la captura, modificaciones y realización de actas del alumnado. A su vez la DIMEI maneja la información correspondiente a horarios, profesorado, salones, planes de estudio, bibliografía y temarios en forma manual .

Debido a esto, la DIMEI no tiene un control exacto de dicha información lo que conlleva a :

- ♦ Pérdida de tiempo en la búsqueda de información.
  
- ♦ Dificultad para adquirir información actualizada del alumnado ; ya que toda la información se encuentra almacenada en la base de datos de la F.I.
  
- ♦ Información (calificaciones, actas) inconsistentes al no recibir en forma rápida las actualizaciones de dicha base de datos.

- ♦ El alumnado no cuenta con una eficiente, rápida y sencilla manera de conocer calificaciones, profesorado de la división, horarios, temarios y bibliografía de un semestre determinado.
  
- ♦ El profesorado pierde tiempo de clase al proporcionar al alumnado el temario y la bibliografía correspondiente al curso.

### **3.2 RECURSOS DISPONIBLES.**

En la División de Ingeniería Mecánica y Eléctrica (DIMEI) se cuenta con tres equipos de computo personal disponibles para la implantación del sistema con las siguientes características :

- HP Vectra/66
- Sistema Operativo Windows 95
- Procesador 486 a 66 Mhz.
- 8 megabytes en RAM
- 258 megabytes en disco duro
- Teclado en español de 101 teclas
- Drive de 3½
- Cuenta con puertos para red y multimedia.

Estos equipos están conectados a una impresora H.P . Lasser Jet Serie II, blanco y negro con bandeja de entrada para hojas tamaño oficio y tamaño carta.

En lo que se refiere a recursos humanos, la DIMEI tiene planeado establecer programas de servicio social en los cuales se capacite a los alumnos de la Facultad para manejar y dar mantenimiento al sistema implementado ya que con esto se minimizarían los costos en el proyecto.

### **3.3 ANÁLISIS.**

La etapa principal de todo sistema de información es el análisis, el cuál tiene como tarea específica desglosar el problema de tal forma que el analista pueda tener una visión clara de los requerimientos del sistema y así poder plantear las soluciones viéndolas reflejadas en un diseño y por consiguiente un desarrollo óptimo del sistema.

Para la realización del estudio de la DIMEI se utilizará la metodología de análisis estructurado, por que existe una generalidad de su uso en varias instituciones, lo cual asegura que pueda ser de fácil mantenimiento en el momento de llevar a cabo una modificación del sistema, ya que existe una documentación formal del estudio realizado. En esa documentación se encuentran representaciones esquemáticas que ayudan a una mayor

comprensión del sistema tanto para los analistas como para el usuario u otras personas que se vean involucradas en el proyecto.

La metodología del análisis estructurado, tiene sus orígenes en los trabajos realizados por Tom De Marco en 1978, Gane y Sarson en 1977 y Weinberg en 1978, estos autores comenzaron sus trabajos en lo que ahora se conoce como el análisis estructurado clásico, que aún cuando a principios de los 80's se vieron muchas deficiencias en dichos trabajos, fueron de gran utilidad en su momento y sirvieron de base para dar cabida a lo que hoy se conoce como el Análisis Estructurado Moderno propuesto por Edward Yourdon.

Dentro del desarrollo de sistemas de información, el uso del análisis estructurado es de gran ayuda para comprender sistemas grandes y complejos.

De esta manera se asegura la captura de todos los detalles necesarios relacionados con los datos y procesos involucrados en el sistema (manual o automatizado) que se maneja actualmente en la institución. El objetivo del análisis estructurado es organizar las tareas asociadas con la determinación de requerimientos para obtener la comprensión completa y exacta de una situación dada. A partir de aquí se determinan los requerimientos que serán la base de un sistema nuevo o modificable.

El objetivo de la metodología de análisis estructurado es representar un sistema en un modelo esencial.

El modelo esencial del sistema es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario, diciendo lo mismo posible (de preferencia nada) acerca de como se implantará. El modelo esencial consiste en dos componentes principales :

- ◆ Modelo ambiental
- ◆ Modelo de comportamiento

### ***Modelo ambiental.***

El Modelo ambiental define la frontera entre el sistema y el resto del mundo (es decir, el ambiente en el cual se desenvuelve el sistema). Consiste en un diagrama de contexto, una lista de acontecimientos y una descripción breve del sistema.

La labor más difícil para el analista en la especificación de un sistema es determinar que es parte del sistema y qué no. Además de determinar qué esta en el interior del sistema y qué en el exterior (lo que se logra definiendo la frontera entre el sistema y el ambiente), también es importante definir las interfases entre ellos. Se necesita saber qué información entra al sistema desde el ambiente exterior, y qué información produce como salida al ambiente externo.

Desde luego, las entradas y salidas no se producen al azar ; ningún sistema de información toma todos los datos disponibles en el universo, ni expulsa cosas

al azar al ambiente exterior. Los sistemas tienen un propósito; específicamente, producen salidas como respuesta a algún acontecimiento, o estímulo, en el ambiente. Así, otro aspecto crítico del modelo ambiental consiste en identificar los acontecimientos que ocurren en el ambiente al cual debe responder el sistema (Figura 3.2).

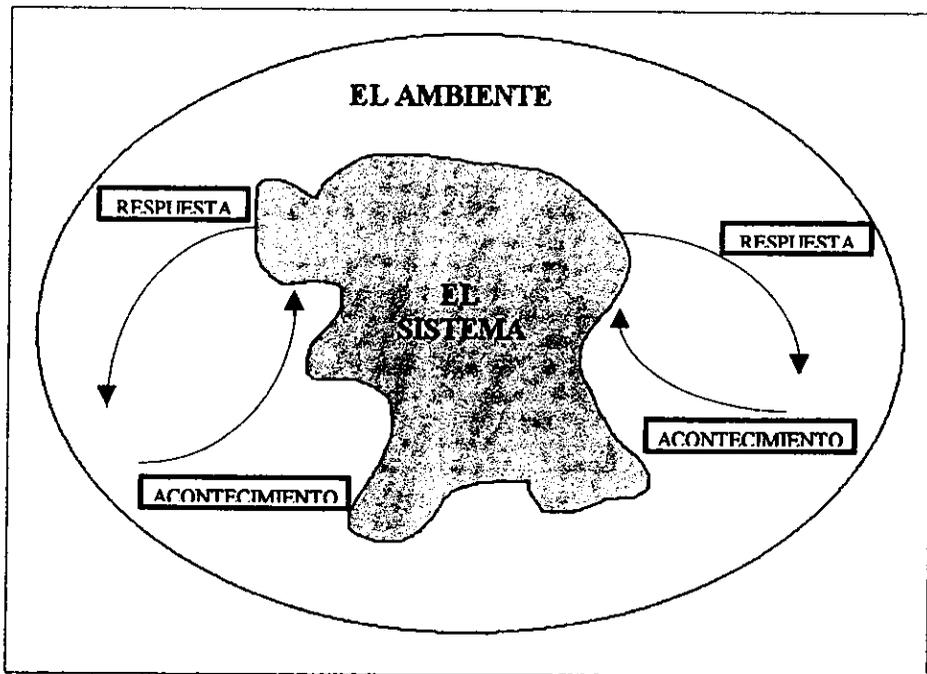


Figura 3.2 Frontera entre el sistema y el ambiente.

El modelo ambiental consta de tres componentes :

- ♦ Definición del objetivo o propósito.
  
- ♦ Diagrama de contexto.
  
- ♦ Lista de eventos o acontecimientos.

*Definición del objetivo.*

Es una declaración textual breve y concisa del propósito del sistema, dirigida a niveles administrativos superiores, la administración de los usuarios, y otros que no están involucrados directamente con el desarrollo del sistema.

Es recomendable que el objetivo, conste de una o varias frases ; sin embargo jamás debe llegar a más de un párrafo, ya que la intención no es de proporcionar una descripción completa y detallada del sistema. Tal esfuerzo iría en contra del objetivo : el propósito del resto del modelo ambiental y del modelo de comportamiento es dar todos los detalles.

*Diagrama de contexto.*

El diagrama de contexto es una visión general del sistema, en la cual se busca definir las fronteras del mismo y las entidades externas con las cuales

interactúa, así como los acontecimientos o estímulos del ambiente que provocan una respuesta del sistema.

El diagrama de contexto enfatiza las siguientes características :

- Los sistemas, personas y organizaciones con los cuales se comunica el sistema. Los cuales son definidos como entidades.
- Los datos que el sistema recibe del medio ambiente y que deben procesarse de alguna forma.
- Los datos que el sistema produce y que se envían al mundo exterior.
- Los archivos que el sistema comparte con otros sistemas o entidades.
- La frontera entre el sistema y el resto del mundo.

#### *Lista de eventos.*

Es una lista narrativa de los “estímulos” que ocurren en el medio ambiente a los cuales el sistema debe responder, por ejemplo : un alumno solicita el horario de clases de un profesor o la calificación de sus asignaturas.

En conclusión la elaboración de un modelo ambiental es lo primero y lo más importante en la construcción de un modelo completo de los requerimientos del

usuario para un nuevo sistema. No se trata de un proceso sencillo ya que en proyectos de gran tamaño, involucran muchas entidades externas y pueden tener más de cien flujos de datos de entrada y salida. La lista de eventos constituye un gran esfuerzo pues resulta necesario identificar una gran cantidad de acontecimientos que el sistema debe de controlar. De aquí la importancia del nivel de abstracción del analista y la manera en que conceptualiza el sistema ; además, puede ser difícil encontrar una declaración sencilla del por qué debe existir el sistema.

Una vez construido el modelo ambiental debe ser revisado cuidadosamente por todos los representantes de los usuarios, además del equipo del proyecto. Al ser aprobado se estará preparado para la construcción del modelo de comportamiento.

### ***Modelo de comportamiento.***

El modelo de comportamiento describe las funciones a desarrollar por el sistema para que interactue de manera exitosa con el ambiente. Esta constituido por los diagramas de flujo, diccionario de datos y diagramas de Entidad - Relación.

Este modelo esta formado por el conjunto de diagramas de flujo (DFD's), que describen cada una de las funciones del sistema en todos sus niveles. Su objetivo es definir la forma en como el sistema se comporta internamente para producir sus entradas en salidas.

*Diagrama de flujo de datos (DFD).*

Son herramientas gráficas que sirven para describir y analizar el movimiento de los datos a través de un sistema.

Los DFD se dividen en dos partes :

a) Diagramas físicos de flujo de datos : Son aquellos que proporcionan un panorama del sistema en uso, que es dependiente de la implantación, muestra las tareas que se lleven a cabo y cómo. Las características físicas incluyen :

- ◆ Nombres de personas.
- ◆ Nombres o números de formatos y documentos.
- ◆ Nombres de departamentos.
- ◆ Equipo y dispositivo utilizado.
- ◆ Ubicaciones.
- ◆ Nombre de procedimientos.

b) Diagramas lógicos de flujo de datos : Proporcionan un panorama del sistema independiente de la implantación que se centra en el flujo de datos entre los procesos sin considerar los dispositivos específicos y la localización de almacenes de datos o personas del sistema.

La simbología utilizada para los diagramas de flujo de datos se presenta en la Figura 3.3.

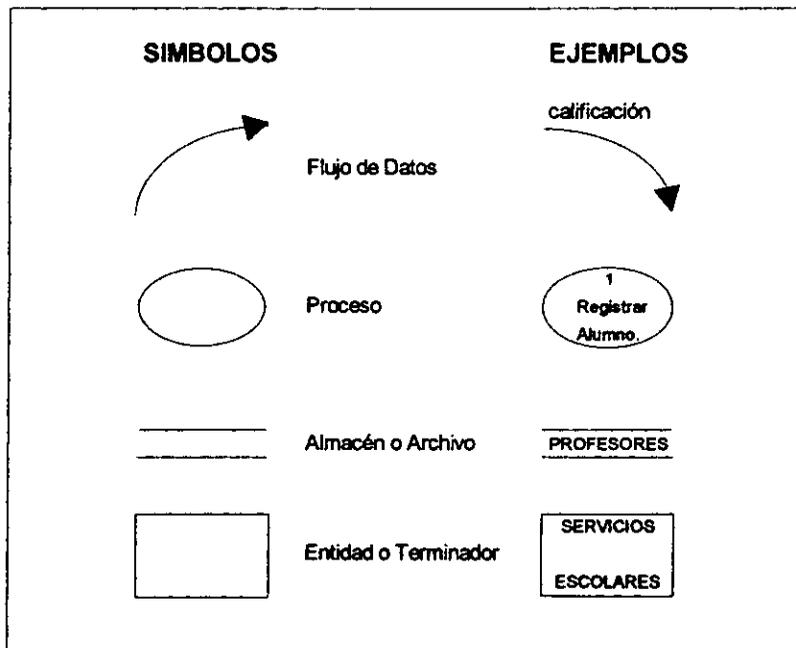


Figura 3.3 Simbología de Diagramas de Flujo de Datos.

### *Diccionario de Datos.*

Es un listado organizado de todos los datos pertenecientes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista

tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios.

Existen muchos esquemas de notación comunes utilizados para el analista de sistemas. El que se muestra a continuación es de los más comunes y sencillos por lo que optamos por emplearlo en el desarrollo de nuestro análisis ; utiliza los siguientes símbolos para su notación :

<b>Símbolo</b>	<b>Significado</b>
=	está compuesto de
+	y
( )	optativo (puede estar presente o ausente)
{ }	iteración
[ ]	seleccionar una de varias alternativas
**	comentario
●	identificador (campo clave) para un almacén
	separa opciones alternativas en la construcción

### *Diagrama de Entidad - Relación.*

El diagrama de Entidad - Relación (también conocido como DER, o diagrama E - R) es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema.

Los diagramas de Entidad - Relación son una representación conceptual de objetos del mundo real y las relaciones entre ellos ; definen la información que

el sistema debe crear, mantener, procesar y eliminar, así como las relaciones que deben ser soportadas por la base de datos ; agrupa los elementos de información en entidades y refleja las definiciones de los archivos de datos del sistema.

Componentes de un diagrama Entidad - Relación :

1. Entidades : se representan con un rectángulo, poseen un nombre que los identifica de los demás y tienen uno o más atributos característicos de él mismo que hacen que tenga un papel necesario en el sistema.
2. Relaciones : las entidades se conectan entre sí mediante relaciones y se representa por medio de un rombo y un nombre que define la forma de la relación.
3. Cardinalidad : indica la forma en como se da la relación entre entidades y estas pueden ser uno a uno (1-1), uno a muchos (1-M) o muchos a muchos (M-M).

### **3.3.1 MODELO AMBIENTAL DE LA DIMEI.**

#### **Objetivo del Sistema.**

Automatizar los procesos tales como : integración de la información del alumnado de la División de Ingeniería Mecánica e Industrial en una base de datos, consultas de horarios, calificaciones, planes de estudio, temarios, bibliografía y curriculums de profesores que permita contar con información actualizada de una manera rápida y eficiente.

Responsabilidades del sistema :

- Control de acceso a la información.
- Captura de datos del alumnado y profesorado de la DIMEI.
- Actualización de la base de datos.
- Realización de consultas y reportes.

No es responsabilidad del sistema :

- ◆ Actualizar la base de datos de la F. I.

- ♦ Manejar información de profesores y alumnos externos de la DIMEI.
  
- ♦ Imprimir papelería oficial (actas, tira de materias).

**Lista de eventos.**

1. La Facultad de Ingeniería proporciona los datos de los alumnos (de la DIMEI).
2. La Facultad de Ingeniería proporciona datos de profesores.
3. La Facultad de Ingeniería entrega datos académicos de los alumnos.
4. Los profesores requieren los datos académicos de los alumnos.
5. Los profesores requieren informe de nuevas adquisiciones.
6. Los profesores requieren información de proyectos de titulación..
7. Los profesores requieren temarios de asignaturas.
8. Los profesores envían curriculumns actualizados.
9. Los profesores entregan calificaciones de los alumnos.
10. Los profesores proporcionan informes de asesorías.
11. Los alumnos solicitan curriculumns de profesores.
12. Los alumnos solicitan informes de proyectos de tesis.
13. Los alumnos solicitan calificaciones de asignaturas.
14. Los alumnos solicitan informes de asesorías.
15. Los alumnos solicitan temarios de asignaturas.
16. Los alumnos solicitan horarios de los profesores.

17. Los alumnos solicitan bibliografía.
18. Los alumnos solicitan estadísticas de aprovechamiento de grupo.
19. Los alumnos proporcionan sus datos personales.
20. La coordinación proporciona listados de horarios.
21. La coordinación proporciona información de temarios.
22. La coordinación proporciona información de proyectos de titulación.
23. La coordinación propone seminarios de titulación.
24. La coordinación solicita información de asesorías.
25. La coordinación solicita calificaciones de alumnos.
26. La biblioteca envía catálogo de libros.
27. La biblioteca envía listado de nuevas adquisiciones.
28. La biblioteca envía listado de tesis.
29. Los egresados proporcionan sus datos personales.
30. Los egresados proporcionan su tema de tesis.
31. Los egresados solicitan información de proyectos de titulación.
32. Los egresados solicitan información de asesorías.
33. Los egresados solicitan información de seminarios de titulación.

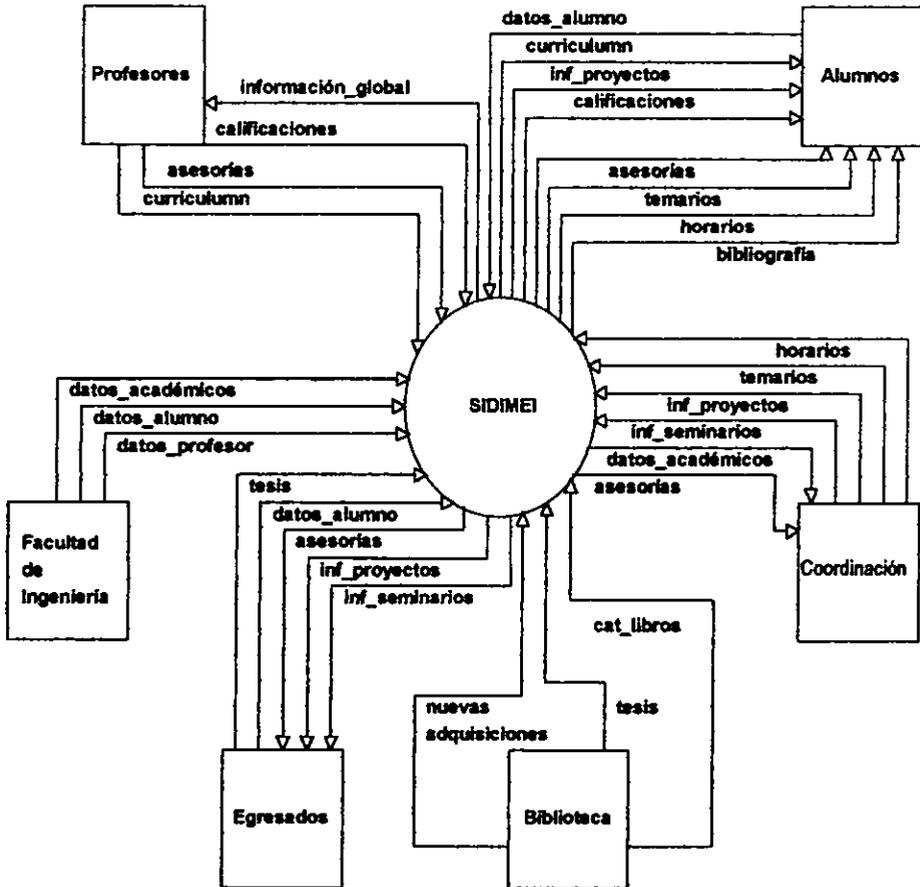
**ESTAS TESIS NO DEBE  
SALIR DE LA BIBLIOTECA**

### **Diagrama de contexto.**

Se muestra de una forma esquemática y general todo el sistema desglosando sus componentes como lo son :

- Las entidades : ALUMNOS, PROFESORES, COORDINACIÓN, BIBLIOTECA, FACULTAD DE INGENIERÍA Y EGRESADOS.
  
- Los datos que el sistema recibe del mundo exterior y que debe procesar de alguna forma.
  
- Los datos que el sistema produce y que se envían al exterior.

## DIAGRAMA DE CONTEXTO



## **Diagrama de flujo de datos (DFD's).**

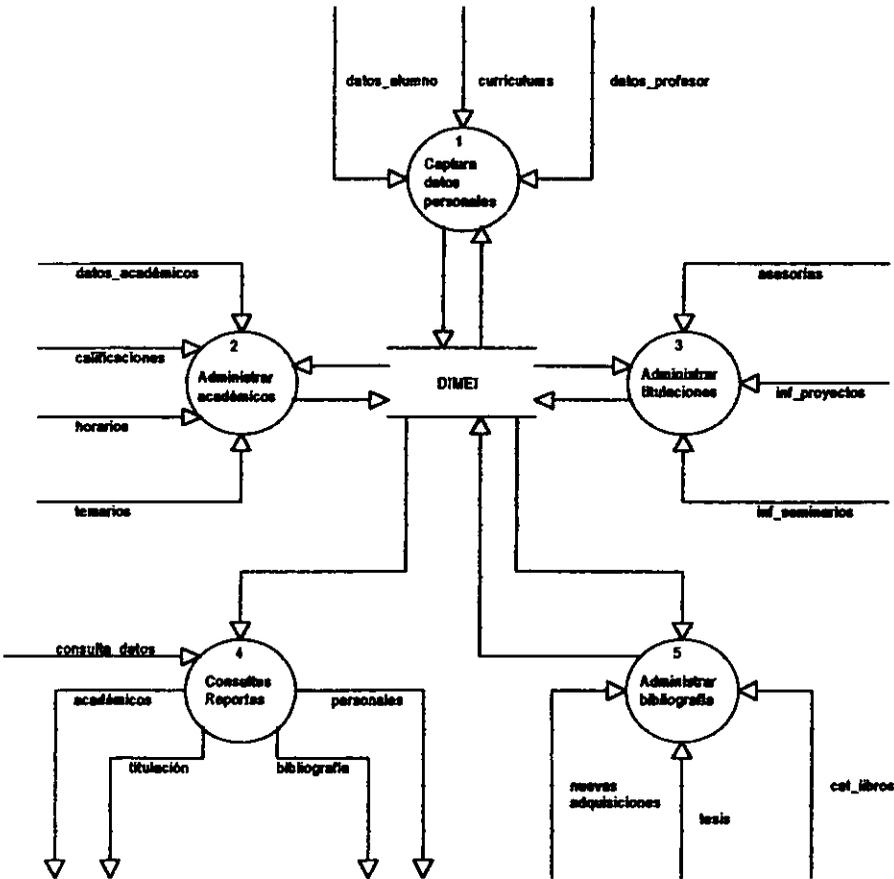
En el primer diagrama de flujo de datos, el D.F.D. 0 representa la vista de más alto nivel de las principales funciones del sistema, al igual que las principales interfases, es por ello que cada burbuja se enumera y se nombra para tener una referencia única y se observe como desciende a niveles de detalle. En el caso del Sistema de Información de la DIMEI (SIDIMEI), se representa el flujo de información que recibe y otorga cada proceso, así como el acceso que es necesario realizar a una base de datos para poder contar con la información que se requiere.

### **D.F.D. 1**

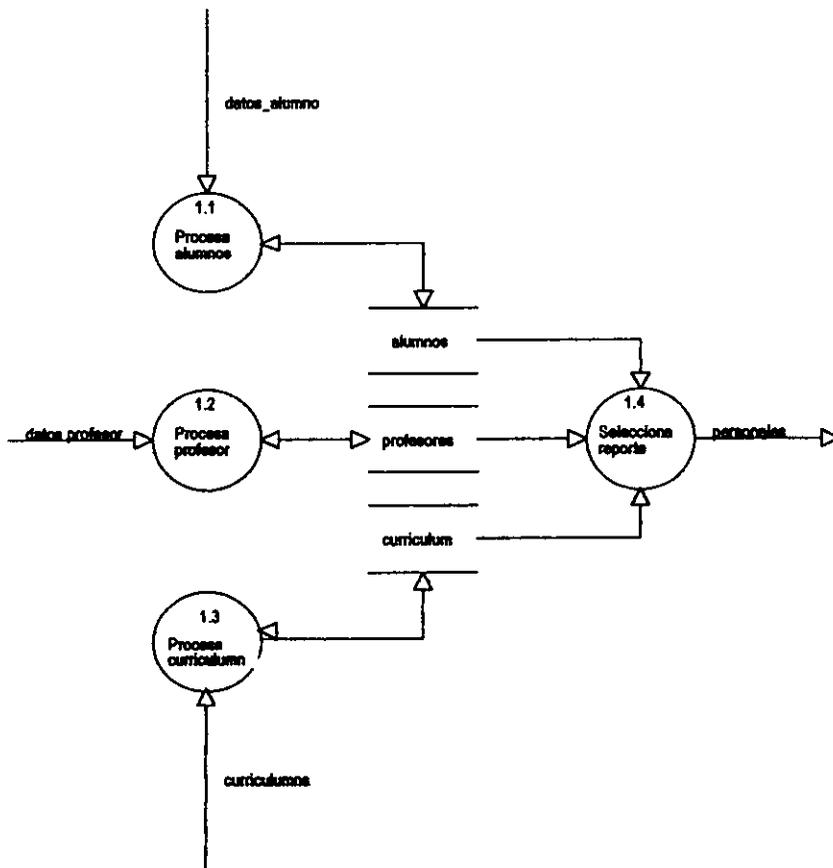
#### **Captura datos personales.**

Es un procedimiento en el cual se tiene como entrada los datos de los profesores y alumnos de la DIMEI que serán proporcionados por la Facultad de Ingeniería. Esta información quedará almacenada dentro de la nueva base de datos anexando la de los alumnos de nuevo ingreso y la de los egresados de la división. Una vez integrada la información de los profesores se procede a la captura de sus respectivos curriculumns los cuáles se estarán actualizando cada vez que el profesor lo envíe. También se tiene la opción de sacar impresos de la información registrada.

### D.F.D. 0



### D.F.D. 1

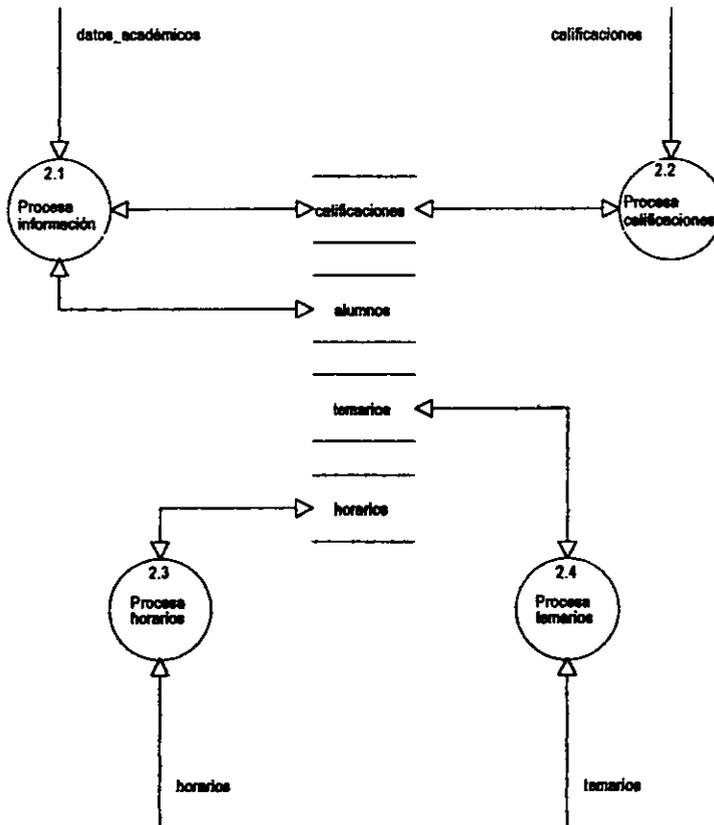


**D.F.D. 2**

**Administra datos académicos.**

En este proceso se lleva a cabo la captura y actualización de la información referente a calificaciones de alumnos que proporciona inicialmente la Facultad de Ingeniería. Una vez instalado el sistema este proceso se llevará a cabo a través de la información que presenten los profesores en la coordinación. Los horarios y temarios quedarán a sujetos a las disposiciones que determine la coordinación de acuerdo al semestre en curso.

### D.F.D. 2

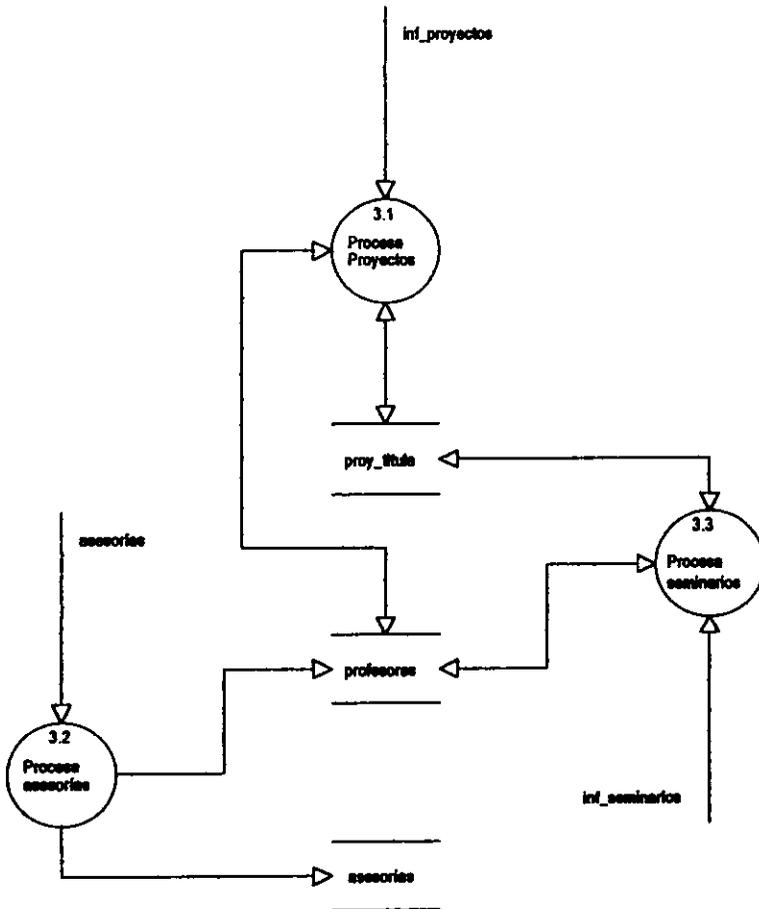


### **D.F.D. 3**

#### **Administra titulaciones.**

Se realiza captura y actualización de información relacionada con los proyectos y seminarios de titulación que se encuentran disponibles para los alumnos de la división. Las asesorías se llevarán a cabo de acuerdo al plan de estudios o proyectos de tesis, estas serán impartidas por los profesores cuya especialización este acorde al tema o materia de que se trate.

### D.F.D. 3

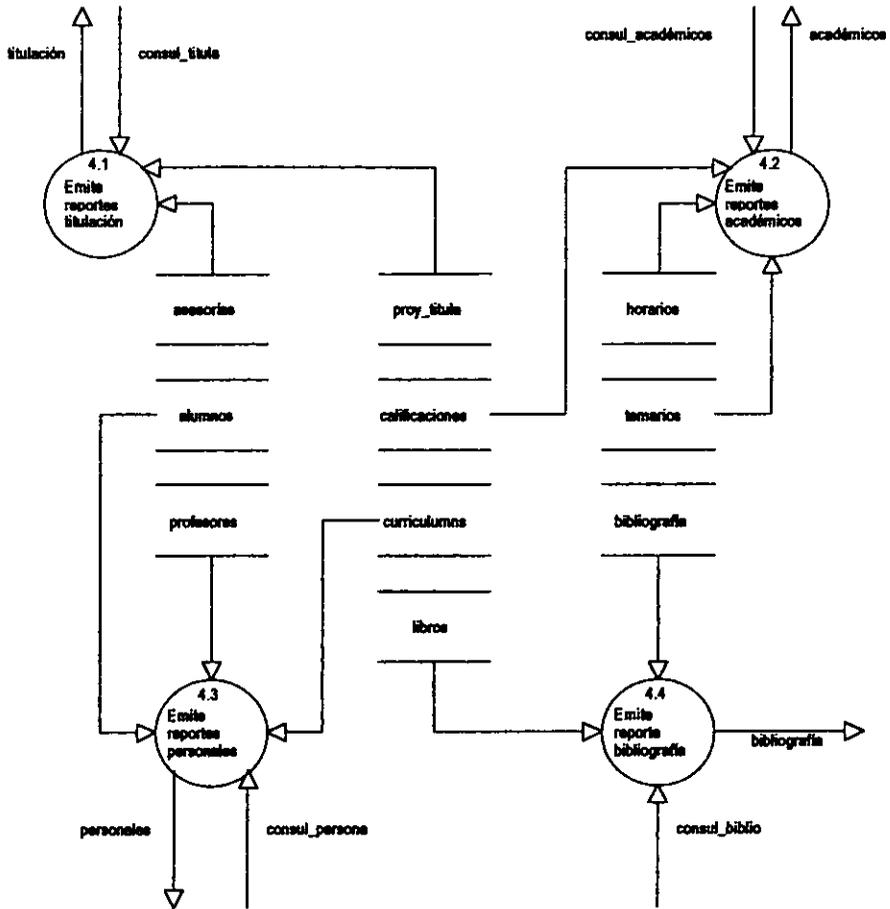


#### **D.F.D. 4**

##### **Consultas - Reportes.**

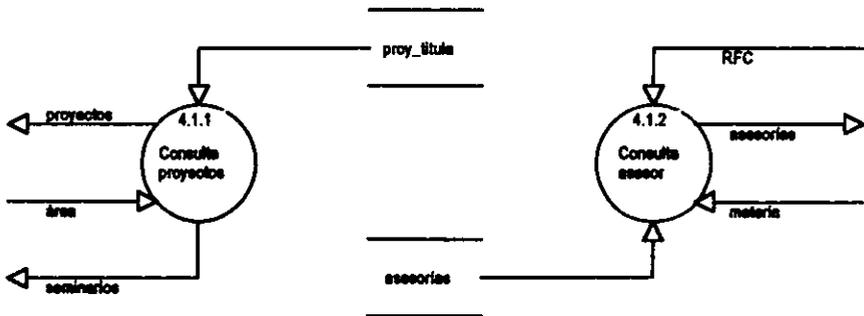
Para cada uno de los procesos del sistema se tiene la opción de consultar en pantalla la información previamente registrada y alternativamente obtener una impresión de la misma. Los alumnos, profesores y egresados podrán consultar calificaciones, horarios, temarios, proyectos de titulación, asesorías, bibliografía y curriculumns de profesores cuando lo requieran. Así mismo la coordinación contará con una confiable información para los tramites escolares necesarios.

D.F.D. 4



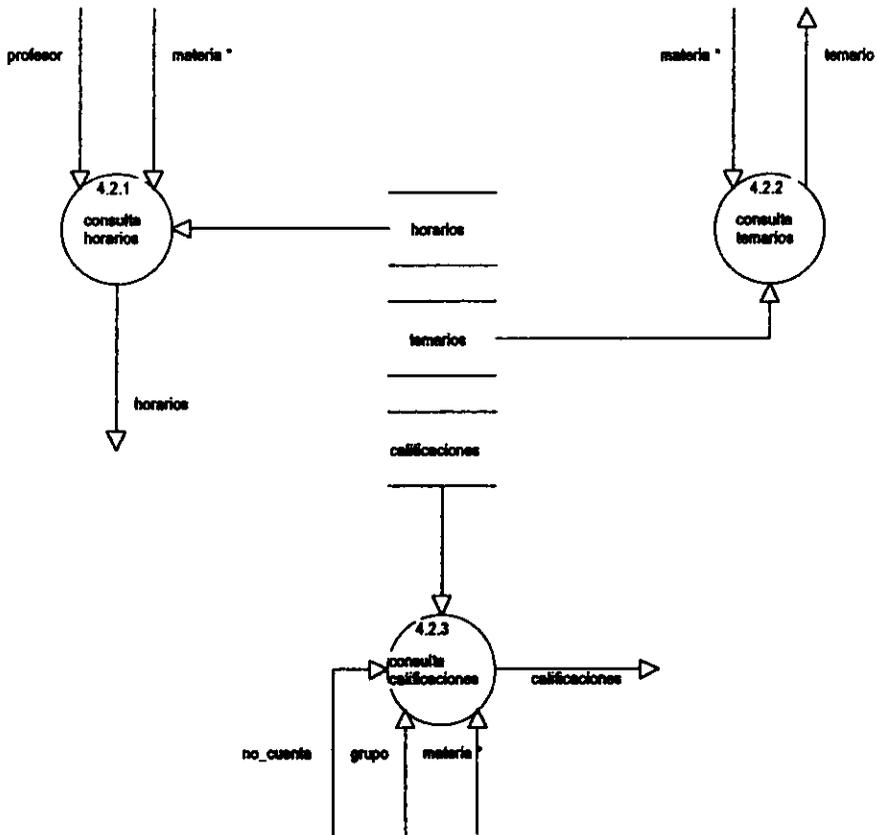
### D.F.D. 4.1

Emite reportes titulación.



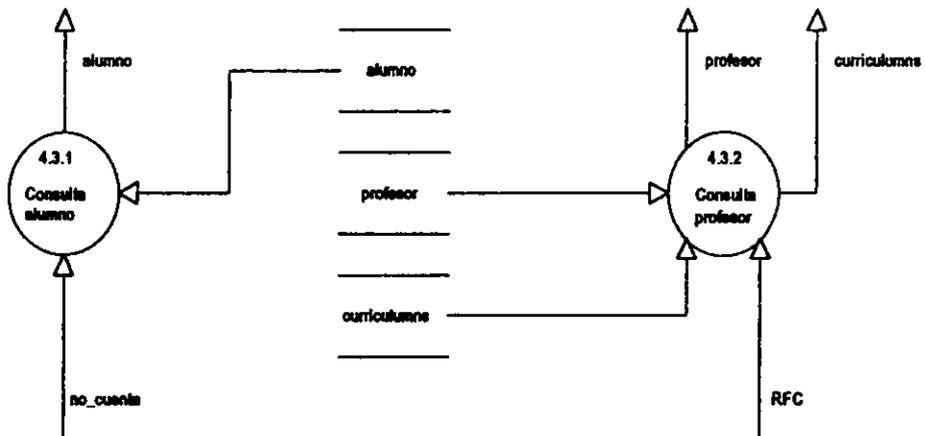
### D.F.D. 4.2

#### Emite reportes académicos.



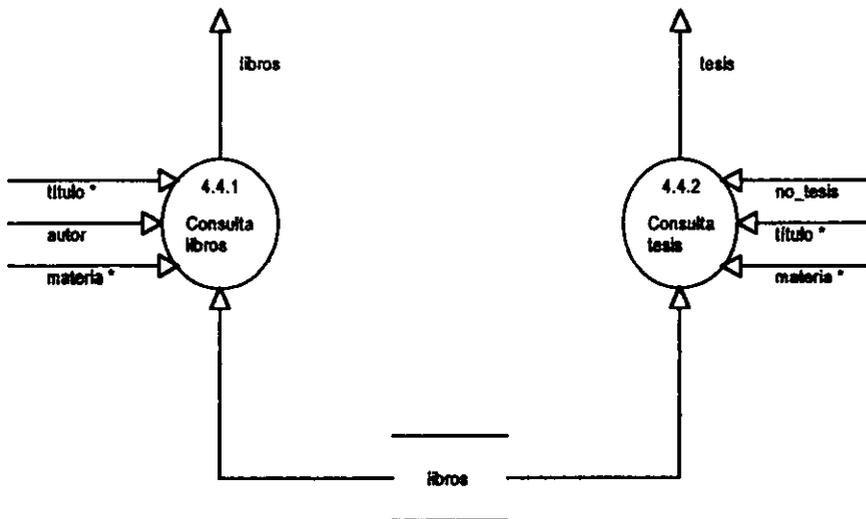
### D.F.D. 4.3

**Emite reportes personales.**



### D.F.D. 4.4

**Emite reportes personales.**

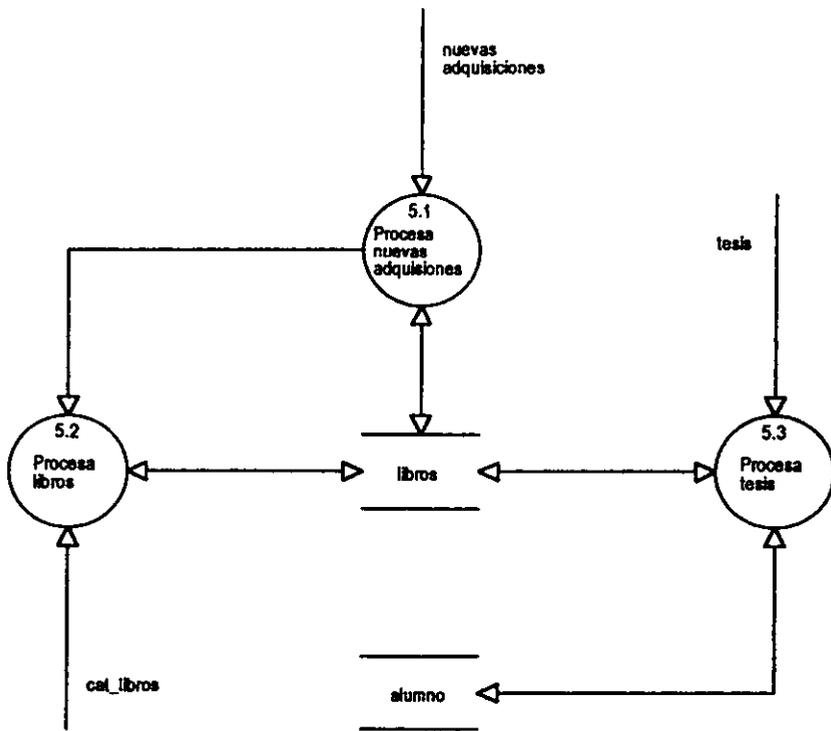


## **D.F.D. 5**

### **Administrar bibliografía.**

Este proceso proporciona información de los diferentes títulos que se encuentran disponibles para las asignaturas de las materias que se imparten dentro de la división. Esta información es actualizada por la biblioteca cada vez que se realice una nueva adquisición. Por otra parte los egresados de la DIMEI donan sus trabajos de tesis a la biblioteca con el fin de que las nuevas generaciones cuenten con una mayor información sobre los temas de actualidad y que conllevarán al buen desarrollo de su aprendizaje.

### D.F.D. 5



## Diccionario de datos.

<b>académicos</b>	= [ datos_académicos   horarios   temarios ]
<b>alumno</b>	= datos_alumno
<b>antigüedad</b>	= * tiempo que se tiene laborando como profesor de carrera dentro de la universidad *
<b>año</b>	= * año de registro de la tesis   año de la publicación *
<b>área</b>	= * agrupación de las materias de acuerdo a su especialización *
<b>asesorías</b>	= nom_profesor + área + [ materia   nom_tesis ] + hora_asesoría
<b>autor</b>	= [ autor_libro   autor_tesis ]
<b>autor_libro</b>	= 1 { nombre } N
<b>autor_tesis</b>	= autor_libro
<b>bibliografía</b>	= [ libros   tesis ]
<b>calificaciones</b>	= 1 { ● no_cuenta + cve_carrera + 1 { cve_materia + materia + calificación } N } N
<b>categoría</b>	= * grado que obtiene el profesor en base a su antigüedad y experiencia *
<b>cat_libros</b>	= 1 { ● no_libro + nom_libro + autor_libro + materia + área + año + editorial + edición +

	lugar_imp + ubicación } N
<b>consulta_datos</b>	= [ consul_titula   consul_académicos   consul_persona   consul_biblio ]
<b>consul_académicos</b>	= [ materia   no_cuenta   grupo ]
<b>consul_persona</b>	= [ título   autor 1 materia 1 no_tesis ]
<b>consul_persona</b>	= [ RFC   no_cuenta ]
<b>consul_titula</b>	= [ área   materia   RFC ]
<b>curriculumns</b>	= * curriculum vitae *
<b>cve_carrera</b>	= * clave de la carrera del alumno *
<b>cve_materia</b>	= * clave de la materia cursada *
<b>datos_académicos</b>	= 1 { ● no_cuenta + cve_carrera + status + semestre + 1 { cve_materia + materia + calificación } N + promedio } N
<b>datos_alumno</b>	= 1 { ● no_cuenta + nom_alumno + dirección + colonia + ciudad + cp + nacionalidad + teléfono + sexo + edad } N
<b>datos_profesor</b>	= 1 { ● RFC + nom_profesor + especialidad + antigüedad + categoría + grado + dirección + colonia + ciudad + cp + nacionalidad + teléfono + sexo + edad + edo_civil } N
<b>grado</b>	= * nivel de estudios del profesor *
<b>grupo</b>	= * conjunto de alumnos asignados a una materia en particular *

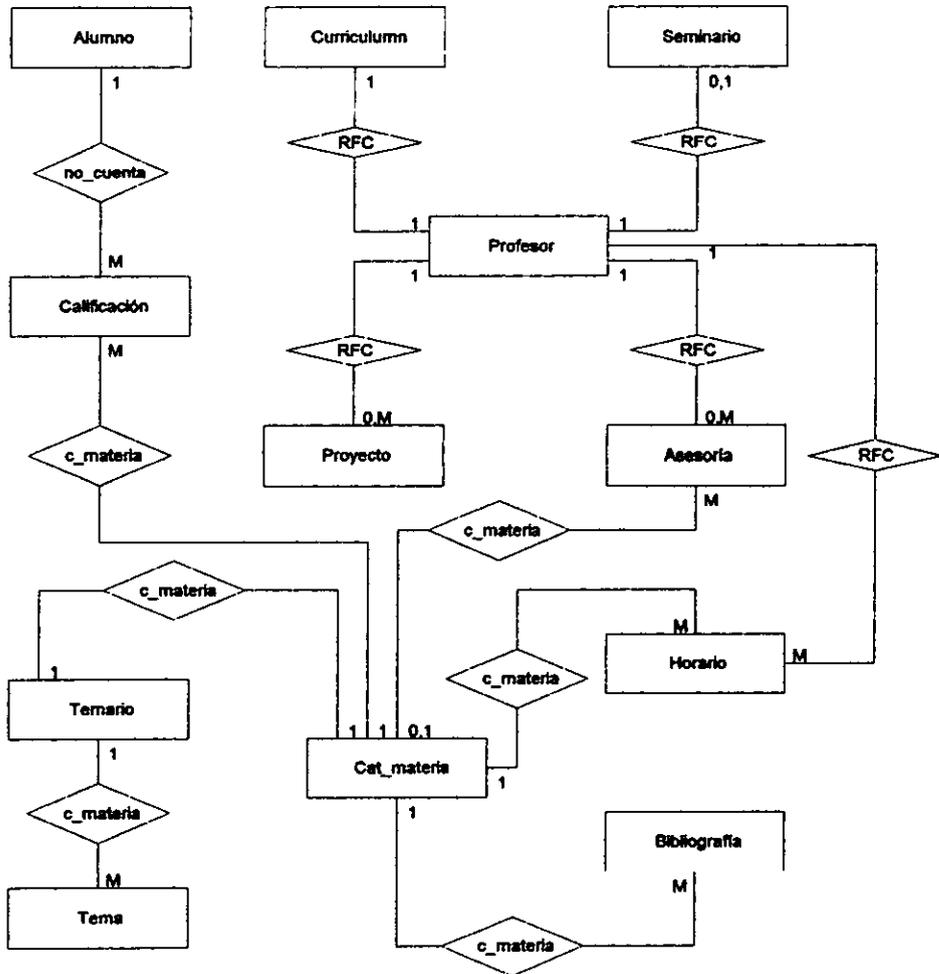
<b>horarios</b>	= semestre + materia + hora_imparte + nom_profesor
<b>hora_asesoría</b>	= * hora en que se imparte la asesoría de la materia *
<b>hora_imparte</b>	= * hora en que se imparte la materia *
<b>hora_seminario</b>	= * hora en que se imparte el seminario *
<b>inf_proyectos</b>	= 1 { área + nom_proy + no_tesistas + nom_profesor } N
<b>inf_seminarios</b>	= 1 { área + nom_semi + nom_profesor + límite_inscritos + hora_seminario } N
<b>libros</b>	= cat_libros
<b>límite_inscritos</b>	= * número de alumnos necesarios para la realización del seminario *
<b>lugar_imp</b>	= * país en que se imprimió el libro *
<b>materia</b>	= * conjunto de conocimientos impartidos a los alumnos *
<b>no_cuenta</b>	= * número identificativo del alumno *
<b>no_tesis</b>	= * número identificativo del trabajo de titulación *
<b>no_tesistas</b>	= * número de tesistas requeridos para la realización del proyecto *
<b>nombre</b>	= [ apellido_paterno + apellido_materno + nombre(s) ]

<b>nom_alumno</b>	= nombre * nombre del alumno *
<b>nom_libro</b>	= * nombre de la publicación *
<b>nom_profesor</b>	= nombre * nombre del profesor *
<b>nom_proy</b>	= * nombre del proyecto *
<b>nom_semi</b>	= * nombre del seminario *
<b>nom_tesis</b>	= * nombre del trabajo de titulación *
<b>nuevas_adquisiciones</b>	= cat_libros
<b>personales</b>	= [ datos_profesor   datos_alumno   curriculumns ]
<b>profesor</b>	= datos_profesor
<b>profesores</b>	= * representa el personal docente de la división *
<b>proyectos</b>	= inf_proyectos
<b>seminarios</b>	= inf_seminarios
<b>status</b>	= [ alumno   egresado   titulado ]
<b>temarios</b>	= @ cve_materia + materia + semestre + temas
<b>temas</b>	= * descripción del capitulado *
<b>tesis</b>	= @ no_tesis + nom_tesis + autor_tesis + materia + área + año
<b>titulación</b>	= [ asesorías   inf_proyectos   inf_seminarios ]

título = [ nom\_libro | nom\_tesis ]

ubicación = \* lugar físico asignado en la biblioteca \*

### Diagrama Entidad - Relación.



### Definición de entidades y atributos.

Nombre de la entidad :           **ALUMNO**

Descripción : En esta tabla se registra la información personal de los alumnos.

Llave primaria :	<b>no_cuenta</b>	<b>Alfabético(9)</b>
Otros atributos :	cve_carrera	Numérico(4) Catálogo
	a_paterno	Alfabético(15)
	a_materno	Alfabético(15)
	nombre	Alfabético(20)
	calle	Alfabético(25)
	colonia	Alfabético(20)
	ciudad	Alfabético(15)
	cp	Numérico(5)
	nación	Alfabético(15)
	tel	Alfabético(9)
	sexo	Alfabético(1)
	edad	Numérico(2)
	estatus	Alfabético(1) Catálogo.

**Nombre de la entidad :            PROFESOR**

**Descripción :** En esta tabla se registran los datos personales de los profesores.

<b>Llave primaria :</b>	<b>RFC</b>	<b>Alfabético(14)</b>
<b>Otros atributos :</b>	<b>a_paterno</b>	<b>Alfabético(15)</b>
	<b>a_materno</b>	<b>Alfabético(15)</b>
	<b>nombre</b>	<b>Alfabético(20)</b>
	<b>especialidad</b>	<b>Alfabético(15)</b>
	<b>t_laborado</b>	<b>Numérico(2)</b>
	<b>categoria</b>	<b>Alfabético(15)</b>
	<b>grado</b>	<b>Alfabético(1) Catálogo</b>
	<b>calle</b>	<b>Alfabético(25)</b>
	<b>colonia</b>	<b>Alfabético(20)</b>
	<b>ciudad</b>	<b>Alfabético(15)</b>
	<b>cp</b>	<b>Numérico(5)</b>
	<b>nación</b>	<b>Alfabético(15)</b>
	<b>tel</b>	<b>Alfabético(9)</b>
	<b>sexo</b>	<b>Alfabético(1)</b>
	<b>edad</b>	<b>Numérico(2)</b>
	<b>edo_civil</b>	<b>Alfabético(1)</b>

Nombre de la entidad : **CURRICULUM**

Descripción : En esta tabla se registra y actualiza el *curriculum vitae* de los profesores.

Llave primaria : **RFC** **Alfabético(14)**

Otros atributos : **texto** **Texto**

Nombre de la entidad : **CALIFICACIÓN**

Descripción : Los alumnos pueden solicitar las calificaciones de las materias que cursaron en un semestre. Dicha información será almacenada en esta tabla.

Llave primaria :	<b>cve_materia</b>	<b>Numérico(4)</b>
	<b>no_cuenta</b>	<b>Alfabético(9)</b>
Otros atributos :	<b>calificación</b>	<b>Numérico(2,1)</b>
	<b>área</b>	<b>Numérico(2) Catálogo</b>
	<b>semestre</b>	<b>Numérico(1)</b>
	<b>cve_carrera</b>	<b>Numérico(4) Catálogo</b>
	<b>grupo</b>	<b>Numérico(2)</b>

Nombre de la entidad :           **HORARIO**

Descripción : En esta tabla se registran los horarios de las asignaturas para el período a cursar.

Llave primaria :	<b>cve_materia</b>	<b>Numérico(4)</b>
	<b>grupo</b>	<b>Numérico(2)</b>
Otros atributos :	<b>RFC</b>	<b>Alfabético(14)</b>
	<b>tipo</b>	<b>Alfabético(1)</b>
	<b>día</b>	<b>Alfabético(15)</b>
	<b>hora_ini</b>	<b>Hora</b>
	<b>hora_fin</b>	<b>Hora</b>
	<b>salón</b>	<b>Alfabético(4)</b>
	<b>cupos</b>	<b>Numérico(2)</b>
	<b>folio</b>	<b>Numérico(6)</b>
	<b>período</b>	<b>Alfabético(6)</b>

**Nombre de la entidad :            TEMARIO**

**Descripción :** En esta tabla se registra la información general de las asignaturas para que oportunamente los alumnos conozcan el objetivo de las mismas.

**Llave primaria :                    cve\_materia            Numérico(4)**

**Otros atributos :**

horas	Numérico(2)
semestre	Numérico(1)
dur_semana	Numérico(2)
objetivo	Texto
obliga	Alfabético(1)

Nombre de la entidad :           **TEMAS**

Descripción : En esta tabla se registra el nombre y la duración por tema de una materia.

Llave primaria :                   **cve\_materia**           **Numérico(4)**

Otros atributos :                   **no\_tema**               **Numérico(1)**  
  **descripción**           **Alfabético(40)**  
  **horas**                 **Numérico(2)**

Nombre de la entidad : **BIBLIOGRAFÍA**

Descripción : En esta tabla se almacena la bibliografía relacionada con cada asignatura, así como la información de las tesis de los egresados de la división.

Llave primaria : **tipo** Numérico(2)

**número** Numérico(4)

Otros atributos : **nombre** Alfabético(50)

**autor** Alfabético(30)

**cve\_materia** Numérico(4)

**área** Numérico(2) Catálogo

**año** Numérico(4)

**editorial** Alfabético(20)

**ubicación** Alfabético(15)

**edición** Numérico(2)

**lugar\_imp** Alfabético(25)

Nombre de la entidad : **ASESORÍA**

Descripción : En esta tabla se registra la información relacionada a las asesorías que imparte el profesorado dentro de la división.

Llave primaria :                    **RFC**                    **Alfabético(14)**  
   **cve\_materia**           **Numérico(4)**

Otros atributos :                    **dia**                    **Alfabético(15)**  
   **área**                    **Numérico(2) Catálogo**  
   **hora\_ini**                    **Hora**  
   **hora\_fin**                    **Hora**

**Nombre de la entidad :            PROYECTO**

**Descripción :** En esta tabla se almacenan los proyectos que se encuentran disponibles para que el alumnado los desarrolle.

**Llave primaria :**

<b>RFC</b>	<b>Alfabético(14)</b>
<b>nom_proy</b>	<b>Alfabético(50)</b>

**Otros atributos :**

<b>área</b>	<b>Numérico(2)</b>
<b>no_tesista</b>	<b>Numérico(2)</b>

Nombre de la entidad :           **SEMINARIO**

Descripción : En esta tabla se registran los datos referentes a los seminarios de titulación

Llave primaria :           **RFC**                   **Alfabético(14)**

**nom\_semi**           **Alfabético(50)**

Otros atributos :           **área**                   **Numérico(2)**

**lim\_ins**           **Numérico(2)**

**dia**                   **Alfabético(15)**

**hora\_ini**           **Hora**

**hora\_fin**           **Hora**

Nombre de la entidad :           **CAT\_MATERIA**

Descripción : En esta tabla se almacenan los datos referentes a las materias que son impartidas dentro de la división.

Llave primaria :	<b>clave</b>	<b>Numérico(4)</b>
Otros atributos :	<b>área</b>	<b>Numérico(2) Catálogo</b>
	<b>nombre</b>	<b>Alfabético(30)</b>
	<b>semestre</b>	<b>Numérico(2)</b>
	<b>créditos</b>	<b>Numérico(2)</b>

### **3.4 PROPUESTA DE SOLUCIÓN.**

En base a la información obtenida referente a la forma en que deberá comportarse el sistema de información de la DIMEI y visualizando las ventajas que nos proporciona el manejador de base de datos relacional Informix se plantea lo siguiente :

- Crear un sistema automatizado que permita satisfacer las necesidades de todas las entidades involucradas (Alumnos, Profesores, Coordinación) haciendo posible la sustitución de los procesos manuales que hasta el momento se han realizado.
- Se propone que el ambiente de desarrollo sea bajo Informix-DOS ya que la División no cuenta con un equipo multiusuario y sí con uno acorde a las características de esta herramienta. La migración a una plataforma multiusuario (UNIX) se podrá realizar con un mínimo de recodificación debido a la flexibilidad que ofrece el Informix.
- La utilización de un lenguaje de cuarta generación como lo es Informix-4gl contempla el crecimiento en la funcionalidad e integridad del sistema. Con esto se prevee la utilización del sistema englobando a nuevas divisiones pertenecientes a la F.I.

### **3.5 DISEÑO DEL SISTEMA.**

El siguiente paso una vez que se ha concluido el análisis es el diseño, el analista debe asegurarse de entender los requerimientos del usuario, ya que en la siguiente etapa el diseñador deberá implantar dichos requerimientos en el mundo real.

Durante el diseño, se desarrollan, revisan y documentan progresivos refinamientos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones del software de calidad que pueden establecerse.

Las estructuras de programa y de datos contribuyen a dar una visión global de la arquitectura del software, mientras que los procedimientos dan los detalles necesarios para la implementación de los algoritmos.

Para evaluar una representación del diseño debemos establecer criterios para su modelo :

1. Un diseño debe exhibir una organización jerárquica que haga un uso inteligente del control entre los elementos del software.
2. Un diseño debe ser modular ; esto es, el software debe estar particionado lógicamente en elementos que realicen funciones y subfunciones específicas.

3. Un diseño debe contener una representación distinta y separable de los datos y procedimientos.
  
4. Un diseño debe derivarse usando un método repetible que este conducido por la información obtenida durante el análisis de requerimientos de software.

La importancia del diseño radica primordialmente en la calidad. Sin un diseño se corre el riesgo de construir un sistema inestable cuyo mantenimiento resulte costoso, que puede ser difícil de probar y que falle cuando se realicen pequeños cambios.

### **3.5.1 ESTRUCTURAS DE PROGRAMA.**

Dentro de las estructuras de programa se utiliza la llamada modularidad que consiste en dividir el programa en varios grupos de instrucciones llamados módulos.

Para tener una buena modularidad es necesario que:

- Cada módulo tenga una sola función.
  
- La función de cada módulo debe ser completa.

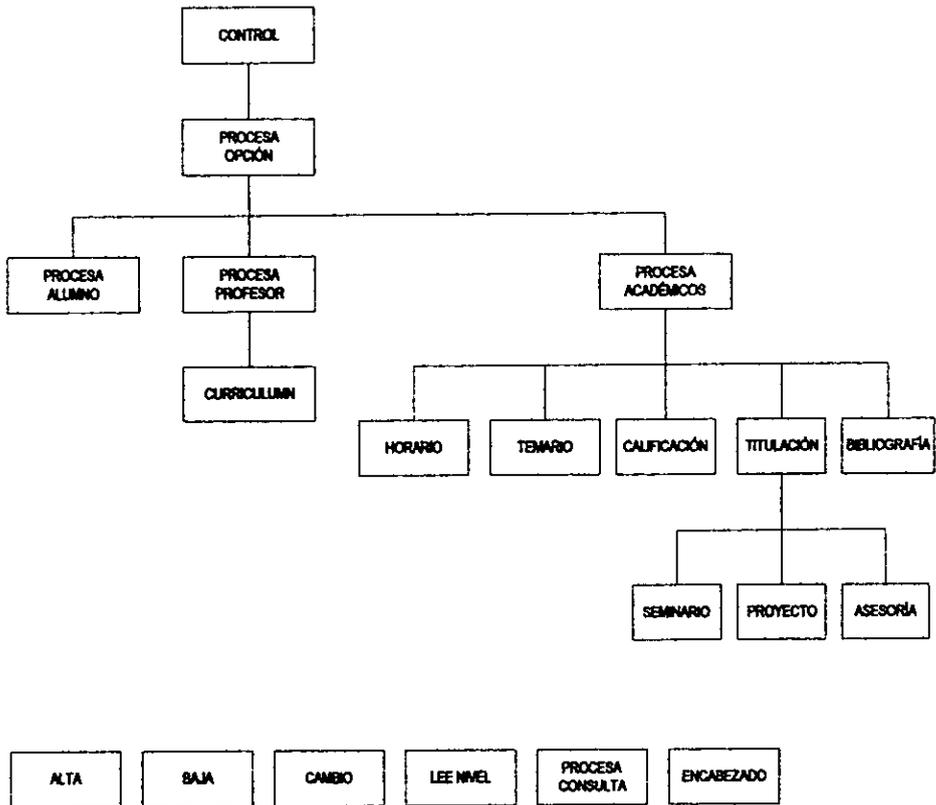
- La división en módulos debe ser de acuerdo a la estructura lógica del problema a resolver.
- La división en módulos debe aislar los efectos de las posibles modificaciones al programa.
- Un módulo debe de ser de un tamaño que sea fácilmente comprensible (si la función es muy extensa, se subdivide en subfunciones y la función del módulo original no se altera).

La estructura del programa representa la organización (frecuentemente jerárquica) de los componentes del programa (módulos) e implica una jerarquía de control. No representa aspectos procedimentales del software tales como secuencia de procesos, ocurrencia de decisiones o repetición de operaciones.

Para representar una estructura de programa se utilizan muchas notaciones diferentes, la más común es el llamado *diagrama de árbol*.

Cada módulo del árbol se representa como un rectángulo y tiene conectados hacia abajo los módulos que dependen de él exclusivamente. En la parte inferior se representan sin estar conectados hacia arriba, los módulos compartidos (módulos que pueden ser ejecutados desde diferentes módulos).

## ÁRBOL DE MÓDULOS.



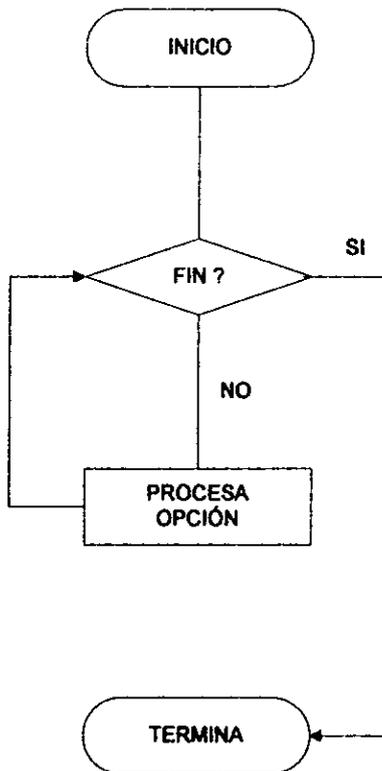
### **3.5.2 DISEÑO GENERAL DEL SISTEMA.**

El diseño general es la siguiente etapa dentro del desarrollo del diseño del sistema, es aquí donde se representa la información que contiene cada uno de los módulos de una manera mas detallada, destacándose la forma en que se va a procesar la información dentro de cada uno de ellos.

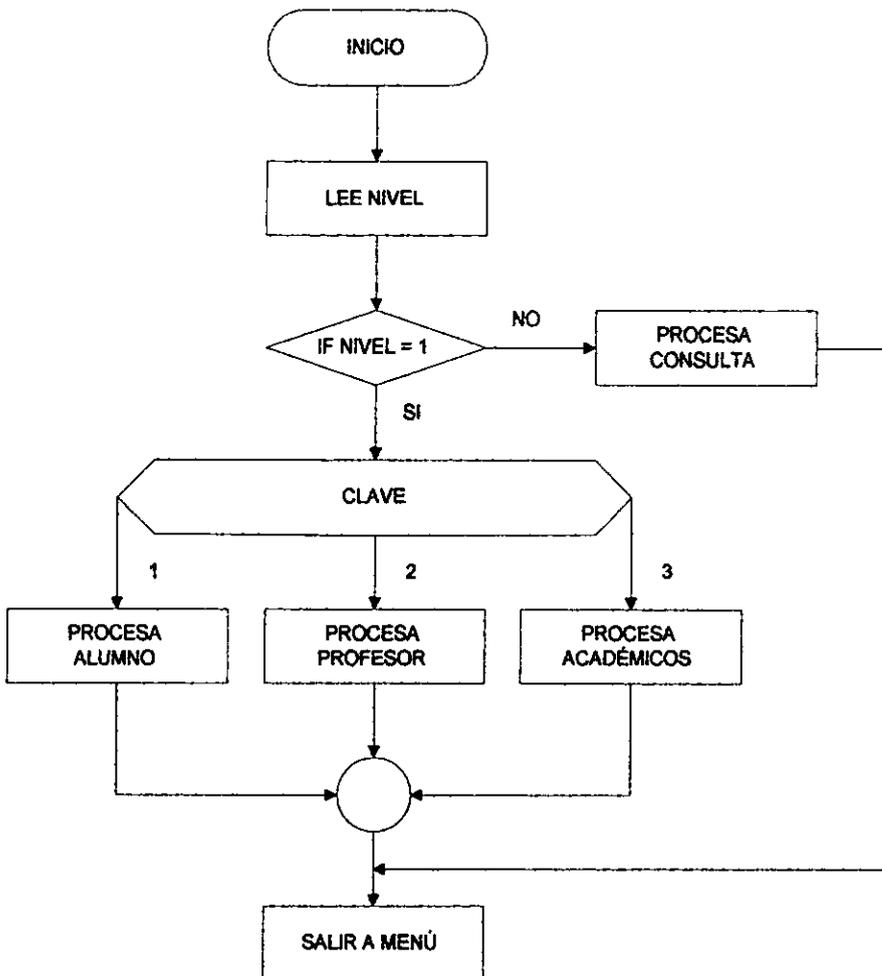
La simbología utilizada en esta etapa son las estructuras de control, las cuáles determinan el flujo de la información durante la ejecución del programa.

La técnica utilizada es la conocida como Top-Down. Esta metodología es útil dentro del diseño de sistemas usando las ideas de modularidad y estructuras de control ; consiste en atacar el diseño de lo general hacia lo particular. Con esta técnica se deben ignorar los detalles de los módulos que se están definiendo, en este momento lo importante es qué harán los módulos y no cómo lo harán.

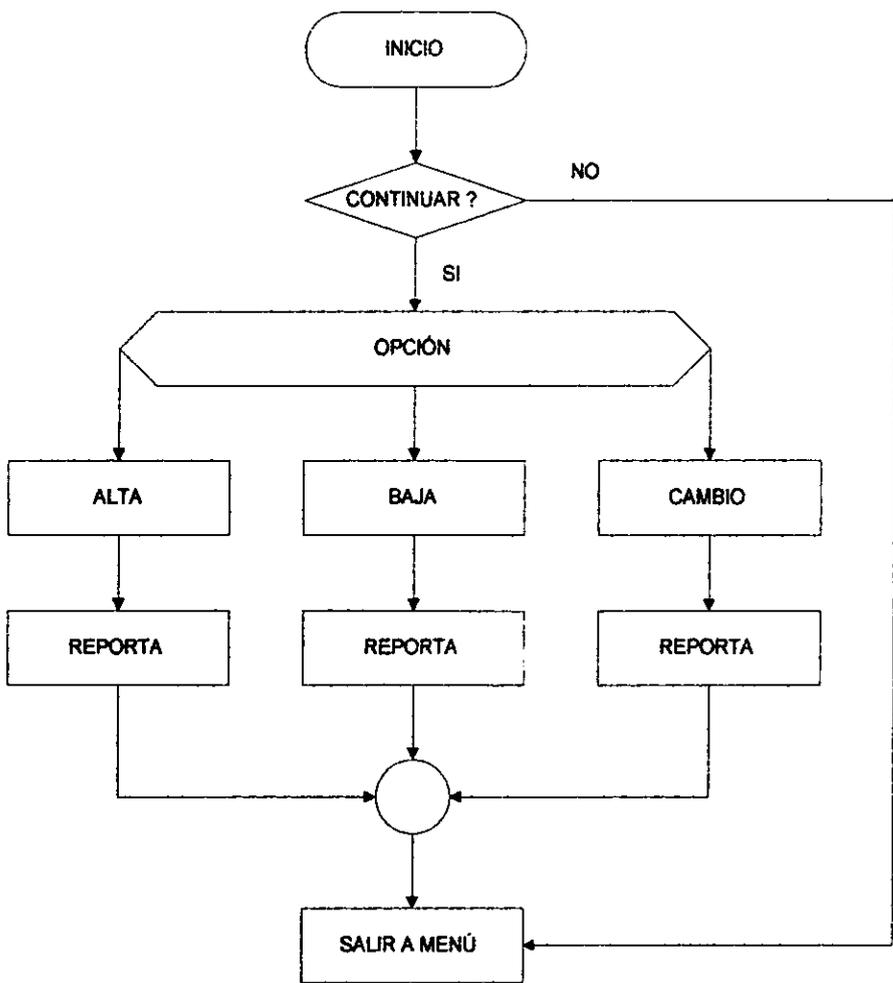
## MÓDULO CONTROL.



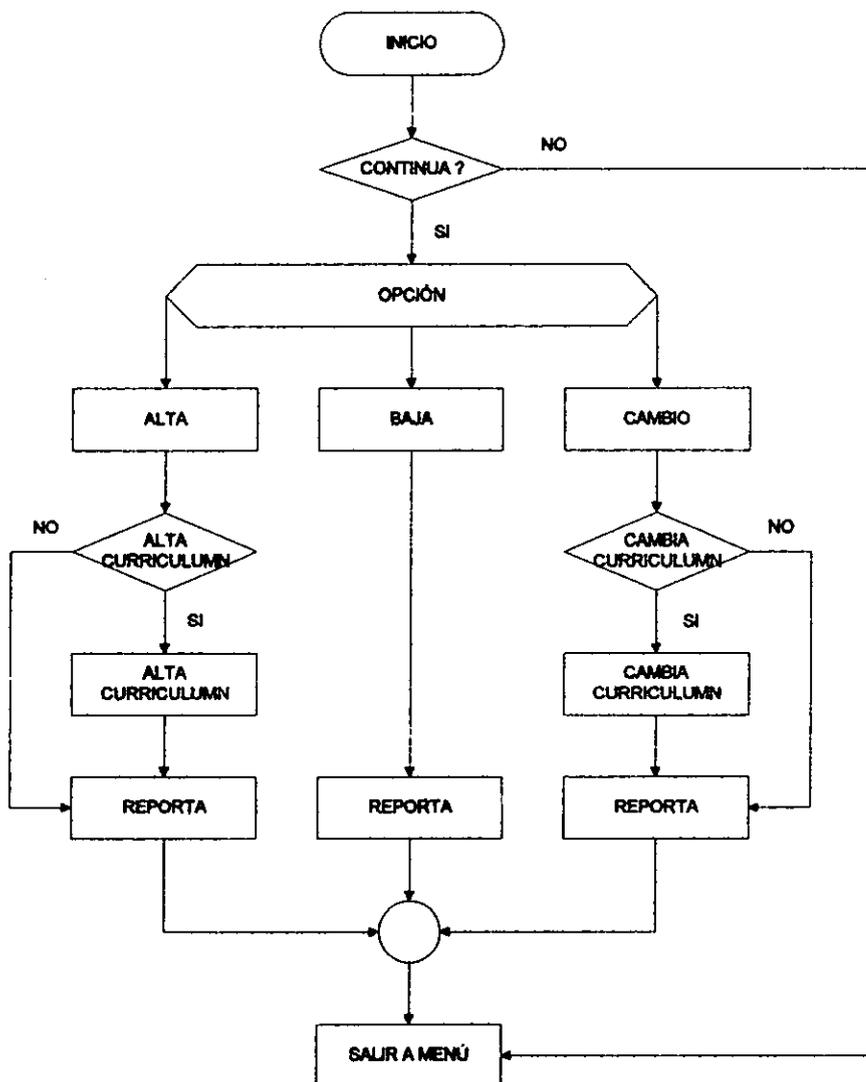
### MÓDULO PROCESA OPCIÓN.



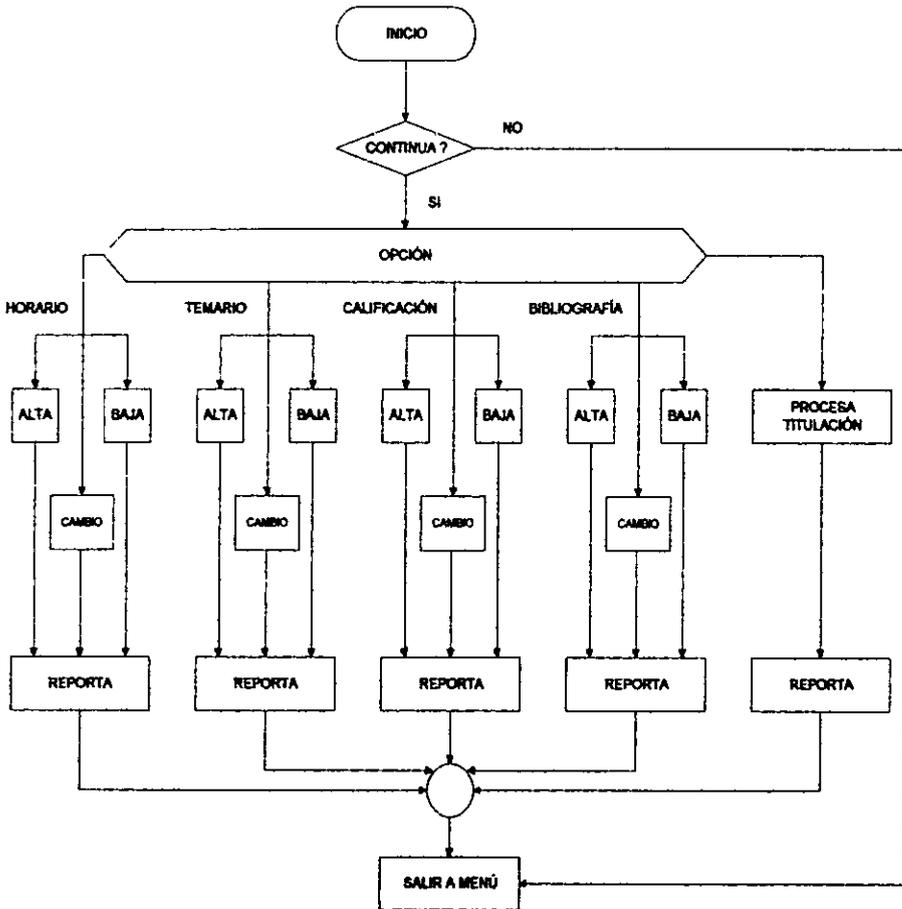
### MÓDULO PROCESA ALUMNO.



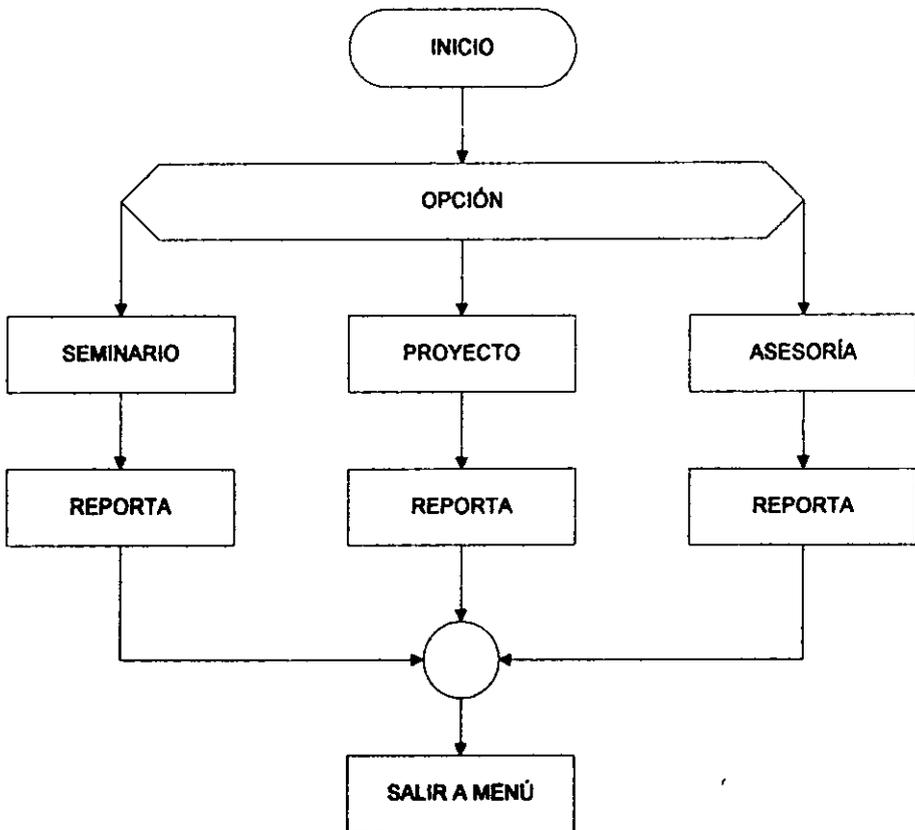
## MÓDULO PROCESA PROFESOR.



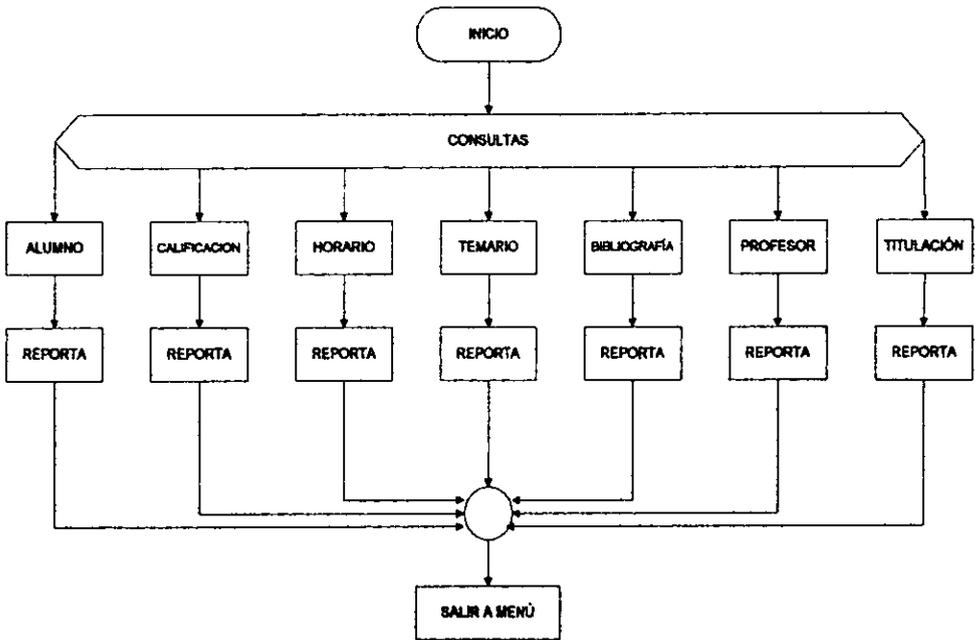
## MÓDULO PROCESA ACADÉMICOS.



## MÓDULO PROCESA TITULACIÓN.



## MÓDULO PROCESA CONSULTA.



### **3.5.3 DISEÑO DE PANTALLAS.**

Los prototipos de las pantallas de captura de información se realizaron en base al generador de formas de Informix, el cuál utiliza menús de anillo y permite tener la opción de scroll entre cada uno de los campos desplegados en la pantalla, con lo que el usuario puede corregir errores de tecleo de la información dentro de los campos antes de confirmar la realización del proceso. Además se puede incluir el formato de entrada de los datos para tener una estandarización en la captura de la información.

Las siguientes pantallas detallan de forma general la presentación de cada una de las entidades y la información que debe capturarse en cada una de ellas.

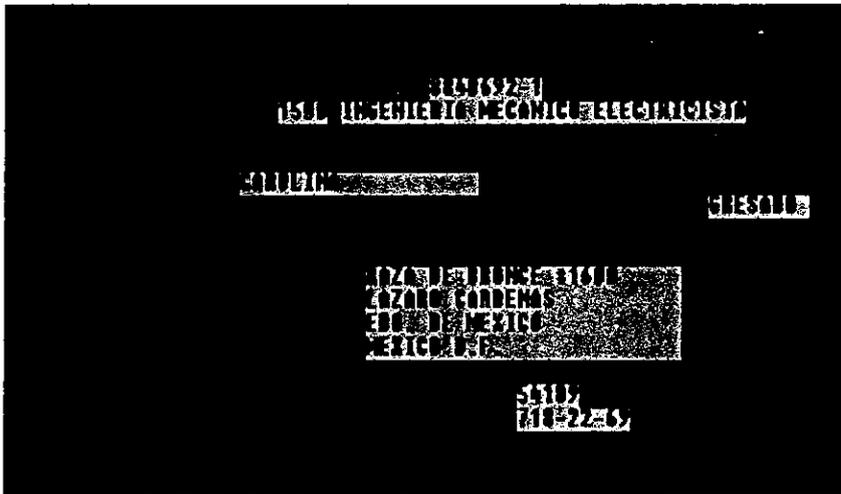
La pantalla de *Materias de la DIMEI* es una de las más importantes ya que está se encarga de actualizar el catálogo en la base de datos. Si es necesario el despliegue del nombre de la materia en alguna pantalla del sistema, la información será tomada de dicho catálogo.

Las pantallas *Alumno* y *Profesor* se encargan de realizar el mantenimiento de sus datos personales, éstas son similares y se dividen en dos secciones ; la primera contiene los datos académicos y la segunda, la dirección de los mismos.

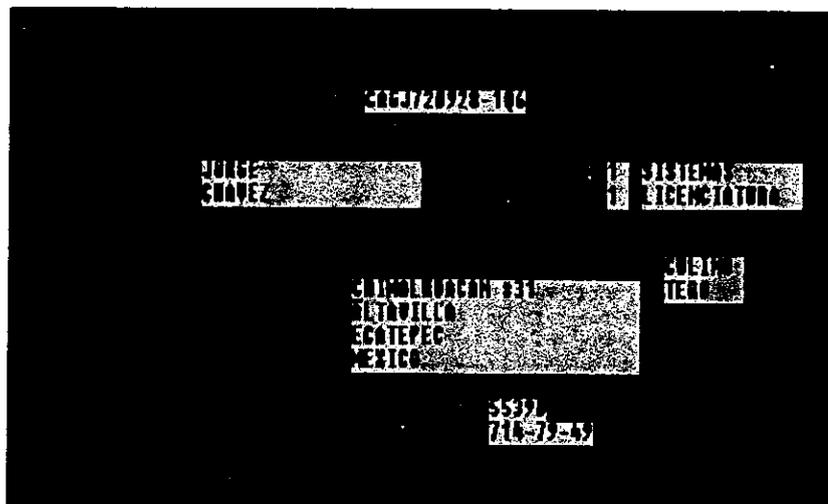
## PANTALLA : CATÁLOGO DE MATERIAS.

II. INDUSTRIAL			
152	ESTUDIO DEL TRABAJO	4	10
112	SEMINARIO DE INGENIERIA I	9	4
162	QUIMICA APLICADA	3	8
1627	ESTADISTICA AVANZADA	6	8

## PANTALLA : DATOS DEL ALUMNO.



## PANTALLA : DATOS DEL PROFESOR.



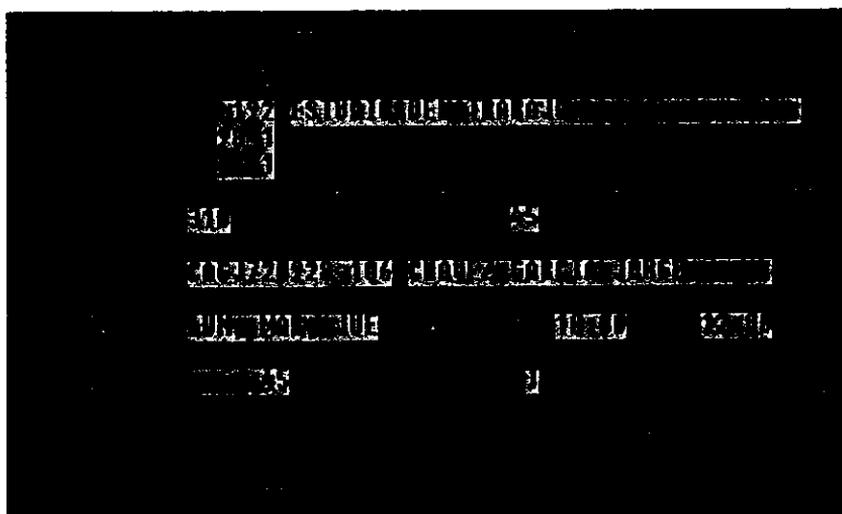
Continuando con los prototipos de entrada de datos, las siguientes pantallas son una selección de aquellas que agrupan la información académica que maneja actualmente la coordinación con respecto a los alumnos y profesores de la DIMEI.

### PANTALLA : CALIFICACIONES.

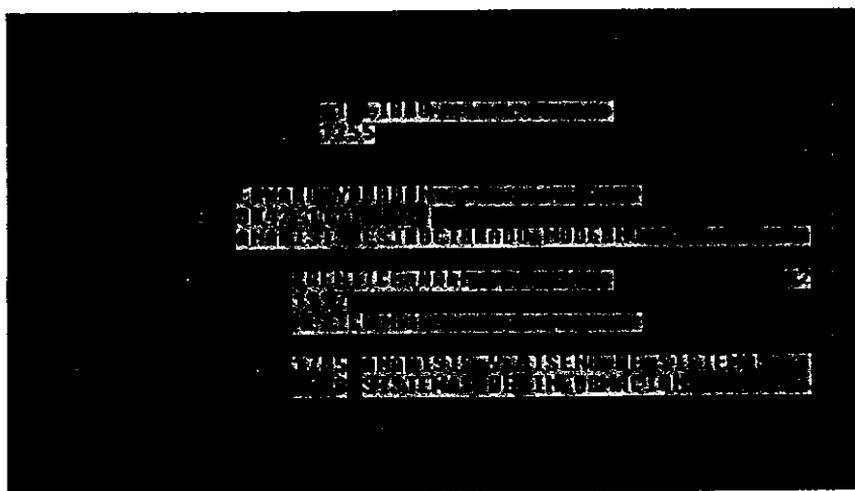
The screenshot shows a list of students and their grades for three subjects: INGENIERIA MECANICO ELECTRICISTA, SISTEMAS ADMINISTRATIVOS, and ANALISIS FINANCIERO. The data is presented in a table format with columns for student ID, name, and grade.

Identificación	Nombre	Calificación
11500	Jorge Cruz Garcia	9.0
201622	Carolina Herrera Estrada	9.0
207807	Mariana Alcocer Herrera	8.0
209202	Josue Aldama Martinez	10.0
209802	Roberto Cruz Hernandez	7.0

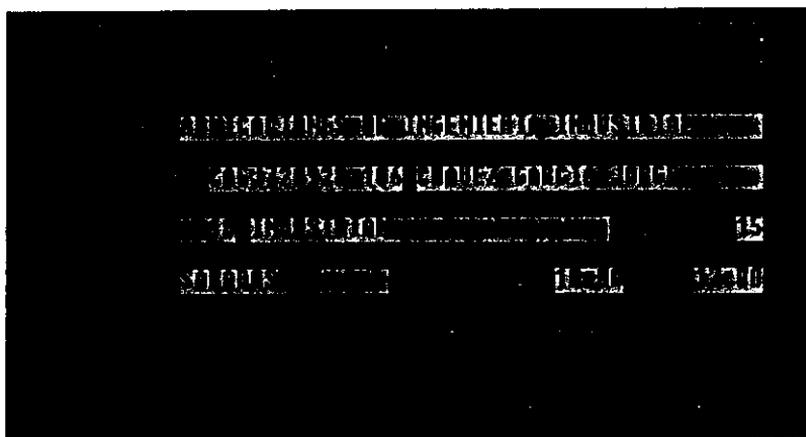
## PANTALLA : HORARIOS.



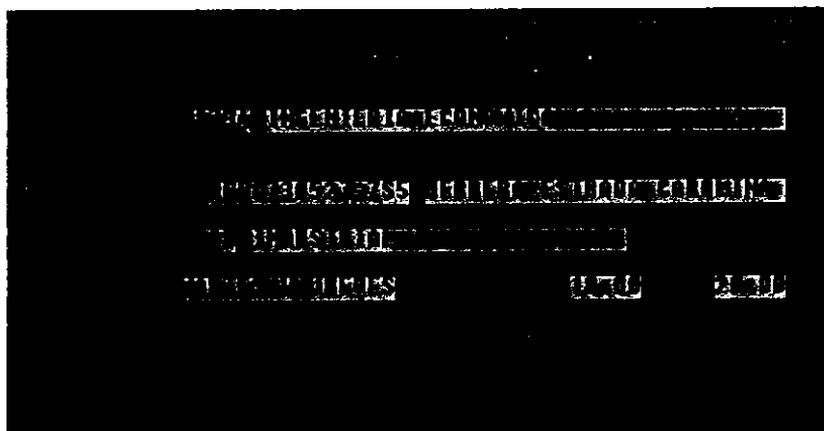
## PANTALLA : BIBLIOGRAFIA.



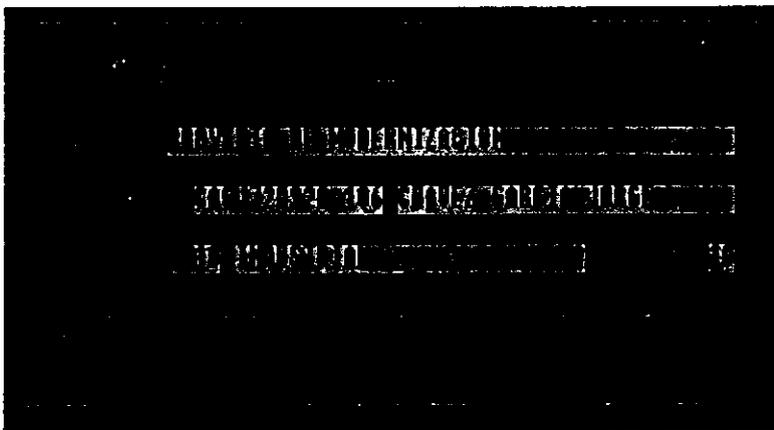
### PANTALLA : SEMINARIO.



## PANTALLA : ASESORÍA.



## PANTALLA : PROYECTO.



### 3.6 PRUEBAS E IMPLANTACIÓN.

La prueba del software es un elemento crítico para la garantía de calidad del sistema y representa un último repaso de las especificaciones del diseño y de la codificación.

La creciente aparición del software como un elemento más de muchos sistemas y la importancia de los *costos* asociados a un fallo del mismo están motivando la creación de pruebas minuciosas y bien planificadas.

La prueba presenta una interesante anomalía del funcionamiento del sistema. Durante las fases anteriores de definición y de desarrollo, el responsable intenta construir el software partiendo de un control abstracto y llegando a una implantación tangible. A continuación llega la prueba. El desarrollador crea una serie de casos de prueba que intentan *demoler* el sistema que ha sido construido.

Los objetivos de la prueba son :

1. La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

3. Una prueba tiene éxito si descubre un error no detectado hasta el momento.

Los objetivos anteriores suponen un cambio dramático del punto de vista. Nos quita la idea que normalmente se tiene de que una prueba tiene éxito si no descubre errores. El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clase de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Aún si se cumplen los objetivos de la prueba, ésta no puede asegurar la ausencia de defectos ; sólo puede demostrar que existen anomalías en el software.

Cualquier producto de ingeniería puede ser probado de dos formas : 1) conociendo la función específica para la que fue diseñado el sistema, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa ; 2) conociendo el funcionamiento del sistema, se puede desarrollar pruebas que aseguren que cada módulo cumpla con la función asignada ; o sea que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. La primera aproximación de prueba se denomina *prueba de la caja negra* y la segunda *prueba de la caja blanca*.

### 3.6.1 PRUEBA DE LA CAJA BLANCA.

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Es decir, aplica pruebas de unidad mediante un análisis minucioso a cada proceso de los módulos independientes, para garantizar que :

- ◆ Se ejecuten por lo menos una vez todas las instrucciones de cada módulo.
- ◆ Se aplican todas las decisiones lógicas en sus bifurcaciones correspondientes a verdadero y falso ( IF-THEN-ELSE).
- ◆ Se ejecuten todos los bucles llevándolos a sus límites operacionales (WHILE, FOR).
- ◆ Se utilicen las estructuras de almacenamiento de datos para asegurar la validez de correspondencia *dato-campo*.

### 3.6.2 PRUEBA DE LA CAJA NEGRA.

Los métodos de la prueba de la caja negra se centran en los requerimientos funcionales del sistema. La prueba de la caja negra garantiza que la entrada se acepta de forma adecuada y se produce una salida correcta, ignorando la

estructura lógica del sistema. Es decir, se ignoran estructuras de control y concentra su atención en el dominio de la información.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías :

- ♦ Funciones incorrectas o ausentes.
  
- ♦ Errores de interfases.
  
- ♦ Errores de estructuras de datos o en accesos a base de datos externas.
  
- ♦ Errores de rendimiento.
  
- ♦ Errores de inicialización y de terminación.

### **3.6.3 IMPLANTACIÓN.**

Se denomina implantación al proceso que asegura la operatividad del sistema de información y que permite al usuario obtener beneficios por su operación.

En esta etapa es importante la capacitación de los usuarios que harán uso del sistema, tanto de aquellos que actualizarán la base de datos como de los que consultarán su información.

En el Sistema de Información de la DIMEI se contempla tener un esquema de seguridad con el fin de resguardar la integridad de la información de la base de datos, ya que sólo se tendrán dos tipos de usuarios se propone tener un nivel de seguridad para cada uno de ellos. El primer nivel está conformado por las personas que están a cargo del mantenimiento y actualización del sistema, y el segundo corresponde a los alumnos, profesores y egresados que obtendrán información académica.

A cada uno de los usuarios se les debe brindar capacitación de acuerdo a su nivel. El tiempo requerido para su realización será determinado por la DIMEI.

El prototipo del sistema debe implantarse inicialmente en los tres equipos disponibles dentro de la división para el conocimiento y aprobación de la comunidad universitaria que conforma a la DIMEI. Una vez aceptado, queda abierta la posibilidad de que el sistema soporte a otras divisiones de la Facultad de Ingeniería si así se dispone.



## **CONCLUSIONES.**

Dentro del análisis se definieron los requerimientos en cuanto al software y hardware para dar solución al problema del manejo de información que presentaba la DIMEI, se vio entonces la imperiosa necesidad de plantear el uso de un sistema manejador de bases de datos de tipo relacional ya que se necesitaba manejar grandes volúmenes de información y se requería que ésta siempre estuviera disponible de una manera actualizada.

La decisión de utilizar el hardware disponible dentro de la DIMEI se hizo principalmente basándose en la disponibilidad de almacenamiento interno del equipo y de la flexibilidad del software para desarrollar la aplicación en su ambiente operativo, en cuanto a la elección del DBMS se hizo después de comparar tres de los importantes manejadores de base de datos de tipo relacional, los cuáles dentro del mercado nacional son los más utilizados en el desarrollo de procesos complejos. Tomamos en cuenta varios puntos que a nuestro criterio y en base a las opiniones recopiladas de aquellas personas que los han manejado son indispensables para tener un buen soporte en la creación de un nuevo sistema, entre los cuales se encuentran: su definición y manipulación de datos y el desarrollo de aplicaciones, lo anterior nos llevo a decidir que el DBMS que cumplía con los requerimientos para la realización del sistema era Informix , ya que a diferencia de los otros tiene un alto performance en todas las aplicaciones requeridas por muy complejas que estas

sean, aunado a las características y ventajas que presenta con respecto a los manejadores de archivos y a otros DBMS.

Para el desarrollo del sistema se contemplo el uso del análisis estructurado, ya que es de gran ayuda para comprender sistemas grandes y complejos, de esta manera, nos ayudó a organizar las tareas asociadas a la determinación de requerimientos de una forma completa y exacta de la situación actual de la DIMEI. Al hacer uso de éste análisis, se planteó la utilización de la programación estructurada, la cuál permite tener lógicas sencillas, comprensibles y modificables, éstas evitan los llamados “parches” de programas, códigos no entendibles y por lo tanto difíciles de modificar en su estructura.

Un factor determinante para el buen funcionamiento del sistema se encuentra en la realización de las pruebas (Caja Blanca, Caja Negra), las cuales establecerán la confiabilidad y eficiencia que ofrece SIDIMEI (Sistema de Información de la División de Ingeniería Mecánica e Industrial), el cuál tiene principalmente las siguientes responsabilidades :

- ◆ Control de acceso a la información.
- ◆ Captura de datos del alumnado y profesorado de la DIMEI.
- ◆ Actualización de la base de datos.
- ◆ Realización de consultas y reportes.

El sistema tiene la opción a crecer debido a como fue desarrollado dentro del ciclo de sistemas, es decir, cuando se requiera puede cubrir el manejo de la información de las divisiones que conforman a la Facultad de Ingeniería sin tener problema alguno.

Para finalizar, la información proporcionada por la DIMEI, aunada a la amplia investigación realizada y en base a los resultados obtenidos en la metodología empleada para el desarrollo del sistema de información, se concluye que el objetivo de la tesis fue cubierto en su totalidad.

## BIBLIOGRAFÍA.



Análisis estructurado moderno.  
Edward Yourdon  
Prentice Hall  
México, 1996.



Fundamentos de base de datos.  
Henry F. Korth  
Abraham Silberschatz  
Mc Graw-Hill  
México, 1988.



Procesamiento de datos en UNIX.  
R. S. Tare  
S. Mc Graw Hill  
México, 1990.



Sistema operativo UNIX.  
Rebeca Thomas  
Jean Yates  
Osborne/Mc Graw-Hill  
México, 1988.



Ingeniería del software.  
Roger S. Pressman  
Mc Graw-Hill  
México, 1988.



Diseño de sistemas de información.  
John G. Burch  
Gary G. Roudnitski  
Limusa  
México, 1991.



Sistemas de administración de base de datos.  
David Kruglinski  
Osborne/McGraw-Hill  
México, 1987.



Systems design in a database enviroment.  
K. S. Brathwaite  
Mc Graw-Hill  
México, 1989.



Developing applications using Informix-4gl.  
Informix Software Inc.  
USA, 1990.



DOS Guía para usuarios expertos.  
Kris Jamsa  
Mc Graw-Hill  
México, 1989.



Using SQL  
J. R. Groff  
K. Weinberg  
Osborne/MC Graw-Hill  
México, 1991.