



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

---

FACULTAD DE INGENIERÍA

SISTEMA DE INFORMACIÓN  
AUTOMATIZADO PARA REALIZAR UN  
SEGUIMIENTO ACADÉMICO DE LOS  
ALUMNOS DE LA DIMEI (DIVISIÓN DE  
INGENIERÍA MECÁNICA E  
INDUSTRIAL)

## TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN  
P R E S E N T A N:

**ISIDRO MERCADO ÁNGEL  
REYES AGUIRRE XOCHITL GHORETY**

DIRECTOR DE TESIS: ING. AURELIO SÁNCHEZ VACA



CIUDAD UNIVERSITARIA 2005

m. 341146



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

## **Agradecimientos de Angel Isidro Mercado:**

A mi amiga y compañera de tesis **Xochitl Ghorety Reyes Aguirre:**

Gracias por el tiempo dedicado a este proyecto de tesis, por tu amistad y el apoyo que he recibido de ti, por los buenos momentos que pase en compañía de tu persona, te lo agradezco mucho.

Porque se que esta tesis es tan importante para ambos y con ella concluimos una etapa importante en nuestras vidas. Felicidades y que todas tus metas se cumplan.

A mi madre **Antonia Mercado de Isidro:**

A ti mamá por darme la vida y proporcionarme la oportunidad de seguir mis estudios, por todo tu apoyo incondicional para conseguir mis objetivos y metas. Gracias te doy por ese gran amor, cariño y comprensión en los momentos difíciles. Por tus grandes y sabios consejos, por esas noches en vela y toda la dedicación que haz tenido con mi persona.

Este es y será el fruto de tu esfuerzo, es algo que deseo compartir con tigo.

### **“MIL GRACIAS DE CORAZÓN”**

De corazón te doy mil gracias, no tengo palabras para expresa lo que ha significado todo este esfuerzo y el logro que he conseguido, he recibido más de lo esperado y deseo regresar este esfuerzo con un primer y pequeño gran presente, para ti con cariño, mucho amor y principalmente mi gran respeto, a todo tu esfuerzo que he observado en tu persona y que lo haz entregado con mucho dedicación año tras año.

A mi padre **Angel Isidro Santos:**

Por tus sabios consejos y gran apoyo que me han motivado a terminar todo lo que me proponga. Y sobre todo por el esfuerzo, amor y dedicación que me ha brindado en todos estos años.

Por el tiempo que me ha dedicado y por las buenas recomendaciones, el ejemplo que me demostraste y todos los valores que de ti he adquirido, por la paciencia y el esfuerzo que entregas, te expreso que eres parte importante en las metas que he logrado en mi vida.

En estas palabras te manifiesto mucho de lo que tengo que agradecer, por darme hogar, estudios y principalmente libertad de elección en la mayor parte de mis decisiones.

### **“PARA MIS PADRES”**

A mis padres les dedico este titulo, que ha sido un gran esfuerzo de mi parte y que deseo compartir con ustedes, gracias por todo su apoyo. No tendré como agradecer todo lo que me han brindado, solamente me queda esforzarme día a día para lograr mis metas, ayudarlos, respetarlos y quererlos mucho.

Gracias por lograr no solo mi formación académica también mi formación como ser humano, por darme la vida y por lograr cimentar en mí sus principios y valores. Los he aprendido con el paso del tiempo y hoy en día los valoro más de lo que en algún tiempo no podía imaginar.

Por lo anterior y por muchas cosas que me han entregado de forma ilimitado doy gracias. Los amo, quiero y respeto mucho.

A mis hermanos **Araceli, Ismael y Hugo:**

Gracias por su confianza, el gran apoyo moral, económico y sentimental. Por el tiempo que han dedicado a la familia y por las buenas experiencias compartidas, me han brindado una formación especial. A ustedes también les agradezco por el apoyo incondicional que siempre llevo presente en la mente y con gran cariño les digo que los quiero mucho y tienen mi pleno respeto. Por todas las palabras de animo que me han expresado y por la confianza que han puesto en mi.

Dedico esta tesis a mi sobrina Dianita (hija de mi hermano Ismael) que quiero mucho y también a la niña que tendrá mi hermana Araceli. Son un motivo más para seguir en este camino.

A mis abuelos **Concepción, Marina y Modesto:**

Ustedes son un gran pilar en mi formación personal, el cariño, la bondad, su gran experiencia y los principios que han cimentado en mí. No puedo más que seguir su gran ejemplo. Esa fuerza y ganas de obtener todo sin importar el obstáculo que se presente, me han proporcionado grandes y buenos consejos.

Con cariño y gran respeto a Conchita, Marina y mi abuelito Modesto..... Gracias a ustedes por su muy cordial apoyo que siempre me han demostrado, los quiere mucho su nieto Angel.

A mis **primos:**

Por los buenos momentos que pase en mi juventud, y su incondicional amistad que me han proveído siempre fuerzas para seguir adelante y enfrentar los retos en esta vida. Por los buenos consejos y por estar cerca de mi constantemente y en los momentos que necesite de su compañía. Con cariño menciono a Félix, Ricardo, Paco Víctor, Sandra, Yulma, Belén, Noemí, Eduardo, etc., y todos los que faltan.

Por su amistad y por que se que en cualquier momento podré contar con sus consejos gracias.

A los **Ing. Aurelio Sánchez Vaca** y al **Ing. Héctor Raúl Mejía**

Por su ayuda y asesoría que proporcionaron en la realización de esta tesis, por los buenos consejos y principalmente por el tiempo y dedicación que brindaron durante la ejecución de este proyecto. Muchas gracias por todo el apoyo. Y por ayudarnos a terminar con nuestros estudios.

Al **Ing. Sergio Cruz** y también al **Ing. Javier Montoya:**

Gracias a los dos por creer en mi y darme una oportunidad. Por la formación que me brindaron, principalmente por su apoyo y plena confianza que han guardado en mi. Agradezco la oportunidad que me ofrecieron para crecer y formarme como Ingeniero.

Esta tesis es realizada también por su apoyo y motivación. Estoy total y plenamente convencido de lo importante que ha sido esta etapa y de la cual les agradezco a ustedes de una forma muy especial, por sus consejos y buenos deseos se los gratifico mucho.

Por la amistad y los buenos momentos que he pasado con ustedes. Son y serán parte importante en mi crecimiento y formación como Ingeniero en Computación al brindarme las herramientas y el equipo para ampliar mis conocimientos en esta Área.

Las palabras plasmadas son una muestra de mi agradecimiento.

A mis compañeros y amigos de **UNICA**:

A mis compañeros y amigos de UNICA Ramses, Gabriela, Omar, Jesús, Alejandro, Alejandra etc. gracias por el apoyo y su amistad, ha sido de gran ayuda su compañía y el incondicional aliento que me han proporcionado de forma directa o indirecta y que he recibido con gran aprecio.

Por las buenas amistades que pueden crecer y por los momentos tan agradables que he pasado en este lugar. De la mejor manera y disponibilidad posible agradezco todo lo recibido de parte de ustedes. Gracias y espero que ustedes realicen sus metas y objetivos siempre.

A mis amigos:

A todos las personas que he conocido en mi crecimiento de las cuales guardo un gran recuerdo, amigos de la secundaria (Alejandro, Víctor, José, Juan etc.), CCH Azcapotzalco (Erica, Daniel etc. ), Facultad de Ingeniería (Oswaldo, Karla, Estela, Socorro, Alejandro, Gerardo, Álvaro etc. ) y amigos cercanos a mi casa (Aurelio, Edgar, etc. ). Que siempre me han brindado un apoyo moral, de forma directa o indirecta, gracias por ser parte de mi mundo y de una gran amistad, con un excelente y muy bien fundamentado motivo los recuerdos a todos ustedes por los buenos momentos y excelentes experiencias vividas.

### ***UNA ÚLTIMA REFLEXION***

*He aprendido que el hecho de conseguir este titulo, no tendría ningún significado para mí si no tengo con quien compartir mi esfuerzo, para lograr una meta como esta.*

*El simple hecho de estar rodeado de personas que me han apoyado y brindado toda su confianza, así como sus palabras de ánimo, me han dado fuerzas para seguir adelante en todo reto que se presenta en mi vida.*

*A partir de este momento disfrutaré todos los éxitos con las personas que me rodean, y aprenderé de los fracasos y obstáculos que se presenten, pues son estos lo que me harán crecer como persona para valorar con justa razón todo objetivo y meta que me proponga.*

*Se que nada es fácil en esta vida, pero también estoy convencido que todo lo que uno se propone lo consigue tarde o temprano con dedicación y esfuerzo. También comprendo que lo que mas trabajo cuesta es lo que mayormente valoramos de forma personal.*

*Gracias por estar conmigo y mi familia en estos momentos que son tan importantes para mí.*

***Da un pescado a un hombre y comerá un día, enséñale a pescar y comerá toda su vida.***

# *Agradecimientos*

## **A DIOS:**

Gracias por acompañarme en todos los momentos importantes de mi vida y trabajar arduamente a mi lado.

## **A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería:**

Por haberme abierto un día sus puertas y formarme cultural, profesional y socialmente. Y por que a donde quiera que yo vaya “**por mi raza hablará el espíritu**”.

## **A mis padres, Ma. Teresa y Raúl:**

Por la oportunidad de existir, por su sacrificio, por su ejemplo de superación, por su comprensión y confianza, por su amor y su amistad incondicional, por que sin su apoyo no hubiera sido posible la culminación de mi carrera.

## **A Ulises:**

A ese hombre extraordinario mitad dureza y mitad ternura que me ha enseñado a amar y a luchar, a ser y defender, a crecer y a creer, a llorar y a valorar, a sonreír y a sufrir. Gracias por todo tu apoyo, comprensión y amor, pero sobre todo por que nunca perdiste la confianza y la fe en mí. Por lo que ha sido y será...

## **A mis hermanas Violeta y Zukeyt:**

Ambas son parte importante de mi vida y les agradezco todos los momentos que hemos compartido juntas y por que siempre han ofrecido su mano para seguir adelante.

## **A mis amigos David, Ema, Esperanza, Gaby, Javier, Sergio y Ramses:**

Por lo que representan en mi vida, por su amistad, su apoyo y sus consejos, cuando más los necesite. Ser en todo momento amigos aún cuando estemos distantes.

## **A Fabiola Olivos:**

Gracias, por que en ti encontré una buena amiga, que me ha brindado su apoyo tanto personal como profesional.

**A mi compañero de Tesis Ángel:**

Mil gracias por todo tu apoyo y paciencia, ya que tú ayuda fue fundamental para llevar a cabo la realización de este trabajo. Pero sobre todas las cosas por tu gran amistad.

**A mi director de Tesis, Ing. Aurelio Sánchez Vaca:**

Por todo el apoyo y paciencia que me tuvo, su ayuda y experiencia fue parte fundamental en la obtención de este logro.

**A un amigo muy especial, Ing. Héctor Raúl Mejía Ramírez:**

Gracias, por el tiempo que me dedicaste, por transmitirme tu experiencia, por todo lo que me enseñaste, pero sobre todo por la amistad que me brindaste. Sigue siendo como hasta ahora.

**Para alguien muy especial, Ing. Sebastián Poblano Ordóñez:**

Gracias por su la amistad brindada por tantos años y por toda su experiencia.

*Con mucho cariño.*

*Xochitl Ghorety Reyes A.*

# ÍNDICE

<b>Introducción</b> .....	1
<b>Capítulo I. Antecedentes</b> .....	3
1.1. Misión, estructura y funciones de la DIMEI .....	4
1.2. Problemática .....	6
1.3. Objetivo .....	8
1.4. Alcances y ámbito del sistema .....	8
<b>Capítulo II. Determinación de los requerimientos del sistema</b> .....	10
2.1. Recopilación de información .....	11
2.2. Análisis de requerimientos de software .....	14
2.2.1. ¿Qué es un sistema operativo? .....	14
2.2.2. Estructura general de los sistemas operativos .....	16
2.2.3. Tipos de sistemas operativos.....	19
2.2.4. Descripción y características de plataformas de sistemas operativos .....	23
2.2.5. Análisis para elección del Data Base Manager System (DBMS) .....	34
2.2.6. Función del Data Base Manager System (DBMS) .....	35
2.2.7. Programas que conforman el DBMS .....	35
2.2.8. Descripción y características de PostgreSQL, MySQL y Oracle .....	36
2.2.9. Descripción de lenguajes utilizados para crear la interfaz (PHP y HTML) .....	42
2.2.10. Servidor Web Apache .....	44
2.2.11. Elección de software .....	45
2.3. Análisis de Requerimientos de Hardware .....	46
2.4. Identificación de Usuarios del sistema .....	48
<b>Capítulo III. Diseño del sistema</b> .....	49
3.1. Diseño del sistema .....	50
3.2. Diagrama de Flujo de Datos .....	51
3.2.1. Simbología .....	52
3.2.2. Diagrama de Flujo de Datos Físico .....	54
3.2.3. Diagrama de Flujo de Datos Lógico .....	54
3.3. Modelo Entidad-Relación .....	59
3.3.1. Modelo de la Base de Datos .....	59
3.3.2. Entidades .....	63
3.3.3. Mapeo del Modelo Entidad Relación .....	64
3.3.4. Normalización .....	65
3.4. Diseño de Base de Datos .....	69

3.4.1. Restricciones de Integridad .....	71
3.4.2. Código para generar la Base de Datos .....	71
3.5. Diccionario de Datos .....	80
<b>Capítulo IV. Desarrollo del sistema .....</b>	<b>87</b>
4.1. Descripción de los procesos .....	88
4.2. Pruebas e implementación .....	96
4.2.1. Implementación .....	96
4.2.2. Propuesta de implementación .....	97
4.3. Pruebas .....	99
4.3.1. Niveles de Pruebas .....	99
4.3.2. Pruebas realizadas al sistema .....	100
<b>Capítulo V. Mantenimiento .....</b>	<b>103</b>
5.1. Revisiones periódicas .....	104
5.1.1. Respaldos de la Base de Datos .....	104
5.1.2. Cambios en la lógica de programación .....	104
5.2. Mantenimiento correctivo, preventivo, adaptativo y perfectivo .....	104
5.2.1. Mantenimiento correctivo .....	104
5.2.2. Mantenimiento preventivo .....	105
5.2.3. Mantenimiento adaptativo .....	105
5.2.4. Mantenimiento perfectivo .....	105
5.3. Nuevas Tecnologías .....	106
5.3.1. Java .....	106
5.3.2. ASP .....	106
5.3.3. Perl .....	107
<b>Capítulo VI. Capacitación .....</b>	<b>108</b>
6.1. Capacitación de los usuarios del sistema .....	109
6.1.1. Sesiones teórico-prácticas .....	109
6.1.2. Documentación .....	109
6.2. Ayuda en línea .....	110
<b>Conclusiones .....</b>	<b>111</b>
<b>Apéndice I (Metodologías) .....</b>	<b>113</b>
<b>Apéndice II (Código de procesos) .....</b>	<b>121</b>
<b>Bibliografía .....</b>	<b>149</b>
<b>Manual del usuario .....</b>	<b>151</b>

# *Introducción*

Respecto a los avances tecnológicos que surgen de forma continua, sobresale el desarrollo de software para la implementación de sitios Web en un medio de comunicación electrónico como Internet, por lo cual es aprovechado por empresas e instituciones con el fin de dar a conocer todo tipo de información, como servicios, productos, noticias, etc., a los miles de usuarios que buscan satisfacer una necesidad.

En la actualidad las Instituciones educativas reconocen la importancia de acceder a estos medios electrónicos con el fin de automatizar la difusión de los servicios de educación continua y especialización, el propósito es que gran parte de la población esté informada a través de su propio sitio Web, ya que actualmente se lleva a cabo en forma impresa.

Tomando como referencia lo anterior, se emprendió la tarea de proponer un sistema de información, cuyo desarrollo tiene la finalidad de ofrecer un servicio útil, aprovechando los conocimientos adquiridos en la Carrera de Computación.

El objetivo del presente trabajo es mostrar el desarrollo del "Sistema de Información automatizado para realizar un seguimiento académico de los alumnos de la División de Ingeniería Mecánica e Industrial". Este sistema tiene como propósito ofrecer una herramienta automática para darles seguimiento a los alumnos que están por titularse tanto a nivel Licenciatura como a nivel de Posgrado.

El desarrollo del sistema se describe conforme avanza el trabajo, desde la necesidad de elaborarlo, hasta su implementación.

En los siguientes párrafos, se expone la estructura general de la tesis.

**Capítulo I.-** Muestra la misión, estructura y aspectos que conforman la División de Ingeniería Mecánica e Industrial (DIMEI), con el propósito de obtener una visión amplia de está, y así elaborar un software de acuerdo a sus necesidades. Siendo este capítulo la base para el desarrollo del proyecto.

**Capítulo II.-** Se hace un análisis tanto de los requerimientos de software como de hardware a fin de que el desarrollo del sistema se lleve a cabo de acuerdo con dichos requerimientos, ventajas y desventajas en su uso, todo esto con el fin de establecer un buen desarrollo del sistema.

**Capítulo III.-** Se lleva a cabo el diseño del sistema y las relaciones de cada uno de los módulos que componen el sistema, por ejemplo, el Diagrama Entidad-Relación, el Diagrama de Flujo de datos, Diccionario de datos etc.

**Capítulo IV.-** Se realiza el programa de aplicación, para controlar la transición y seguridad de los datos en el servidor Web, siendo amigable a los usuarios que tengan acceso a él. Menciona también el tipo de pruebas que se llevaron a cabo al sistema para asegurar y garantizar la calidad del mismo.

**Capítulo V.-** Menciona los tipos de mantenimiento que se le pueden aplicar a un software y la importancia que esto tiene.

**Capítulo VI.-** En este último capítulo se presentan algunas de las técnicas utilizadas para la capacitación de los usuarios que ocuparan el sistema de tal manera que se le saque el mejor provecho.

# *CAPITULO I.*

## Antecedentes

En las siguientes líneas se describen las razones para llevar a cabo la tesis que lleva por nombre **“Sistema de información automatizado para realizar un seguimiento académico de los alumnos de la División de Ingeniería Mecánica e Industrial”**, y que tiene por finalidad resolver una de las actuales necesidades de la División permitiendo una renovación de está en beneficio de sus alumnos.

### 1.1. Misión, estructura y funciones de la DIMEI.

El propósito de la **División de Ingeniería Mecánica e Industrial (DIMEI)** es preparar ingenieros profesionales en las ramas de Ingeniería Mecánica, Industrial, Mecatrónica, Termoenergía y Mejoramiento Ambiental, que tengan conocimientos, habilidades y actitudes para prestar servicios útiles a la sociedad con ética, calidad y productividad, logrando competitividad a través de la tecnología, el diseño y la innovación en materiales, productos, sistemas y procesos que apoyen al desarrollo nacional.

La DIMEI tiene las siguientes funciones:

- Impartir y coordinar académica y administrativamente las carreras de ingeniería mecánica, ingeniería mecatrónica e ingeniería industrial.
- Actualizar los planes y programas de estudio a su cargo y proponer las modificaciones pertinentes, así como la creación de nuevas materias, áreas o carreras, acordes con los avances de la ciencia, la tecnología y con el desarrollo nacional.
- Coordinar y supervisar el cumplimiento de los planes y programas de estudio, así como proporcionar el material de apoyo y las condiciones adecuadas para que dichos planes y programas se lleven a cabo.
- Desarrollar las actividades tendientes a la superación y actualización de su personal docente, que propicien el mejoramiento del proceso de enseñanza-aprendizaje en las asignaturas a su cargo y conforme a las políticas académicas de la Facultad de Ingeniería.
- Mantener y fomentar las relaciones de intercambio con dependencias universitarias, instituciones de educación superior, asociaciones, colegios profesionales y con otras instituciones afines tanto nacionales como extranjeras.
- Promover la realización de conferencias, seminarios, exposiciones, cursos y demás actividades tendientes a la difusión científica y técnica en las disciplinas bajo su responsabilidad.
- Dar asesoría y realizar actividades de investigación tecnológica en las disciplinas a su cargo.
- Establecer y controlar programas de servicio social, titulación, prácticas, estancias y visitas industriales, con el objeto de vincular a la División con la empresa.

- Coordinar las actividades del Consejo Asesor Externo y demás cuerpos colegiados, pertinentes para el desarrollo académico de la División.

Para cumplir con sus objetivos y funciones, la División esta integrada en su administración académica por (ver Figura 1.1.):

- Una Jefatura
- Una Secretaría académica
- Coordinaciones de carrera
- Departamento de Ingeniería Mecánica
- Departamento de Ingeniería Industrial
- Departamento de Mecatrónica
- Departamento de Termoenergía y Mejoramiento ambiental
- Un Centro de manufactura
- Una unidad de investigación y asistencia técnica en materiales
- Una unidad de investigación y desarrollo en Ingeniería Térmica
- Una unidad de asesoría y servicios en Ingeniería Industrial

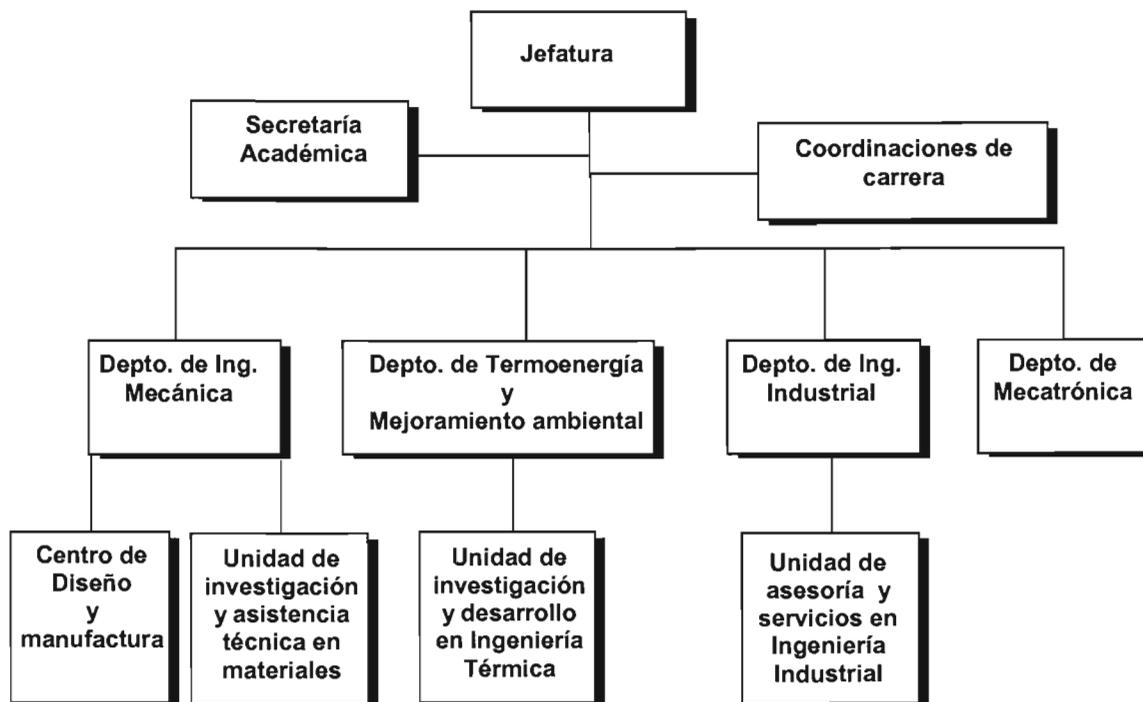


Figura 1.1. Organigrama de la División de Ingeniería Mecánica e Industrial

Cuenta además con coordinaciones que responden a tareas específicas en la División, como son:

- Servicio social e información laboral
- Administración académica
- Titulación, prácticas y visitas
- Vinculación Universidad empresa
- Coordinación de Ingeniería Mecánica
- Coordinación de Ingeniería Industrial
- Coordinación de Ingeniería Mecatrónica

## 1.2. Problemática

**La División de Ingeniería Mecánica e Ingeniería Industrial (DIMEI) de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México**, se estableció con el objetivo de ofrecer un nivel superior a los estudios de licenciatura y dada la nueva reestructuración del posgrado también tendrá dentro de sus funciones principales la de formar especialistas, maestros y doctores de alto nivel, con calidad y excelencia en las diversas ramas de la ingeniería, a través de los diferentes departamentos, mediante actividades académicas y de investigación vinculadas a la innovación científica y tecnológica, a fin de contribuir a la transformación que en los ámbitos educativos, social, cultural y económico exige el país.

En la actualidad **la DIMEI**, al igual que todas las demás Divisiones que conforman la Facultad, enfrentan un bajo índice de graduados ya que únicamente obtienen el grado alrededor del 30% de los alumnos que ingresan por generación.

Como sabemos para poder recibir un título o grado universitario es un requisito presentar un trabajo de investigación (tesis). Para ello cada alumno al estar por concluir sus materias o al haberlas concluido, debe participar en un proyecto de tesis para poder obtener el título de su respectiva carrera o nivel de estudios, entre los que se encuentran contemplados:

- Licenciaturas
- Especialidades
- Maestrías
- Doctorados

Esto haría suponer que por cada generación que ingresa después de cierto tiempo debería de haber igual número de alumnos que estén por titularse. Sin embargo esto no es así, esto puede deberse a varias causas entre las que podemos mencionar la deserción, atraso en la terminación de materias, etc. Lo que ya reduce el porcentaje de alumnos por titularse. Pero existe otro problema, ¿que pasa con la gente que ya habiendo registrado una tesis no puede titularse en el tiempo reglamentado? Por ejemplo, de acuerdo a la información recaudada por la Coordinación de Titulación de la DIMEI en 1996 ellos tuvieron un ingreso de 246 alumnos de los cuales solo se han graduado 60 alumnos, los cuales representan tan solo un 24%. A esto agreguemos que algunos de ellos se titularon mucho tiempo después de haber terminado las materias. Esto lleva a plantear ¿Por qué los alumnos se gradúan tanto tiempo después? ¿Qué hacer para que se gradúen a tiempo?

Las razones para que un alumno no se gradúe en el tiempo reglamentario pueden ser muchas, complejidad del propio tema, falta de información, falta de tiempo, presupuesto, problemas personales, trabajo etc.

También sabemos que en la elaboración de una tesis intervienen directamente tanto los alumnos como los directores. El director es la persona que asesora al alumno durante la elaboración de su proyecto. Es un docente capaz de orientar la investigación del estudiante dentro de un contexto académico.

Como se puede observar los directores juegan un papel muy importante en el desarrollo de las tesis, por lo que debe de existir una condición de balance para atender sus propias actividades y por otro lado atender a los tesisistas. Es importante señalar esto, porque hay tutores que tienen mayor productividad, mientras que la productividad de otros es menor, lo que repercute en el tiempo de titulación de los alumnos. Hay casos de directores en los que sus tesisistas llevan varios períodos realizando su tesis, lo que conlleva a que los directores acumulen una gran cantidad de trabajos no finalizados. Por lo que para nuestra investigación se plantean las siguientes interrogantes:

¿Quiénes son los profesores con mayor cantidad de tesis asesoradas y aprobadas?

¿Quiénes son los profesores que las concluyen dentro del tiempo reglamentado?

La DIMEI preocupada por esta situación comenzó a recaudar información importante de cada alumno que se titula, para que dicha información le permitiera hacer un análisis del comportamiento de los alumnos que se titulan, dentro de la información que se consideró importante recaudar se encuentra la siguiente:

- Número de Cuenta del Alumno
- Nombre
- Nacionalidad
- Carrera
- Año de Ingreso
- Año de Egreso
- Fecha de Examen de Titulación
- Promedio (licenciatura)

Al analizar dicha información, se puede ver que existen alumnos que se titulan dentro del tiempo reglamentado y muchos otros que se titulan fuera de tiempo. Ahora bien la importancia de esta información radica en lo que se puede hacer para ayudar a que los alumnos se titulen en el tiempo establecido.

### 1.3. Objetivo

El objetivo principal es diseñar y desarrollar un sistema de información automatizado que nos permita realizar un seguimiento académico exhaustivo a los alumnos de la **División de Ingeniería Mecánica e Industrial de la Facultad de Ingeniería**, y con base en él, determinar que acciones se deben llevar a cabo con los alumnos que aún no se gradúen y que puedan concluir sus estudios en el tiempo reglamentario.

### 1.4. Alcances y ámbito del sistema

Al desarrollar el sistema se pretende seguir más de cerca a los alumnos que tiene registrada una tesis, verificando cada determinado tiempo sus avances, y en caso de observar un estancamiento hablar con el alumno o director para saber la razón de dicha situación.

Este seguimiento permitirá que en caso de que el alumno tenga algún problema para seguir avanzando es su tesis, la gente a cargo junto con el alumno planee las actividades y determinen un plan de trabajo, que le permita seguir avanzando para concluir.

Toda la información necesaria para dar seguimiento a los alumnos estará concentrada en una Base de Datos. La Base de Datos facilitará el control, la búsqueda y recuperación de información acerca

de las tesis, alumnos, tiempos de finalización, etc. Dicha información podrá ser consultada a través de Internet por los alumnos, profesores y por la Coordinación de Titulación de la DIMEI.

Con la operación de este sistema, se tendrán los siguientes resultados:

1. Registrar las tesis que se estén elaborando en la DIMEI y llevar un control de éstas.
2. Tener información de los alumnos que están inscritos en un proyecto de tesis.
3. Contar con información de los profesores que están dirigiendo tesis o hayan dirigido tesis.
4. Generar reportes del avance de las tesis.
5. Generar reportes de tesis que ya se hayan concluido.
6. Verificar cuantas tesis a dirigido un profesor y en cuanto tiempo logra su conclusión.

# *CAPITULO II.*

## Determinación de los requerimientos del sistema

El análisis es el proceso de recolectar los requerimientos de los usuarios, por ello cuando se desarrolla un sistema se tiene que analizar bien el problema y los objetivos a seguir, para así poder elegir las herramientas adecuadas, con el fin de cubrir las necesidades de los usuarios.

El análisis y especificación de requerimientos, puede parecer una tarea relativamente sencilla, sin embargo, es la tarea que plantea la asignación de software al nivel de sistema y del diseño de programas. El análisis de requerimientos facilita especificar la función y comportamiento de los programas, relacionar la interfaz con otros elementos del sistema y establecer las relaciones que deben cumplir los programas ó el programa. También permite establecer las características que deberá tener el hardware; ya que de dichos requerimientos de hardware dependerán de la ejecución del sistema.

Dado que en el análisis se hace todo lo posible para identificar qué información requieren los usuarios para desempeñar sus tareas, el primer paso es analizar lo que se está haciendo actualmente y en que forma, así como las necesidades adicionales que han sido solicitadas. De esta manera se puede obtener una mayor comprensión de las funciones actualmente realizadas para manejar la información y de hacia a donde se debe dirigir el nuevo sistema.

Como se puede ver las actividades y eventos que comprenden el análisis del sistema se dirigen en su mayor parte en responder a la pregunta **¿qué va a incluir el sistema?** Al responder esta pregunta general se deben plantear muchas preguntas específicas como:

- ¿Qué información se necesita?
- ¿Quién la necesita?
- ¿En qué forma?
- ¿Cómo puede obtenerse?

En general para determinar y/o especificar cualquier tipo de requerimiento, pueden llevarse a cabo diferentes actividades como son: **el estudio de los datos, la entrevista, los cuestionarios, etc.**

Una vez que han realizadas estas actividades y se ha determinado que deberá realizarse se puede comenzar la fase de diseño.

## **2.1. Recopilación de Información**

Como se menciona con anterioridad, el propósito de este trabajo es el de promover un mejor manejo de la información referente a los **proyectos de tesis en la DIMEI**, esto a través de una Base de Datos que será consultada vía Internet. En dicha Base de Datos se encontrarán alojados todos los datos e información correspondiente a alumnos, profesores y proyectos de tesis.

Por otra parte es conocido que una Base de Datos refleja la organización y flujo de la información en una institución y al iniciar el análisis del diseño del sistema primero se debe averiguar que tipo de información se manejará dentro del sistema.

Para identificar el tipo de información a procesar se llevaron a cabo una serie de entrevistas con el responsable de la **Coordinación de Titulación**.

A continuación se enlista y cataloga la información, que después del análisis realizado se consideró importante para formar parte de la Base de Datos.

### **Información de Profesores**

- Nombre completo (apellido paterno, apellido materno y nombre)
- Nacionalidad
- Domicilio (calle, número, colonia, delegación, estado y C.P.)
- Teléfono
- RFC, que acredita la identidad de las personas.
- Correo electrónico, nos brinda la ventaja de contar con un medio de comunicación directo, rápido y eficiente.

El sistema también almacenará los datos académicos del profesor como:

- Nivel de Estudios (licenciatura o posgrado)
- Nombramiento
- Tipo de profesor

### **Información de Alumnos**

La recopilación de información de los alumnos fue dividida en dos partes, datos personales y datos académicos, logrando de esta manera la mejor utilización y administración de dicha información, a continuación se muestra el tipo de información que se recopiló.

#### **Datos personales**

- RFC
- Nombre completo (apellido paterno, apellido materno y nombre)
- Nacionalidad
- Domicilio (calle, número, colonia, delegación, estado y C.P.)
- Teléfono
- Correo electrónico

#### **Datos académicos**

- Número de cuenta, que acredita la identidad del alumno
- Nivel de Estudios (licenciatura o posgrado)
- Carrera de procedencia
- Plantel donde realizó sus estudios
- Fecha de ingreso (licenciatura o posgrado)
- Fecha de egreso (licenciatura o posgrado)
- Avance en créditos (licenciatura o posgrado)
- Promedio (licenciatura)
- Si se encuentra becado

### Información de proyectos (tesis o seminario)

La información que se recopiló para los proyectos fue la siguiente:

- Tipo de proyecto (tesis o seminario)
- Clave
- Departamento
- Área
- Título del proyecto
- Nivel (licenciatura o posgrado)
- Descripción
- Objetivo
- Temario
- Campo de investigación
- Tipo de investigación
- Número de alumnos inscritos en el proyecto
- Avances del proyecto
- Fecha de registro
- Fecha de inicio
- Financiamiento (si es que existe)

### Información de jefes de departamento

Los jefes de departamento contarán con un login y un passwords único, el cual le dará acceso al sistema. El sistema deberá proporcionar una forma a través de la cual podrá consultar la información alojada en la Base de Datos. La información recopilada para este tipo de usuario es la siguiente:

- Nombre
- RFC
- Departamento del cual es responsable
- Login
- Password

## Información de los departamentos

- Clave
- Nombre del departamento
- Teléfono
- Nombre del jefe de departamento

Esta información recopilada es de gran utilidad ya que en base a ella se diseñará el sistema, permitiendo que de esta forma el sistema procese información de interés para la **Coordinación de Titulación** y está a su vez ofrece información importante para sus usuarios, como por ejemplo el avance registrado en un proyecto de tesis.

## 2.2. Análisis de requerimientos de software

En este punto se darán a conocer los requerimientos de software que será utilizado para el desarrollo del sistema, así como sus especificaciones y definición. Se analizarán las principales propiedades del software propuesto y los beneficios que cada uno de ellos ofrece.

Para el desarrollo del sistema se requiere de:

- Un sistema operativo
- Un manejador de base de datos
- Un lenguaje de programación para la interfaz

De acuerdo a lo anterior es necesario saber que es y que hace cada uno de estos.

### 2.2.1. ¿Qué es un sistema operativo?

Existen diversas definiciones de lo que es un sistema operativo por lo que a continuación se presentan algunas de las tantas definiciones que existen:

- Se puede definir un sistema operativo como un conjunto de programas, instalados que hacen utilizable el hardware. El hardware proporciona la "capacidad del equipo de cómputo" y los

sistemas operativos ponen dicha capacidad de cómputo al alcance de los usuarios, administrando cuidadosamente el hardware para lograr un buen rendimiento.

- Los sistemas operativos son ante todo administradores de recursos, el principal recurso que administran es el hardware de la computadora; además de los procesadores, los medios de almacenamiento, los dispositivos de entrada/salida, los dispositivos de comunicación y los datos.
- Un sistema operativo es un programa que actúa como intermediario entre el usuario y el hardware de la computadora y su propósito es proporcionar el entorno en el cual el usuario puede ejecutar programas. El objetivo principal de un sistema operativo es que el hardware de la computadora se emplee de manera eficiente y el objetivo secundario es lograr que el sistema sea lo más sencillo de manejar por el usuario.
- Un sistema operativo es el conjunto de programas que controla la ejecución de programas de aplicación y actúa como una interfaz entre el usuario y el hardware de la computadora, esto es, un sistema operativo explota y administra los recursos de hardware de la computadora con el objetivo de proporcionar un conjunto de servicios a los usuarios del sistema.

Basándonos en las definiciones anteriores podemos decir en resumen, que el sistema operativo es un conjunto de programas, que están encargados de brindar al usuario una forma agradable y sencilla de operar, interpretar, codificar y emitir las ordenes al procesador central para que este realice las tareas necesarias y específicas para completar una orden.

Por lo tanto un sistema operativo hace de una computadora un instrumento útil que tiene dos funciones primordiales:

- Administrar el hardware, es decir, distribuir de forma más eficiente los recursos de un equipo.
- Facilitar el trabajo al usuario, permitiendo una comunicación cómoda y sencilla con los dispositivos de la máquina.

## 2.2.2. Estructura general de los sistemas operativos

Si bien no todos los sistemas operativos tienen la misma estructura; los actuales poseen las siguientes características:

- Administrador de procesos
- Administrador de memoria
- Administrador de almacenamiento secundario
- Administrador de sistemas de Entrada/Salida
- Administrador de archivos
- Sistema de protección
- Sistema intérprete de comandos

A continuación se hace una descripción de cada uno de ellos:

### Administrador de procesos

Uno de los módulos más importantes de un sistema operativo es la del administrador de los procesos, que tiene la tarea de planificar al procesador, es decir asigna tiempo de ejecución, memoria y dispositivos de Entrada/Salida a los procesos del sistema operativo. Estos recursos son reservados cuando se crea el proceso o bien se otorga en tiempo de ejecución.

El administrador de procesos es responsable de:

- La creación y eliminación de procesos de sistema y de usuario.
- Detener y continuar ejecutando un proceso.
- Proveer de mecanismos para sincronizar procesos.
- Proveer de mecanismos para comunicar procesos.
- Proveer de mecanismos para proteger procesos.

### Administrador de memoria

Este módulo se encarga de asignar ciertas porciones de la memoria principal (RAM) a los diferentes programas o partes de los programas que la necesiten, mientras el resto de los datos y los programas se mantienen en los dispositivos de almacenamiento masivo. De este modo, cuando se

asigna una parte de la memoria principal se hace de una forma estructurada, siguiendo un determinado orden. Por ejemplo para que la CPU procese datos del disco, primero se deben cargar éstos en la memoria.

El administrador de memoria es responsable de:

- Mantener un historial de las partes de memoria que pueden ser accedidas concurrentemente y que procesos pueden hacerlo.
- Decir que procesos se cargarán en memoria cuando se encuentre lugar disponible en ésta.
- Asignar y limpiar espacios de memoria según se necesite.

#### **Administrador de almacenamiento secundario**

Como la memoria principal es muy chica como para almacenar todos los datos y programas necesarios, la computadora posee un almacenamiento secundario para guardar los datos no utilizados en ese momento pero son necesarios para el usuario. Las computadoras modernas utilizan un disco para ese fin. La mayoría de los programas son almacenados en disco hasta que son cargados en memoria.

El administrador de memoria secundaria es responsable de:

- Administrar el espacio libre.
- Asignar espacio de almacenamiento.
- Organizar los datos en el disco.

#### **Administrador de sistemas de Entrada/Salida**

Este módulo presenta al usuario la Entrada/Salida de datos como una cuestión independiente de los dispositivos; es decir, para los usuarios todos los dispositivos tienen las mismas características y son tratados de la misma forma, siendo el sistema operativo el encargado de atender las particularidades de cada uno de ellos (como su velocidad de operación).

El administrador de Entrada/Salida está compuesto por:

- Un sistema de buffer intermedio.
- Una interfaz general.
- Manejadores de dispositivos de hardware específicos.

el usuario teclea a instrucciones que el mismo sistema operativo reconoce. Algunos sistemas operativos incluye el intérprete de comandos en el Kernel<sup>2</sup>.

El intérprete de comandos es conceptualmente un programa que se ha cargado del disco en la memoria de la computadora y cuya misión es el interpretar y ejecutar las órdenes que el usuario teclea en su terminal<sup>3</sup>.

Los sistemas operativos tienen diferencias en el área de interpretación de comandos, según el grado de facilidad que presentan a los usuarios. Por ejemplo en Windows 2000 para copiar un archivo de una unidad a otra el usuario puede seleccionar con el mouse el archivo que desea copiar y arrastrarlo hasta su nuevo destino; mientras que en DOS, debe ingresar una sentencia desde la terminal. Es el mismo caso de Linux, cuyo intérprete de comandos se encuentra en cada terminal.

### 2.2.3. Tipos de sistemas operativos

#### Sistemas operativos por lotes

Los sistemas operativos por lotes, procesan una gran cantidad de trabajos con poca o ninguna interacción entre los usuarios y los programas en ejecución. Se reúnen todos los trabajos comunes para realizarlos al mismo tiempo, evitando la espera de dos o más trabajos como sucede en el procesamiento en serie.

Pero aún existía un problema, cuando un programa detenía su ejecución (terminación normal o anormal), el operador tenía que darse cuenta observando la consola, determinar por que se había detenido el programa, realizar un volcado a memoria si era necesario, cargar la lectora de cintas con el siguiente trabajo y poner de nuevo en marcha la computadora. Durante esta transición la CPU permanecía inactiva. Para eliminar el tiempo de inactividad se introdujo secuenciamiento automático de trabajos y con el se crearon los primeros y rudimentarios sistemas operativos. Lo que se deseaba era un procesamiento para la transferencia automática del control de un trabajo al siguiente. Con este propósito se creó un pequeño programa llamado *monitor residente*. Este programa se encuentra siempre en memoria.

---

<sup>2</sup> El Kernel es considerado el núcleo o parte fundamental de un Sistema Operativo. La parte que se carga inicialmente y permanece en memoria, encargada de proporcionar los servicios básicos de gestión de memoria, ficheros y recursos.

<sup>3</sup> Una terminal es un punto de introducción y salida de datos de una computadora central. Se les suele denominar "tontas" si carecen de procesador, es decir, dependen por completo del servidor, e "inteligentes" en caso contrario.

Al iniciar la computadora el programa residente se cargaba y controlaba la misma, transfiriendo el control a cada programa. Cuando el programa se colgaba, el monitor residente tomaba nuevamente el control y continuaba con el siguiente programa. Para que el monitor residente supiera que programas tenía que ejecutar se le indicaba una breve descripción de que programas tenía que ejecutar y con que datos. Esta información era suministrada por el operador a través de tarjetas de control. Por lo tanto el programa residente realizaba el secuenciamiento automático de trabajos según le indicaban las tarjetas de control.

### **Sistema operativo multiprogramado**

Se distinguen por sus habilidades para poder soportar la ejecución de dos o más trabajos activos (que se están ejecutando) al mismo tiempo. Esto trae como resultado que la Unidad Central de Procesamiento (CPU) siempre tenga alguna tarea que ejecutar, aprovechando al máximo su utilización.

Su objetivo es tener a varias tareas en la memoria principal, de manera que cada uno está usando el procesador, o un procesador distinto, es decir, involucra máquinas con más de una CPU.

Un sistema operativo multiprogramado es bastante sofisticado. Tener varios trabajos para ejecutar significa tenerlos simultáneamente en memoria. Tener varios programas en memoria requiere una gestión de memoria, además el sistema operativo deberá seleccionar uno de ellos. Esto se denomina planificación de CPU.

### **Sistema operativo de tiempo compartido**

Este sistema operativo se caracteriza por simular que el sistema y sus recursos son todos para cada usuario. El usuario hace una petición a la computadora, esta la procesa tan pronto como le es posible y la respuesta aparecerá en la computadora del usuario.

Los principales recursos que comúnmente son utilizados entre los usuarios son: el sistema, el procesador, la memoria, dispositivos de entrada/salida, dando a cada usuario la ilusión de que tiene el sistema dedicado para sí mismo. Esto trae como consecuencia una gran carga de trabajo al sistema operativo, principalmente en la administración de memoria principal y secundaria.

Los sistemas de tiempo compartido se caracterizan por la falta de interacción entre el usuario y la computadora. El trabajo se prepara y se deja. Al cabo de un cierto tiempo aparece la salida.

Un sistema operativo en tiempo compartido utiliza la planificación de CPU y la multiprogramación para dotar a cada usuario de una pequeña parte de la computadora. En lugar de dejar que la CPU permanezca inactiva, cuando sucede esto, el sistema operativo cede el control de la CPU al programa de otro usuario.

### **Sistema operativo multiprocesador**

Un sistema multiprocesador se reconoce por tener más de un CPU compartiendo memoria y periféricos. Las ventajas de este sistema son una mayor potencia de computación y fiabilidad.

Estos sistemas asignan una tarea específica a cada procesador. Un procesador maestro controla el sistema, los otros esperan al maestro o tienen tareas predefinidas. Este esquema define una relación maestro/esclavo. Los sistemas de tiempo compartido se componen generalmente de una computadora grande (principal) y una computadora más pequeña que solo es responsable de las entradas/salidas sobre terminales.

Podemos clasificar en sistemas multiprocesadores *simétricos*, en los cuales cada procesador tiene asignado una tarea predefinida.

### **Sistema operativo distribuido**

La principal diferencia entre los sistemas de multiprocesadores y distribuidos, es que estos últimos no comparten memoria y periféricos. Cada procesador tiene su propia memoria local. Los procesadores se comunican entre sí a través de varias líneas de comunicación, tales como líneas telefónicas o buses de alta velocidad.

### **Sistema operativo de tiempo real**

Se utiliza cuando hay requerimientos de tiempo muy rígido en las operaciones o en el flujo de datos, generalmente se utilizan como sistemas de control en una aplicación dedicada. Por ejemplo, sistemas de control industrial, experimentos científicos y sistemas médicos, así como en sistemas de defensa y en aviones caza.

Se pueden clasificar en:

Los sistemas de tiempo real por *hard*, que garantizan que las tareas críticas se ejecutan en tiempo. Los datos son almacenados en memorias no volátiles (ROM).

Los sistemas de tiempo real por *soft*, ejecutan tareas menos críticas y manejan técnicas de prioridades para la asignación de tareas.

### **Sistema operativo monotarea**

El sistema monotarea permite solamente la ejecución de un programa a la vez. Por esta causa, los nuevos sistemas operativos se diseñan para permitir el uso de programas múltiples simultáneamente.

### **Sistema operativo multitarea**

En este tipo de sistemas operativo los programas van turnándose en el procesador, permitiendo así que un usuario tenga varios programas ejecutándose.

### **Sistema operativo multiusuario**

Este tipo de sistemas operativos nos permite tener una serie de usuarios que pueden encontrarse utilizando los mismos recursos de una máquina. Esto se denomina "*time-sharing*" o tiempo compartido. Este tipo de sistemas se emplean especialmente en donde se encuentran instaladas redes; ya que cumplen las necesidades de dos o más usuarios, que comparten mismos recursos.

### **Sistema operativo monousuario**

Los sistemas monousuario son aquellos que nada más pueden atender a un solo usuario, gracias a las limitaciones creadas por el hardware, los programas o el tipo de aplicación que se este ejecutando.

Este tipo de sistemas son muy simples, por que todos los dispositivos de entrada, salida y control dependen de la tarea que se esta utilizando, esto quiere decir, que las instrucciones que se dan, son procesadas de inmediato; ya que existe un solo usuario.

## 2.2.4. Descripción y características de plataformas de sistemas operativos

### Características del sistema operativo Linux

Linux es un sistema operativo gratuito enfocado a las redes y perfeccionado para Internet, que está basado en el sistema operativo Unix, de código abierto y que puede trabajar bajo distintos tipos de hardware. Mucho se dice que Linux es libre y en ocasiones gratuito. Es gratuito por que puede obtenerse sin costo alguno, si se tiene acceso a Internet y descargarlo sin mucho trámite. Existen sin embargo empresas que venden Linux, pero lo que venden es únicamente el disco y sus manuales, para poder seguir costearo sus investigaciones y desarrollos. Es libre en el sentido de que ofrece toda la libertad de manipularlo a criterio propio, ya que cuando uno descarga Linux, se está descargando algo más que el propio sistema operativo, se obtiene el código fuente, de tal forma que se puede adecuar a las necesidades personales modificando dicho código fuente para obtener los resultados que se desee.

Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers<sup>4</sup>, las herramientas de desarrollo y todos los programas de usuario; además de todo, estos pueden distribuirse libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux, actualmente sin código.

Linux, es un sistema operativo multitarea real, que utiliza la llamada multitarea preventiva, que básicamente asegura que todos los programas que se están utilizando en un momento dado sean ejecutados. En caso de que varias personas requieran hacer una o varias consultas teniendo que entrar al sistema, este tomará cada consulta como una tarea y dará a cada una el tiempo necesario para que finalice satisfactoriamente. El sistema operativo es el encargado de ceder tiempo de microprocesador a cada tarea programada.

En lo referente a multiusuario nos es de gran ventaja manejar un sistema Linux, ya que de esta forma podremos garantizar un servicio a varios usuarios una vez conectados a una misma máquina y que puedan trabajar al mismo tiempo.

Entre las características sobresalientes de Linux es que es multiplataforma, logrando que corra en casi cualquier equipo. Las plataformas en las que en un principio se puede utilizar Linux son 386, 486,

---

<sup>4</sup> **drivers**: Existen muchos periféricos que se pueden conectar a una computadora (unidades de 3 1/2, impresoras, lectores de CD, escáner, etc.). Para que el sistema sea capaz de aprovechar al máximo las capacidades de cada uno de estos dispositivos, los fabricantes incluyen unos programas llamados "Drivers", que son los que saben gestionar adecuadamente ese periférico.

todos los Pentium de Intel, AMD y Cyrix, Amiga y Atari, aunque también existen varias versiones para su utilización en otras plataformas, como Alpha, ARM, PowerPC y SPARC.

Es posible la escalabilidad de procesadores, ya que también es un sistema multiprocesador. Tiene soporte para sistemas con más de un procesador pero solo esta disponible para tecnologías Intel y SPARC.

Otras características importantes es su protección de la memoria entre procesos, de manera que evite que alguno de estos pueda "trabar" el sistema (otros sistemas que no han logrado solucionar este problema sufren continuos volcados de memoria). Se tiene una carga de ejecutables por demanda y política de copia en escritura para la comparación de páginas entre ejecutables. Con lo primero, Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente. Con esta política aseguramos que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

Posee una excelente memoria virtual que usa la paginación (sin intercambio de procesos completos) a disco. Da la ventaja de que esta puede ser una partición o archivo en el sistema de archivos, o ambas, con la posibilidad de añadir más áreas de intercambio sobre la marcha. En estos momentos se sabe de un total de 16 zonas de intercambio de 256 Mb de tamaño máximo que pueden ser usadas en un momento dado con un límite teórico de 4 Gb para intercambio. Este límite se puede aumentar en caso necesario de unas cuantas líneas en el código fuente.

La ventaja del manejo de memoria que se lleva a cabo, es que está se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de manera que toda la memoria libre puede ser utilizada para caché o reducida en caso de que algún usuario ejecute grandes programas.

Linux cuenta con muchos lenguajes de programación, compiladores y herramientas de desarrollo asociadas como son:

- ADA
- Basic
- C
- C++
- Expect, un lenguaje de scripts para automatizar sesiones de red
- FORTRAN
- GTK, un conjunto de herramientas para crear aplicaciones GUI en Linux

- PASCAL
- Pitón, un lenguaje de scripts orientado a objetos
- Lenguajes de shell (csh, bash, ksh y sh)
- TCL/TK un lenguaje de scripts y un conjunto de herramientas con interfaz grafica de usuario
- Perl
- Librerías compartidas de carga dinámica (DLL`s) y librerías estáticas

Es un sistema operativo con una capacidad de llevar a cabo volcados de estado, que permiten llevar un análisis post-mortem de lo sucedido, de esta forma se puede hacer un análisis de lo ocurrido utilizando depuradores sobre los programas que fueron abortados y hasta de aquellos que se encuentran en ejecución.

Posee una emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá tener procesador matemático. Por su puesto, si el ordenador ya tiene una FPU (unidad de coma flotante), esta será usada en el lugar de la emulación, pudiendo incluso compilar su propio Kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.

Posee consolas virtuales múltiples, es decir que se puede tener varias sesiones de Login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (sin que intervenga con el hardware de video).

Es capaz de soportar varios sistemas de archivos comunes, incluyendo minix-1, Xenix y todos los sistemas de archivos típicos de System V, posee un avanzado sistema de archivos propios con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud. No es necesario definirle una extensión a estos archivos para que posean ciertas características particulares, logrando tener cualquier nombre que se desee.

Es posible un acceso transparente MS-DOS (o particiones OS/FAT) mediante un sistema de archivos especiales, logrando que está parezca un sistema de archivos normal de Unix (Excepto por algunas restricciones en los nombres de archivos, así como de sus permisos). El soporte para VFAT, FAT32 (WNT, Windows 95/98) se encuentra soportado desde la versión 2.0 del núcleo y el NTFS de WNT desde la versión 2.2 (Este último solo en modo lectura).

Es posible instalar Linux en un equipo que contenga MS-DOS, esto en base a un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.

Linux es una excelente plataforma para servidores de una Intranet o de Internet, ya que ofrece una óptima potencia a las redes, proporcionando clientes y servidores para todos los protocolos esenciales, entre los que se incluyen:

- FTP (Protocolo de transferencia de archivos)
- Protocolo Gopher
- http (Protocolo de transferencia de Hipertexto)
- NNTP (Protocolo de transferencia de Noticias en Red)
- POP (Protocolo de Oficina de correos)
- PPP (Protocolo punto a punto)
- SLIP (Protocolo Simple de Transferencia de Correo)
- Protocolo Telnet
- TCP (Protocolo de control de Transmisión)

Es posible llevar un control de acceso a redes, es decir, cuenta con la capacidad par permitir a determinados usuarios y hosts conectarse entre sí. De gran ayuda cuando el sistema Linux es un servidor de Internet, ya que proporciona una gran variedad de mecanismos de cifrado para proteger los datos que circulan en una red local o de Internet.

Ya que es posible saber cuando se recibirá un ataque a nuestro host, Linux lleva a cabo un registro de los movimientos de las personas que realizan dichos ataques. Linux detectará y marcará la hora, grabando las conexiones de red que se llevaron a cabo. Esta información se redirige a archivos especiales para su posterior revisión. Sabiendo que existen un gran número de metodologías de ataque distintas, Linux lleva registros a nivel de red, de host y de usuario. Registrando todos los mensajes del núcleo y del sistema. Las conexiones de red, la dirección IP de la que partió la conexión, su longitud y en algunos casos el Login de la persona que llevo a cabo el ataque, además de identificar a que archivo tuvieron acceso. Lleva un registro de todos y cada uno de los comandos que ha emitido cada usuario y que procesos se encuentran bajo su control.

## Requerimientos del sistema operativo Linux

Los siguientes requerimientos son para una instalación óptima:

- Procesador Pentium i486 y cualquier clónico de Intel, como mínimo un Intel 80386.
- RAM de 32 Mb a 2 Gb, soporta hasta 8 Mb como mínimo.
- Tarjeta gráfica SVGA o tarjeta soportada por las XFree86, como mínimo una VGA.
- Disco duro de 500 Mb o más, mínimo 125 Mb.
- Bus de cualquier tipo.
- Lector CD-ROM se recomienda velocidad de X8 o mayor, mínimo X2.
- Mouse Microsoft, Logitech o compatible.
- Tarjeta de red, cualquiera soportada o puede trabajar sin esta.
- Modem, todos los modem's externos o aquellos que no sean win- modem.

Para una instalación mínima se requiere de lo siguiente:

- Un procesador Intel 80386
- 10 Mb de memoria RAM
- Si se desea una instalación mínima con ventanas gráficas, es necesario alrededor de los 80 Mb.

Para una instalación completa se necesita de lo siguiente:

- Si lo que se esta buscando es tener instalado un sistema Linux completo que sea funcional, habrá que tener de 900 Mb a 1.5 Gb de espacio en el disco duro.
- Como mínimo se necesita 4 Mb de memoria RAM, esta cantidad como se indica es mínima y no apta para trabajar con sistemas gráficos.
- Para poder trabajar con las XWindows de una manera óptima se necesitará como mínimo 32 Mb y dependerá del entorno gráfico utilizado.

## Ventajas de Linux

- Precio
- Estabilidad
- Seguridad, sobre todo a nivel servidor

- Compatibilidad, puesto que reconoce la mayoría de los sistemas operativos en la red.
- Velocidad
- El usuario posee el código fuente por lo que puede adecuarlo a sus necesidades
- Se puede usar en cualquier computadora
- Multitarea real
- Manejo de múltiples procesadores (hasta 16 procesadores)
- No se ve afectado por los virus
- Maneja discos duros de hasta 16 terabytes
- Fiabilidad
- Robustez
- Adaptabilidad
- No hay restricciones legales para su uso
- Es pequeño comparado con otros sistemas operativos (puede ejecutarse en 150 Mb)
- Posee memoria virtual, que permite al sistema mantener los procesos activos en el disco duro.

#### **Desventajas de Linux**

- No cuenta con una empresa que lo respalde.
- Requiere de un número mayor de conocimientos por parte del usuario.
- Algunos dispositivos de última generación no están soportados.
- Es un poco más difícil de instalar que los Sistemas Operativos de Microsoft.

#### **Características del sistema operativo Windows 2000 Server**

Esta nueva versión del sistema operativo Windows 2000 Server nos proporciona una gran cantidad de herramientas de conectividad, cambios en la interfaz, buen reconocimiento de hardware y estabilidad. Se añade a esto el soporte de nuevas tecnologías, las mejoras en sus funciones de informática remota, aplicaciones centralizadas y de servicio, que en conjunto crea un mejor uso de la computadora.

Esta nueva versión de Windows 2000 Server está destinada a dar uno de los mejores servicios para servidores de impresión, archivos y aplicaciones de Web de una empresa de pequeña a mediana. Es un sistema con fiabilidad y capacidad para gestionar, que ha sido mejorado con herramientas que ayudaran a los usuarios y administradores de una red a administrar de forma más sencilla los

movimientos y actividades que se han llevado a cabo. La gestión global del sistema la realiza a través de un modulo denominado Administración del equipo, que organiza los recursos, servicios, dispositivos de almacenamiento y seguridad que utilizan tanto en el sistema local como en ordenadores remotos.

En herramientas del sistema dispone de un visor de sucesos y del administrador de dispositivos que proporcionan una síntesis jerarquizada de los dispositivos instalados en la PC y que permiten hacer modificaciones y búsquedas para resolver conflictos IRQ<sup>5</sup> o DMA<sup>6</sup>. Por otro lado, desde el almacenamiento es posible acceder a las propiedades de las unidades del disco, incluyendo unidades flexibles y a sus opciones de verificación, particiones y copias de seguridad. Finalmente servicios y aplicaciones nos dan información más clara sobre los servicios de Microsoft. En general, el Administrador del equipo es un mapa completo y detallado de la PC, incluyendo informes sobre la forma en que el usuario lo utiliza.

Dado que Windows 2000 Server es un sistema operativo enfocado al trabajo en red proporciona una de las mejores opciones cuando se trata de aplicaciones para Internet, pues ofrece una alta seguridad en cuanto a conexiones con los clientes se refiere y a la accesibilidad de información, pero para ello también requiere de seguridad por lo que cuenta con seguridad de tipo local, corporativa y pública.

La seguridad local se refiere a la protección de datos en la computadora, por lo que el sistema esta diseñado para evitar que usuarios no autorizados se "salten" el sistema de arranque y, por tanto, también las funciones de seguridad.

La seguridad corporativa es la protección de datos en una red local y para ello Windows 2000 Server utiliza el protocolo de autenticación Kerberos<sup>7</sup>, que es un estándar de seguridad en redes locales e intranets que verifica y hace un seguimiento de las actividades de cada usuario dentro de la red. Permitiendo un control del acceso unificado a casi cualquier entorno de la red, eliminando la necesidad

---

<sup>5</sup> (Interrupt Request) Petición de interrupción. Este recurso es utilizado para dar prioridad a unos periféricos respecto a otros en el uso del procesador. Como los sistemas operativos modernos son multiproceso, el procesador del sistema esta casi siempre ocupado en diferentes tareas. Los niveles IRQ legitiman a los periféricos para poder interrumpir las tareas que realiza el procesador imponiendo las suyas propias. Además al estar jerarquizadas, si dos elementos solicitan acceso al mismo tiempo, el que tenga mayor prioridad será el que consiga el uso de un procesador.

<sup>6</sup> (Direct Memory Acces), Acceso directo a memoria. La filosofía de DMA antiguamente era de acceder a memoria sin intervención del procesador central para transferir datos mientras el bus no esta siendo usado.

<sup>7</sup> Kerberos es un nuevo protocolo básico de seguridad, seleccionado por Microsoft para Windows 2000 Server.

de obtener permisos y esperar la respuesta cada vez que un cliente desea acceder a un nuevo recurso en la red.

La seguridad pública utiliza sistemas de llaves públicas y protocolos de autenticación para mantener la seguridad de las comunicaciones que se realizan por Internet, de forma que verifique la procedencia de mensajes de correo o garantiza las fuentes de donde proceden las descargas. Por otra parte, incluye soporte para redes privadas virtuales (VPN), protocolos encapsulados que crean un "canal" de comunicación privado a través de redes públicas.

Windows 2000 Server es uno de los sistemas más revolucionados de la familia Microsoft ya que tiene menos interrupciones y hay un mayor tiempo de funcionamiento para el servidor, que es de gran importancia para las operaciones de aspecto crítico o que requieren de una respuesta de forma constante como son el comercio electrónico en donde el tiempo fuera de servicio es crucial ya que las ventas pueden perderse. Debido a esto Windows 2000 Server incluye avanzados servicios de clustering y balance de cargas para que pueda implementar soluciones que distribuyan la carga en múltiples servidores. Con el modo de arranque seguro para una rápida detección y solución de errores, los administradores pueden tener un servidor de nuevo en línea para sus clientes.

De acuerdo a lo anterior se puede decir que las principales características de Windows 2000 Server son las siguientes:

- Fiabilidad
- Manejabilidad
- Integración
- Estabilidad

#### **Requerimientos del sistema operativo Windows 2000 Server**

Para poder tener Windows 2000 Server como sistema operativo se requiere tomar en cuenta las características del equipo en el cual será instalado por lo que se requiere lo siguiente:

- Como mínimo un procesador Pentium a 133 Mhz
- 256 Mb de memoria en RAM
- Disco duro de 2Gb con un mínimo de 1Gb de espacio libre

### **Ventajas de Windows 2000 Server**

- Utiliza la Web para conectar a sus clientes, incluso aunque estén dispersos geográficamente o más allá de los límites de su red corporativa.
- Construcción de aplicaciones internas para el negocio.
- Compartir información seleccionada sin comprometer la confiabilidad de sus datos.
- Hacer crecer un sistema a medida que sus aplicaciones necesiten cambiar.

### **Desventajas de Windows 2000 Server**

- Es un producto muy costoso.
- Windows siempre ha sido uno de los sistemas operativos más atacados por virus.

### **Características del sistema operativo Windows XP**

Windows XP profesional constituye una de las alternativas más avanzadas en sistemas operativos. Es un sistema operativo multiproceso, multitarea, multiusuario y de tiempo real que proporciona herramientas de productividad avanzadas para poder administrar, implementar y soportar las aplicaciones de forma eficiente, esta diseñado para proporcionar una base confiable con lo más reciente en seguridad y confiabilidad, al tiempo que ofrece rendimiento, facilidad de uso y movilidad. Es un Sistema Operativo con la capacidad de migrar los datos de usuarios y la configuración de aplicaciones desde equipos que están diseñados bajo plataformas anteriores o más bajas.

Windows XP también tiene dentro de sus características la fiabilidad, que garantiza que verificará los drivers de cada dispositivo que utiliza, lo que lo hace un sistema más estable y que las estructuras críticas de datos Kernel sean de solo lectura, de tal manera que los drivers no se pueden ver afectados, adicionalmente todo el código de drivers de dispositivos es de sólo lectura y está protegido. De esta forma las aplicaciones defectuosas no pueden afectar todas las áreas del sistema operativo aumentando significativamente la confiabilidad. Para llevar a cabo esta protección se busco que los procesos fueran separados, para que las aplicaciones errantes no provocaran daños al equipo, logrando que cada aplicación este en cada espacio de memoria separado y protegido; además de que en esta nueva versión de Windows se busco eliminar la mayoría de los escenarios que obligaban a los usuarios a reiniciar su equipo, incluso cuando no es necesario y en caso de que por alguna razón tuviese que restaurar el sistema, Windows XP le permite regresar al estado anterior sin perder datos personales. Al restaurar el

sistema se supervisa de forma activa los cambios realizados en los archivos para registrar o guardar versiones anteriores. Posee un administrador de tareas que ofrece información útil del rendimiento del equipo y permite terminar programas inactivos.

Dentro de las muchas características que ofrece esta la de proteger los archivos del sistema para que no sean sobrescritos y en todo caso la protección de archivos de Windows lo reemplazará con la versión correcta reduciendo de esta manera fallas comunes del sistema. Pero esto también se puede evitar desde antes gracias a que el sistema cuenta con restricciones para el software que proporciona a los administradores mecanismos para identificar que software se esta corriendo o ejecutando.

Posee un sistema de encriptación y desencriptación de archivos que son transparentes al usuario, permitiendo la seguridad sobre todo contra el robo de información. También posee seguridad en cuanto a las IP's ayudando a proteger los datos que se transmiten a través de la red, proporcionando de esta manera seguridad tanto a redes privadas como virtuales y permitiendo transmitir datos de forma segura a través de Internet. En cuanto al tema del Internet y la seguridad se puede mencionar que ofrece un *firewall*<sup>8</sup> que puede ayudar a proteger a las empresas pequeñas y grandes de ataques comunes provenientes de Internet, además permite con el permiso de los usuarios hacer actualizaciones para la seguridad en Internet.

Otra de sus capacidades es la de poder monitorear el estado del procesador a fin de reducir sobrecargas de energía evitando que el sistema sufra algún desperfecto o que se apague prematuramente.

De acuerdo con lo anterior se puede decir que las principales características de Windows XP son las siguientes:

- Fiabilidad
- Escalabilidad
- Manejabilidad
- Robustez
- Seguridad

---

<sup>8</sup> Firewall (corta fuegos). Se denomina así al Sistema de seguridad que se coloca entre la red local e Internet, de esta manera la empresa o compañía regulará completamente toda la comunicación hacia Internet, sus políticas de seguridad. En ocasiones este sistema incorpora autenticación de usuarios entre otras cosas.

### Requerimientos del sistema operativo Windows XP

Para poder instalar Windows XP como sistema operativo se requiere tener un equipo con las siguientes características:

- Un procesador Pentium III a 350 Mhz o mínimo de 233 Mhz
- 128 Mb de RAM o superior
- 2 Gb de espacio disponible en disco duro
- Unidad de CD-ROM

### Ventajas de Windows XP

- Rapidez
- Facilidad de uso
- Estabilidad
- Recuperación de archivos
- Soporta nuevo hardware
- Incluye firewall

### Desventajas de Windows XP

- Precio excesivo
- No reconoce todo el hardware
- Consumo excesivo de recursos
- No corren programas que se ejecutan directamente en DOS
- Incompatibilidad con cierto hardware
- Actualmente es uno de los sistemas operativos más atacado por virus
- Continúa actualización de parches para quitar desperfectos

### 2.2.5. Análisis para elección del Data Base Manager System (DBMS)

En la actualidad las bases de datos han adquirido un papel sobresaliente en el campo de la información, debido al impresionante crecimiento de volumen e importancia de los archivos de datos utilizados en las computadoras.

Las causas de que hoy en día, se tenga una gran disponibilidad de información en el momento requerido se debe principalmente al desarrollo tecnológico, así como al desarrollo de nuevos métodos para el almacenamiento.

Una Base de Datos se define como una colección de datos interrelacionados, los cuales se encuentran almacenados en un conjunto sin redundancias innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible.

El sistema de Base de Datos es la integración de programas y archivos que en conjunto dan apoyo a las Bases de Datos (éste apoyo consiste en registrar y mantener información, garantizando su seguridad), esto es lo que forma un sistema de manejo de datos; el sistema debe proporcionar facilidades para disponer de la información cuando se le solicite.

**BDMD (Data Base Manager System):** Es un Sistema de Administración de Bases de Datos, que consiste en un conjunto de datos relacionados entre sí, también es un conjunto de programas que tienen acceso a los datos, algunos de sus principales objetivos son:

- Eliminar o disminuir la redundancia
- Mantener la consistencia de los datos
- Resolver los problemas de concurrencia
- Recuperar información de las Bases de Datos en una forma conveniente y eficiente
- Conservar la integridad de los datos

El funcionamiento del DBMS está muy interrelacionado con el sistema operativo, especialmente con el sistema de comunicaciones, ya que utiliza las facilidades del sistema de comunicaciones para recibir las peticiones del usuario (que bien puede estar utilizando una terminal físicamente o de manera remota).

El DBMS realiza la traducción entre los diferentes esquemas de la Base de Datos. Si un usuario quiere acceder a los datos, el manejador comprobará su esquema externo para averiguar a que datos

tiene acceso; posteriormente estudia el esquema conceptual completo, a continuación al esquema físico para saber como trabajar con ellos y finalmente los proporcionará al usuario.

### 2.2.6. Funciones del Data Base Manager System (DBMS)

Las funciones del DBMS se pueden clasificar de la siguiente manera:

**Funciones de definición:** Permiten describir los elementos de datos, sus estructuras, sus interrelaciones y sus validaciones a nivel externo, lógico e interno. Esta función es realizada por una parte del DBMS denominado lenguaje de definición de datos (DDL-Data definition Language).

**Funciones de manipulación:** Permite buscar, añadir, suprimir, y modificar los datos de la Base de Datos. Esta función es realizada por una parte del DBMS denominada de manipulación de datos (DML-Data Manipulation Language).

**Función de utilización:** Incluye otras funcionalidades tales como, modificar la capacidad de los registros, cargar archivos, realizar copias de seguridad, arranque, protección frente a accesos no autorizados, estáticas de utilización, etc.

### 2.2.7. Programas que conforman el DBMS

El DBMS se compone de una serie de módulos que son los encargados de realizar las funciones antes mencionadas:

**Compilador DDL (Data Definición Lenguaje):** Se utiliza para crear y mantener la base de datos y los elementos que contiene a nivel externo, lógico e interno. Es propio de cada DBMS y permite definir entidades, identificadores (claves), atributos, interrelaciones, autorizaciones de acceso, restricciones de integridad, etc. A nivel interno facilita la definición del espacio físico, longitud de los campos, representación de los datos (binarios, alfanuméricos...), caminos de acceso (punteros, índices...), etc.

**Procesador de consultas:** Permite al usuario consultar los datos sin necesidad de construir un programa de aplicación. Cuenta con un *Optimizador de DML* para optimizar consultas.

Una vez definidas las características generales de los administradores de Bases de Datos, daremos paso a la elección de aquel que se hará cargo de la administración de la Base de Datos que se desarrollará en este trabajo.

### 2.2.8. Descripción y características de PostgreSQL, MySQL y Oracle

#### PostgreSQL

PostgreSQL es un administrador de Bases de Datos Relacional, que como otros productos libres y comerciales salió de la investigación de las universidades. En el caso de PostgreSQL, este surge del proyecto de Ingres de la Universidad de Berkeley.

PostgreSQL soporta la mayor cantidad de las instrucciones de SQL que es el estándar, y por ello es uno de los manejadores de Bases de Datos más populares y de mejor funcionamiento dentro de lo que es el software libre, que para lograr una mayor eficiencia une todas las estructuras clásicas con los conceptos de la programación orientada a objetos, lo que la convierte en una Base de Datos Objeto-Relacional.

PostgreSQL es un administrador de Bases de Datos con todas las características y ventajas de un manejador comercial. Soporta SQL92, transacciones, Triggers, Constrain, foreigns Keys, replicación, etc. Además de la conectividad ODBC para ser utilizado con aplicaciones Windows permitiendo de esta manera generar nuevas aplicaciones, mantener las que ya existen y hacer migraciones de Access, Visual Basic, Foxpro, C/C++, Delphi, etc.

Un concepto que aparece en PostgreSQL y lo distingue de otros modelos es la herencia, este concepto surge de la programación orientada a objetos. Así pues, cuando se crea una nueva clase, heredada de otra, la clase creada adquiere todas las características de la clase de la que proviene, más las características que se definen en la nueva clase. De la misma forma cuando se hereda una clase, se está creando una tabla igual a la de origen de la herencia.

Para su buen funcionamiento utiliza el modelo cliente/servidor, que nos da como resultado un proceso por usuario. Este modelo consta de un programa supervisor, uno o varios programas de aplicación y un servidor de Base de Datos por cada programa de aplicación. El programa supervisor se ejecuta en el sistema y gestiona una colección de Base de Datos. Los programas de aplicación realizan las peticiones de acceso a una Base de Datos por medio de la biblioteca de funciones llamada libpq. La

conexión entre el programa de aplicación y el servidor de Base de Datos se realiza por medio de un socket TCP/IP (Transmisión Control Protocol/ Internet Protocol).

Posee las siguientes características que le dan más flexibilidad y potencia:

### Características Operacionales

- Transacciones (Transactions)
- Disparadores (Triggers)
- Restricciones (Constraints)
- Replicación (Replication)
- Backup y Recuperación (Backup & Recovery)
- Reglas (Rules)
- Stored
- Procedimientos Almacenados/Funciones (Procedures/Functions)
- Integridad Referencial
- Outer Joins
- Sintaxis ANSI SQL 89,92 y 98
- Logging
- Extensible
- Orientación de Objetos
- Integridad de datos
- Funciones definidas por el usuario
- Cliente/Servidor

### Requerimientos mínimos para su instalación

- Memoria principal de 8 Mb
- Espacio libre en disco duro 100 Mb
- Sistema Operativo 95/98/NT/ME/2000 ó Linux
- Protocolo TCP/IP

### Alcances de PostgreSQL

- El tamaño de Base de Datos esta sólo limitada por la capacidad de almacenamiento de hardware.
- El tamaño máximo de una tabla es de 64 terabytes
- Máximo tamaño de un campo 1 Gb
- Puede ser utilizado en varias plataformas

### Ventajas

- Por su arquitectura de diseño, escala perfectamente bien al aumentar el número de CPU's y la cantidad de RAM.
- Soporta transacciones y claves foráneas (con comprobaciones de integridad referencial)
- Tiene mejor soporte para triggers y procedimientos en el servidor
- Esta apegado a las especificaciones de SQL92
- Es de libre distribución

### Inconvenientes

- Consume más recursos y carga más el sistema
- Es un poco más lenta que MySQL

### Descripción y características de MySQL

MySQL es un administrador de Base de Datos de uso "libre" y de código abierto, que soporta a múltiples usuarios, usa el lenguaje SQL como estándar para el almacenamiento, actualización y acceso de la información. Por sus características es capaz de almacenar una gran cantidad de información haciendo que el acceso a esta sea muy seguro, utilizando para ello una protección mediante contraseñas.

MySQL es uno de los tantos productos de código abierto, y esta disponible para plataformas como Linux y Windows. Para las aplicaciones de Windows tiene un controlador ODBC, así que puede conectarse al servidor con cualquier aplicación que soporte ODBC, puede conectarse también utilizando la biblioteca libmysql.dll, que es más rápida pero menos transparente.

Debido a la característica antes mencionada soporta varios lenguajes de programación, dentro de los cuales podemos encontrar C, C++, Eiffel, Java, Perl, PHP, Pitón y TCL.

Posee una velocidad capaz de superar en mucho a los administradores de Bases de Datos comerciales, además es compacta, eficiente y amistosa con los recursos del sistema ya que tiene menos requerimientos. Otra de las características que posee es su portabilidad permitiendo la emigración de información a otras plataformas.

En este manejador de Base de Datos la información es tratada con gran cuidado por lo que las claves viajan de manera encriptada por la red, proporcionando de esta manera seguridad. Contiene también las características que le dan más flexibilidad y potencia.

### **Características operacionales**

- Multiproceso es decir que puede usar varias CPU's si están disponibles
- Cliente/Servidor
- Puede ser usada en varias plataformas entre ellas Linux y Windows
- Utiliza la función Isamchk para verificar, optimizar y reparar tablas
- Multi-usuario
- Multi-tarea
- Tiene un sistema de contraseñas y privilegios muy flexible y seguro
- Soporta mensajes de error en diferentes idiomas
- ODBC para Windows 95, se puede utilizar Access para conectarse con el servidor

### **Requerimientos mínimos para su instalación**

- Sistema operativo Windows 95/98/NT/ME/2000/XP ó Linux
- Memoria principal de 9 Mb
- Espacio en disco duro 400 Mb

### **Alcances de MySQL**

- Capacidad de hacer 101 conexiones al mismo tiempo
- Registros de longitud fija y variable

- La cantidad de índices por tabla es fijado en 16
- Tiene una capacidad de 16 índices por tabla, donde cada índice puede estar compuesto de 1 a 15 columnas o partes de ella con una longitud de 127 bytes.
- Todas las columnas pueden tener valores por defecto
- Puede utilizar variables de tipo entero, flotante, doble precisión, carácter, fecha, enumerado, etc.
- Confiable
- Robusta
- Rápida
- Flexible
- Libre distribución

### **Ventajas**

- Manejo de grandes cantidades de información
- No es necesario realizar un desglose de los campos enviados desde el formulario para obtener el valor de las variables.
- Posee una buena seguridad

### **Desventajas**

- Requiere estar corriendo como un proceso permanente dentro de su servidor para efectuar consultas a través de la Web.
- Se requiere de buenos conocimientos en SQL para explorar todo su potencial

### **Descripción y características de Oracle**

Oracle es uno de los manejadores de Bases de Datos más potente y estable de tipo objeto-relacional en el mercado y es considerado un gigante de la información; ya que puede albergar millones de datos y crecer lo suficiente para cubrir cualquier tipo de necesidad, además de combinar la eficiencia, puesto que posee herramientas de optimización para las consultas.

Para hacer de Oracle un Manejador de Bases de Datos más potente se utilizan restricciones que tiene como función la validación de contraseñas para poder acceder a la información almacenada.

En su versión llamada Oracle9i, éste ofrece herramientas de desarrollo para Internet y para transferir grandes cantidad de información a través de la red. Para llevar a cabo esta tarea utiliza una tecnología que consiste en tener un cache para la Base de Datos y un cache dinámico de páginas Web, dicha tecnología permite a las páginas Web alcanzar altos rendimientos y rapidez.

### **Características operacionales**

- Disparadores (Triggers)
- Restricciones (Constraints)
- Transacciones (Transactions)
- Orientada a objetos
- Procedimientos almacenados
- Utilización de password y contraseñas
- Descripción de los eventos fallidos
- Interrupción de procesos
- De tipo cliente/servidor
- Bloqueo de registros
- Multiproceso es decir que puede usar varias CPU's si están disponibles

### **Requerimientos mínimos para su instalación**

- Procesador Pentium 133 ó 166 Hz
- Disco duro de 2 Gb
- 128 Mb en RAM
- 950 Mb de espacio disponible en disco duro
- Sistema Operativo 95/98/Windows/NT/ME/XP ó Linux

### **Alcances de Oracle**

- Permite hacer subconsultas
- Puede aumentar sin límite
- Portable
- Robusta
- Escalable

- Estabilidad
- Rapidez

### **Ventajas**

- Tiene una selectividad muy alta; ya que puede respaldar desde un registro de la Base de Datos o toda la Información almacenada en ella.
- Posee una buena portabilidad, pues un archivo de "export" puede ser exportado de y desde cualquier sistema operativo que soporte Oracle7 o superior y ser importado en cualquier sistema operativo con la ayuda de SQL.
- Herramienta de reorganización que se utiliza una vez hecho un "export".
- Permite recuperar información perdida por errores de usuario o del servidor como son, drops, truncates, delates, corrupción de registros en tablas, perdida de tablas al perderse el tablespace o la Base de Datos, borrado de objetos.

### **Desventajas**

- Tamaño y tiempos impredecibles
- Puede que requiera pasar todo el archivo de export para importar solo una parte, debido al recorrido secuencial para realizar un import, si el objeto buscado esta al comienzo del archivo se detiene después de importarlo, pero si está al final tiene que recorrer todo el archivo para recuperar solo ese Objeto.
- Altos costos de la Licencia

### **2.2.9. Descripción de lenguajes utilizados para crear la interfaz (PHP y HTML)**

#### **PHP**

El lenguaje PHP (Acrónimo de PHP Hipertext Preprocessor) es un lenguaje para páginas Web que funcionan en un servidor. Es un lenguaje interpretado de alto nivel y con una programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones; etc. Toda la programación elaborada en PHP se coloca dentro de una página HTML.

Los navegadores no tienen ningún problema al trabajar con PHP, ya que nunca ven el código fuente, puesto que este se interpreta directamente en el servidor para generar la página HTML solicitada antes de ser transmitida al navegador.

PHP ha sido desarrollado principalmente para usarse conjuntamente con un servidor Web libre de Apache en cualquiera de las plataformas que este soporta. De hecho, está implementado como un módulo de forma que aprovecha la máxima integración con el servidor y la velocidad disponible.

Uno de los puntos principales de PHP es su API<sup>9</sup> para construir aplicaciones y conectar Bases de Datos. PHP no sólo puede usar muchos tipos de Bases de Datos de forma nativa sino que permite Interfaz ODBC<sup>10</sup> y JDBC<sup>11</sup>. PHP es muy utilizado sobre todo con PostgreSQL y MySQL, Base de Datos libres, haciendo entre Apache, PHP y MySQL una de ambas soluciones basadas en Software "libre" para hacer sitios Web dinámicos de alta calidad, con datos seguros, y sobre todo sin dependencia tecnológica de ninguna empresa en particular.

Todas las aplicaciones de PHP se ejecutan en el servidor, por eso nos permiten acceder a los recursos que este tiene, como por ejemplo, siendo este último una página HTML.

## HTML

Las siglas HTML representan los términos HyperText Markup Language, es decir un lenguaje utilizado para representar información hipertextual. En realidad, HTML es una serie de normas que permiten especificar características y funcionalidades de un documento manejado por un sistema hipertexto como los que se utilizan en WWW (World Wide Web).

HTML es un lenguaje derivado de uno de más altura, y más antiguo, denominado SGL (Standard Generalized Markup Language), que permite gestionar y formatear documentos mucho más complejos que una página Web.

---

<sup>9</sup> Application Program Interface. Interface de Aplicación del Programa, que es el conjunto de rutinas del sistema que puede usarse en un programa para la gestión de entrada/salida, gestión de ficheros, etc.

<sup>10</sup> ODBC es un Intermediario entre Bases de Datos y aplicaciones, cuya tarea es sostener una conversación de preguntas y respuesta entre dos "sujetos" que no hablan el mismo idioma y que gestionan sus recursos de forma diferente.

<sup>11</sup> JDBC (Java Database Connectivity). La conectividad de la Base de Datos de Java es un marco de programación para los desarrolladores de Java que escriben los programas que tienen acceso a la Información guardada en la Base de Datos, hojas de cálculo y archivos "planos".

Un documento HTML es un conjunto de párrafos caracterizados por una serie de rasgos que determinan su estilo. De este modo, un encabezado tiene características de presentación diferentes a un párrafo de texto convencional. El lenguaje HTML es utilizado para la creación de páginas Web. Con este lenguaje se consigue dar forma al texto que queremos publicar en la red, pudiéndole añadir fotografías, fondos, viñetas, etc.

HTML no se compila, ni se ejecuta, sino que se interpreta. Esto quiere decir que el servidor, cuando recibe una petición de página Web, envía el texto tal como está escrito. Es el browser (explorador) del cliente el que a partir del texto recibido y de las marcas que lo acompañan, presentan la página al usuario de una forma u otra. El explorador interpreta las marcas y aplica su contenido al texto, de modo que el usuario vea el resultado final en modo texto.

#### 2.2.10. Servidor Web Apache

Apache es un servidor Web, que permite el alojamiento de páginas Web en una máquina específica. Esta herramienta tiene varias funciones tales como, permitir a los usuarios tener sus propias página Web, restricción a determinados sitios Web, conexiones seguras a través de SSL<sup>12</sup>, configuración de módulos de programación. El objetivo del proyecto Apache es crear un servidor Web completo, robusto y capaz de competir con cualquier producto comercial por lo que ya desde hace varios años mucha gente contribuye con código, ideas y dinero al proyecto.

Tan sólo hace unos años, el servidor Web más popular era httpd de NCSA<sup>13</sup>. A mediados de 1994, el principal desarrollador del httpd de NCSA abandonó el proyecto, dejando que muchos Webmasters de todo el mundo tuvieran que hacer frente y lidiar con el código.

Finalmente se implementaron una serie de mejoras y se corrigieron errores, estos cambios se distribuyeron en forma de parches del código fuente. No pasó mucho tiempo hasta que el número de parches se volvió inmanejable, y la necesidad de una distribución estándar se convirtió en algo aún más importante.

El nombre Apache surge del hecho de que inicialmente no era más que una versión compuesta de un gran número de parches, disponibles gratuitamente para el código fuente httpd de NCSA.

---

<sup>12</sup> Secure Socket Layer, Protocolo desarrollado por Netscape para dar seguridad a las transacciones en Internet.

<sup>13</sup> Servidor basado en Unix, el cual proporciona archivos de dominio público.

El lanzamiento oficial del primer servidor Web Apache se realizó de 1995 con la versión número 0.6.2, la base de esta fue la versión 1.3 del httpd de NCSA. Durante 1995 continuó de forma rápida el desarrollo de Apache a partir de una versión original, dando lugar a la aparición de muchas versiones a lo largo de ese mismo año.

Finalmente el 10. de diciembre de 1995 se liberó la versión 1.0 de Apache. A mediados de 1997 este servidor Web se convirtió en el más popular, para 1999 The Apache Software Foundation, se incorporo como una organización sin fines de lucro.

Apache es le servidor Web más popular de Internet, ya que funciona en muchas plataformas, entre ellas, Linux, Solaris, HP-UX, Sun 4.x, la familia BSD e Incluso Windows 2000 y XP. Sin embargo Linux es definitivamente una de las plataformas más usadas para ejecutar apache debido a su estabilidad y bajo costo.

Aunque hay servidores Web que ofrecen ligeras mejoras de funcionamiento con respecto a Apache, esté último es el más rápido y funcional de todos los servidores Web gratuitos, debido a su bajo o nulo costo y a su rendimiento superior. Apache puede gestionar una gran cantidad de peticiones sin presentar ningún problema además de ser muy estable.

#### **2.2.11. Elección de software**

Tomando en cuenta todo lo expuesto en páginas anteriores llegamos a la conclusión de que Linux es la opción como sistema operativo para nuestros fines, por estable, rápido, potente, con un buen aprovechamiento de los recursos de la máquina y ser software libre.

Como manejador de Base de Datos optamos por utilizar MySQL puesto que nos ofrece muchas ventajas en cuanto a la capacidad de información que puede almacenar, además de ser gratuita. Otro punto importante para escoger MySQL es que es el manejador de Base Datos instalado en los servidores de la DIMEI.

En cuanto al desarrollo de la interfaz, uno de los mejores sin duda es PHP, ya que nos permite crear ambientes amigables a los usuarios, pero lo más importante es que permite tener una buena conexión con las Bases de Datos, además de que por sus propias características ofrece opciones para tener mayor seguridad en el sistema.

### 2.3. Análisis de Requerimientos de Hardware

Durante el tiempo que se elaboro el presente trabajo, nos apoyamos en dos distintos equipos de cómputo para realizar el sistema. Trabajar con estos equipos nos dio la posibilidad de verificar que el sistema a exportar tiene las mismas ventajas de correr tanto en una computadora personal como en un equipo servidor (Net Server).

Es necesario aclarar que aunque ambos equipos son óptimos para conectar el sistema, la ventaja del equipo Net Server radica en que el equipo en si esta hecho para mantenerse en servicio por largos periodos de tiempo sin necesidad de darlos de baja momentáneamente, es decir, no es necesario apagarlos, salvo para darles mantenimiento. Además de que cuenta con una unidad de cinta, útil para poder hacer respaldos de toda aquella información que fuera de gran importancia.

A continuación mencionamos las características técnicas de ambos equipos:

#### **Hewlett Packard Vectra VL400**

Hewlett Packard Vectra VL400 cuenta con arquitectura flexible misma que ofrece opciones en desempeño y velocidad, integra el nuevo chipset Intel 815 con gráficos de la siguiente generación (de manera integrada o en una tarjeta AGP); ofrece procesadores, desde Celeron 566 MHz hasta Pentium III a 1000 MHz, y posee una velocidad de Bus de 133 MHz en todos los canales.

Hewlett Packard Vectra VL400 viene con tres factores de forma para intercambiar, escritorio, minitorre y factor de forma pequeño (Small Form Factor). Como función opcional es posible adquirir un convertidor de PCI a ISA, de forma VL400 que protege las inversiones hechas por el cliente en periféricos ISA anteriores.

Se tuvo la oportunidad de evaluar nuestro Sistema en un equipo HP Vectra VL400 con procesador Intel Pentium III a 733 MHz, 128 Mb en RAM, 10 Gb en disco duro, unidad de CD-ROM a48x, tarjeta de red 3Com integrada, puertos PS/2 para ratón y teclado, un puerto paralelo, un puerto serial y dos puertos de USB. Véase la Fig. 2.1.

Para medir el rendimiento de la computadora Vectra VL400 se realizaron diversas pruebas que involucraron trabajo con programas de oficina, de diseño gráfico y de trabajo en Internet.

Para trabajar en Internet se utilizó Netscape Navigator e Internet Explorer. En ambos casos el desempeño fue bueno. En resumen, se puede decir que Hewlett Packard Vectra VL400 es una opción de considerarse debido a su desempeño y opciones de mantenimiento.



HP VL400  
 Procesador: Intel Celeron 733 MHz.  
 RAM: 32MB, 64MB o 128MB. Caché: 512 KB.  
 Disco Duro: 10 GB.

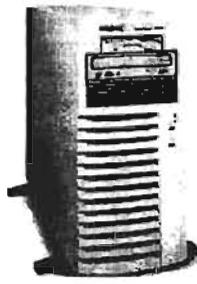
Figura 2.1. Equipo de cómputo HP VL400

### NetServer serie E800

Un Servidor como el NetServer E800 nos da una mayor seguridad en las comunicaciones remotas, en la administración de la red o en el almacenamiento de información. Esto se debe a que los proveedores han diseñado Software especializado para controlar sus diferentes dispositivos. Estos Servidores cuentan con controladores y dos discos duros que permiten tener una mayor seguridad y velocidad de intercambio de información. Cabe señalar que el precio de estos Servidores se encuentra alrededor de los \$2,000 USD y los dispositivos que utilizan (tarjetas, discos), son más caros que los dispositivos para una PC. Véase Figura 2.2.

Esté servidor posee las siguientes características:

- Dos procesadores Intel Pentium III, desde 733 MHz hasta 866 MHz.
- Bus frontal del sistema (FSB) 133 MHz.
- Memoria ECC SDRAM, aplicables a 2Gb (4 Slots), 256 Kb de caché de 22 niveles integrada en el procesador.
- Controladora de Videos Super-VGA integrada con 4 Mb de memoria SDRAM, capaz de conseguir una resolución de 1024X768, a 256 colores.
- Ultra SCSI de doble canal integrada en placa base dos puertos serie de 9-pin RS-232, un puerto paralelo de 25-pin, puerto SCSI, un mini-DIN para el ratón, un mini-DIN para el teclado, un puerto SVGA, dos puertos USB y un conector SCSI externo.
- CD-ROM 40X EIDE incluido y una unidad de Cinta para respaldos de Información.
- Siete slots de expansión PCI (cinco slots para 32-bit y dos slots para 64-bit).
- Sistemas de seguridad, entre los que se encuentran habilitador y deshabilitador del puerto.



HP NetServer E800  
Doble procesador Pentium III 800  
Dos DD de 9.1 GB cada uno  
384 MB SDRAM, expandible hasta 2GB  
CD-ROM  
Unidad de cinta

Figura 2.2. Equipos de cómputo NetServer serie E800

#### 2.4. Identificación de Usuarios del sistema

Se presenta la información dirigida a dos tipos de usuarios:

- Usuario común
- Administrador del sistema

**Usuario Común:** Es la persona o personas que realmente usarán el sistema; son quienes se sientan frente a la computadora, ingresan los datos o leen las salidas. También se puede decir que es aquel que interactúa con el sistema a través de las interfaces que proveen las aplicaciones estructuradas. Para el sistema que se desarrollo se identificaron los siguientes usuarios comunes:

- Alumnos
- Profesores
- Jefes de Departamento

**Administrador del Sistema:** Es el responsable de los datos almacenados y de su contenido. En este caso serán las personas que laboran en la Coordinación de Titulación de la DIMEI quienes manipularan constantemente el sistema.

# *CAPITULO III.*

## Diseño del sistema

El diseño es el primer paso de la fase de desarrollo de cualquier sistema y su objetivo es producir un modelo o representación de una entidad que será construida. El diseño es el proceso creativo de transformación del problema en una solución; la descripción de una solución también se denomina diseño.

El diseño es un proceso mediante el que se traducen los requisitos en una representación del software. El diseño puede realizarse en dos pasos. El diseño preliminar que se centra en la transformación de los requisitos en datos y arquitectura del software. El diseño detallado que se ocupa del refinamiento de la representación arquitectónica que lleva a una estructura detallada y a las representaciones algorítmicas del software.

El diseño de un sistema es de gran importancia, pues de este depende que el sistema no sea inestable o que falle al realizarse pequeños cambios. Por lo tanto es en el diseño donde se toman las decisiones que afectan finalmente el desarrollo, implementación y facilidad de mantenimiento que tendrá el sistema.

El diseño del sistema permite llevar a cabo el diseño de los datos, es decir transformar el modelo de información creado en el análisis, en las estructuras de datos que se van a requerir para implementar el sistema, así como las entidades, relaciones definidas en el modelo entidad relación y el contenido detallado del diccionario de datos.

Se definen además las relaciones entre los principales elementos estructurales del programa. Se parte del modelo de análisis y de la interacción de subsistemas definido en ese modelo.

### 3.1. Diseño del sistema

Existen diversos métodos para desarrollar un sistema, que conlleve a una aplicación exitosa.

Nuestro sistema utilizará la metodología de programación estructurada, esto significa escribir un programa de acuerdo a las siguientes reglas:

- El programa tiene un diseño modular
- Los módulos son diseñados de modo descendente
- Cada módulo se codifica utilizando las tres estructuras de control básicas: secuencia, selección y repetición.

El diseño modular o programación modular es uno de los métodos de diseño más flexible y potente para mejorar la productividad de un programa. En programación modular, la programación se divide en módulos (partes independientes). Cada uno de los cuales ejecuta una única actividad o tarea y se codifica independientemente de otros módulos. Cada uno de estos módulos se analiza y codifica por separado.

Los módulos son independientes en el sentido de que ningún módulo puede tener acceso directo a cualquier otro módulo excepto el módulo al que llama y sus propios submódulos. Sin embargo, los resultados producidos por un módulo pueden ser utilizados por cualquier otro módulo cuando se transfiere a ellos el control.

Dado que los módulos son independientes, diferentes programadores pueden trabajar simultáneamente en diferentes partes del mismo programa. Esto reducirá el tiempo de diseño del algoritmo y posterior codificación del programa. Además, un módulo se puede modificar radicalmente sin afectar a otros módulos, incluso sin alterar su función principal.

La descomposición de un programa en módulos independientes más simples se conoce también como el método de "divide y vencerás". Se diseña cada módulo con independencia de los demás, y siguiendo un método ascendente o descendente se llegará hasta la descomposición final del problema en módulos en forma jerárquica.

El **Diseño descendente (top-down)** es el proceso mediante el cual un problema se descompone en una serie de niveles o pasos sucesivos de refinamiento. La metodología descendente consiste en efectuar una relación entre las sucesivas etapas de estructuración de modo que se relaciona una con otras mediante entradas y salidas de información. Es decir, se descompone el problema en etapas o

estructuras jerárquicas, de forma que se puede considerar cada estructura desde dos puntos de vista: ¿Qué hace? y ¿Cómo lo hace?

En el diseño del sistema se lleva a cabo la estructura modular para definir las interfaces entre los diferentes módulos y programas, así como su estructura y flujo de datos.

Las fases que contiene el diseño del sistema son: La elaboración del diagrama de flujo, el modelo de datos, el diseño de la Base de Datos y la documentación de todo lo anterior en el diccionario de datos.

### 3.2. Diagrama de Flujo de Datos

El diagrama de flujo de datos (DFD) es una representación gráfica que describe los flujos de datos, los procesos que cambian o transforman los datos en un sistema, las entidades externas que son fuente o destino de los datos, y los almacenamientos o depósitos de datos a los cuales tiene acceso el sistema, permitiendo así describir el movimiento de los datos a través del sistema.

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

En síntesis, el Diagrama de Flujo de Datos describe:

- Los lugares de origen y destino de los datos (los límites del sistema)
- Las transformaciones a las que son sometidos los datos (los procesos internos)
- Los lugares en los que se almacenan los datos dentro del sistema
- Los canales por donde circulan los datos

### 3.2.1. Simbología

#### Entidad Externa:

Una **entidad externa** consiste en un sistema o subsistema que provee información de entrada pero que no está comprendido en el sistema a estudiar.

Son generalmente clases lógicas de cosas o de personas, las cuales representan una fuente o destino de transacciones, como por ejemplo alumnos, profesores, etc., con las que el sistema se comunica.

Al designar alguna cosa o algún sistema como entidad externa estamos estableciendo implícitamente que se encuentra fuera de los límites del sistema que estamos considerando, por lo cual no nos interesa la transformación o proceso que se realiza dentro de ellos, es decir que están fuera del control del sistema que se está modelando. Estos sólo proveen o solicitan datos del sistema bajo consideración.

Para su representación se utiliza el rectángulo:



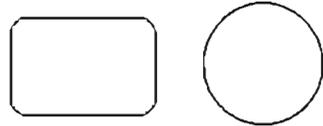
#### Proceso:

Un **proceso** es la acción o conjunto de acciones que permiten convertir o transformar los datos de entrada en los datos de salida del sistema. Gráficamente indican aquellos lugares dentro del sistema en donde la información (flujos de datos) que ingresa se procesa o transforma. Es decir, son los procesos que transforman entradas de datos en salidas de información.

El nombre del proceso generalmente se escribe dentro del componente, y se coloca mediante una frase imperativa, que consistirá idealmente de un verbo activo seguido por una cláusula objeto, cuanto más simple mejor. Estas frases imperativas no tienen sujeto; tan pronto como se introduce un sujeto se habrá indicado como deberá realizarse físicamente la función ("El operador ingresará los datos del alumno").

En otras palabras un proceso es una acción que se ejecuta sobre un dato, como por ejemplo, calcular, imprimir, almacenar, validar información, etc.

Es representado por un círculo o un rectángulo con puntas redondeadas:



### Flujo de datos:

Representa una transferencia de información desde su origen hasta su destino, es decir que representa una estructura de datos en movimiento de una parte del sistema a otro.

Un flujo muestra las interfaces entre los elementos del DFD. Puede imaginarse como una tubería por donde se envían paquetes de datos, pero deberá tener una descripción de su contenido la cual deberá elegirse de forma que sea lo más útil posible a los usuarios que revisen el DFD.

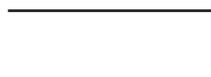
El flujo de datos es representado por una flecha, en donde la punta señale el destino u origen de los datos. Los datos que se mueven a lo largo del flujo, pueden viajar de un lado a otro (como entrada), o a un almacén. La dirección de la flecha indica si el flujo se está moviendo de adentro hacia fuera o hacia adentro del proceso.

La flecha indica la dirección del flujo: 

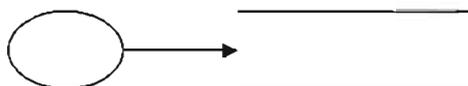
### Almacén:

Es el depósito de los datos, que se utiliza para modelar una colección de datos en reposo. Un flujo hacia un almacén se puede describir como una escritura, una actualización o una eliminación.

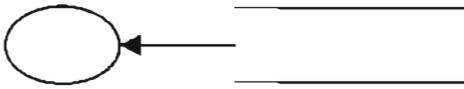
Gráficamente se utiliza un rectángulo sin los bordes laterales:



Deberá elegirse el nombre que sea más descriptivo para el usuario, que identifique los paquetes de datos que contiene.



Implica escritura, actualización o borrado de datos.



Implica lectura o recuperación de información almacenada.

### 3.2.2. Diagrama de Flujo de Datos Físico

El Diagrama de Datos Físico nos muestra como es el flujo de la Información a través del sistema. El ciclo de la información se completa de manera general cuando el usuario accede al sistema y a la base de datos para presentarle la información contenida en ella, el usuario puede manipular esta información por medio de los diferentes módulos con que se cuenta, para realizar una actividad específica, entonces el sistema accederá a la base de datos para extraer o actualizar la información que solicita el usuario. Véase la Figura 3.1.

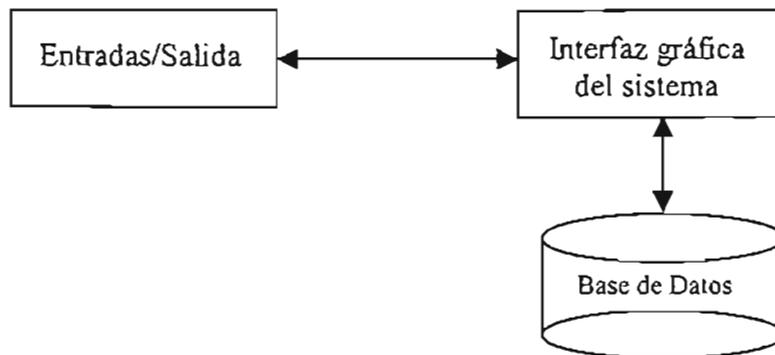


Figura 3.1. Representación del flujo de datos físico en el sistema

### 3.2.3. Diagrama de Flujo de Datos Lógico

El Diagrama de Flujo de Datos Lógico nos muestra como es el flujo de la información a través de los procesos que el sistema realizará. El ciclo de la información se completa cuando el sistema ha llevado a cabo los diferentes procesos que completan una tarea específica.

**Altas:** El sistema permitirá dar de alta, los proyectos de tesis, alumnos, directores de tesis, áreas, departamentos; etc. Véase Figura 3.2.

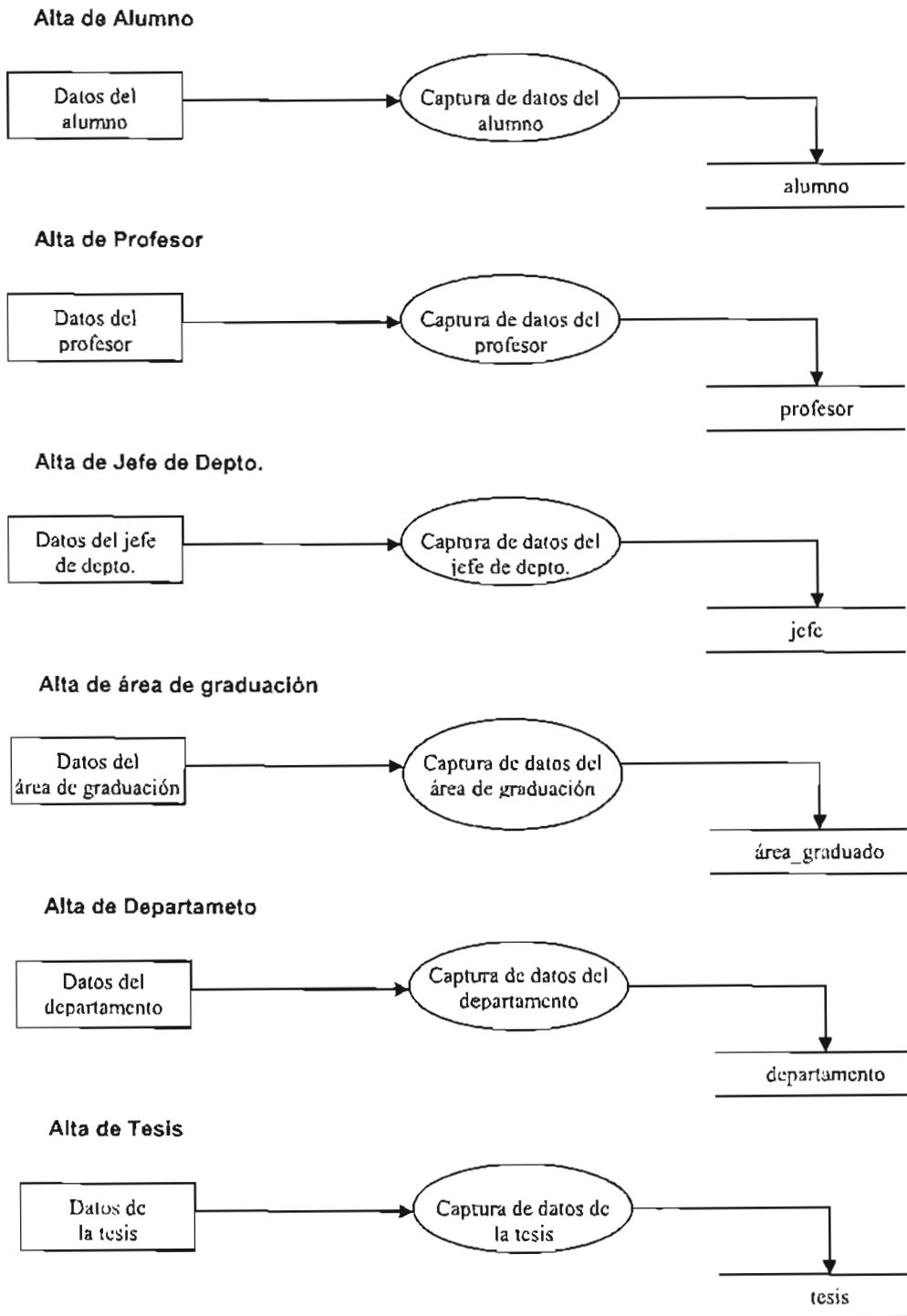
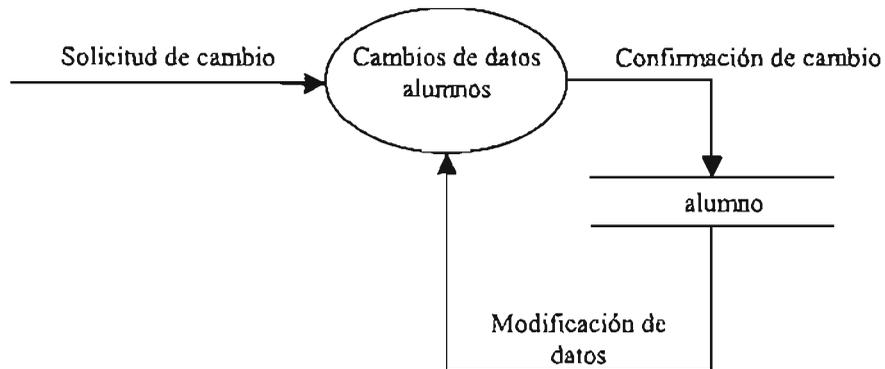


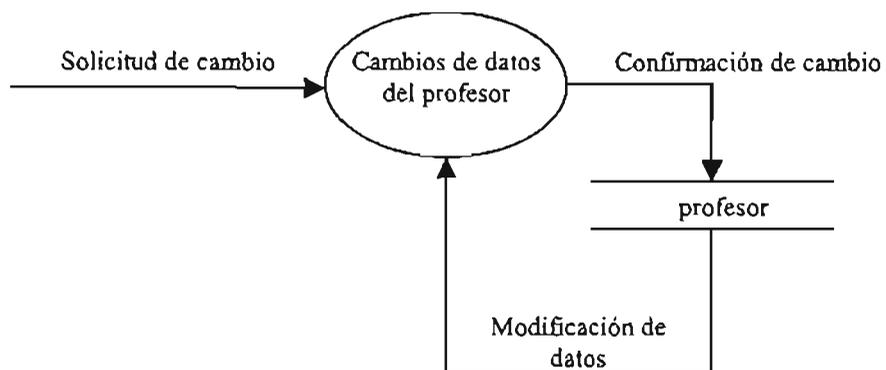
Figura 3.2. Representación gráfica de los procesos de alta

**Cambios:** El sistema permitirá la modificación de la información almacenada. Véase Figura 3.3.

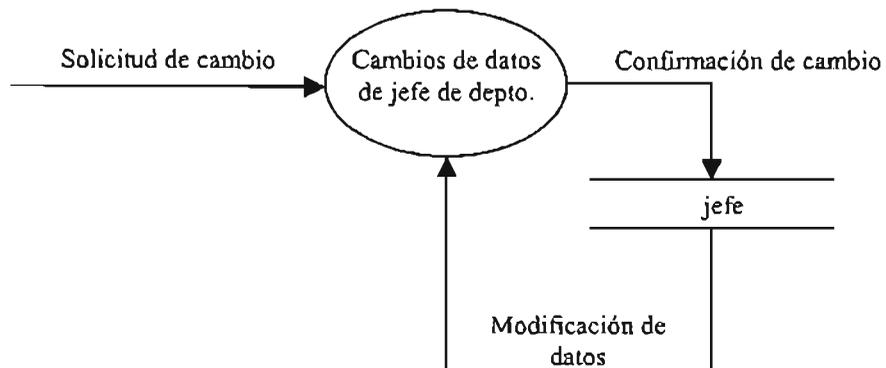
### Cambio de datos del Alumno

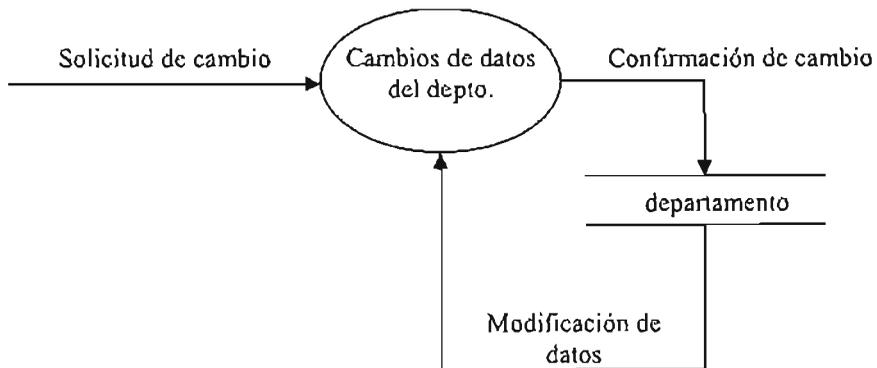
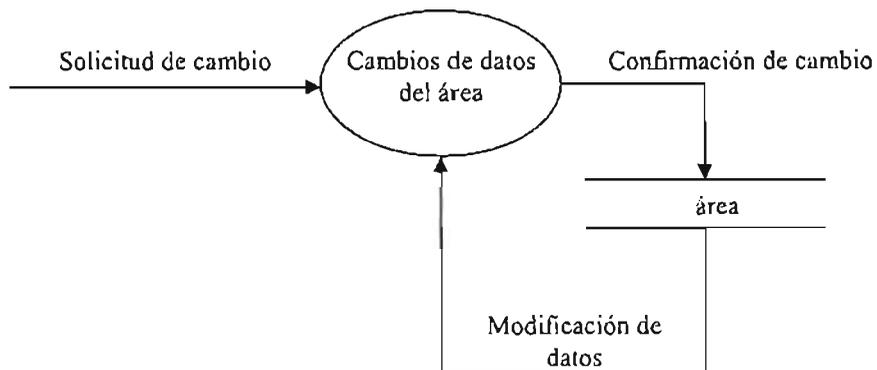
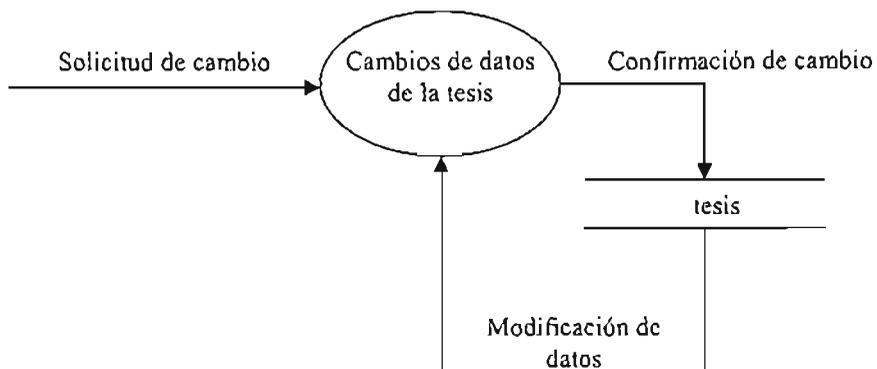


### Cambio de datos del Profesor



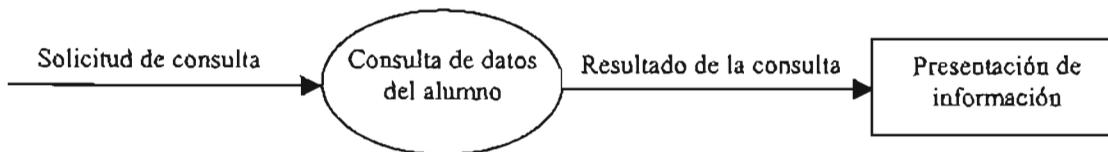
### Cambio de datos del Jefe de Depto.



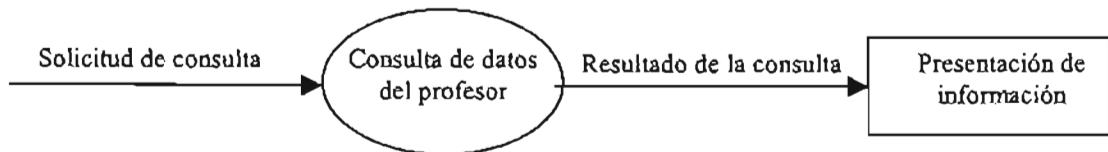
**Cambio de datos del Depto.****Cambio de datos del Área****Cambio de datos de Tesis****Figura 3.3. Representación gráfica de los procesos de cambio**

**Consultas:** los usuarios pueden realizar consultas de la información alojada en la Base de Datos. La información que cada uno de ellos pueda ver depende de los permisos que se le hayan otorgado. Véase la Figura 3.4.

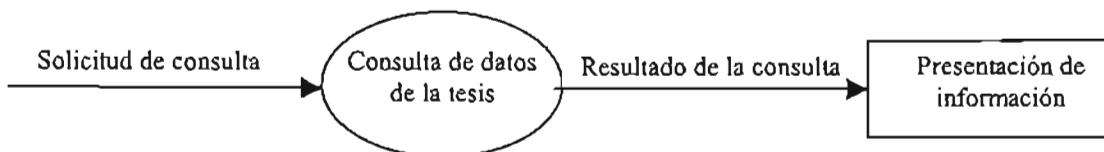
#### Consulta de información de Alumnos



#### Consulta de Información de Profesores



#### Consulta de información de Tesis



#### Consulta de Información de Tesis propuestas por la división

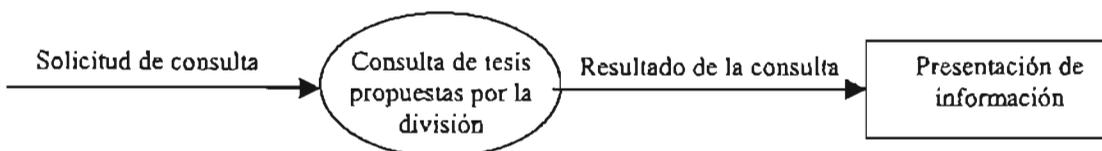


Figura 3.4. Representación gráfica de los procesos de consulta

**Reportes:** La información de los proyectos de tesis podrá ser enviada a un reporte que se puede imprimir. Ver Figura 3.5.

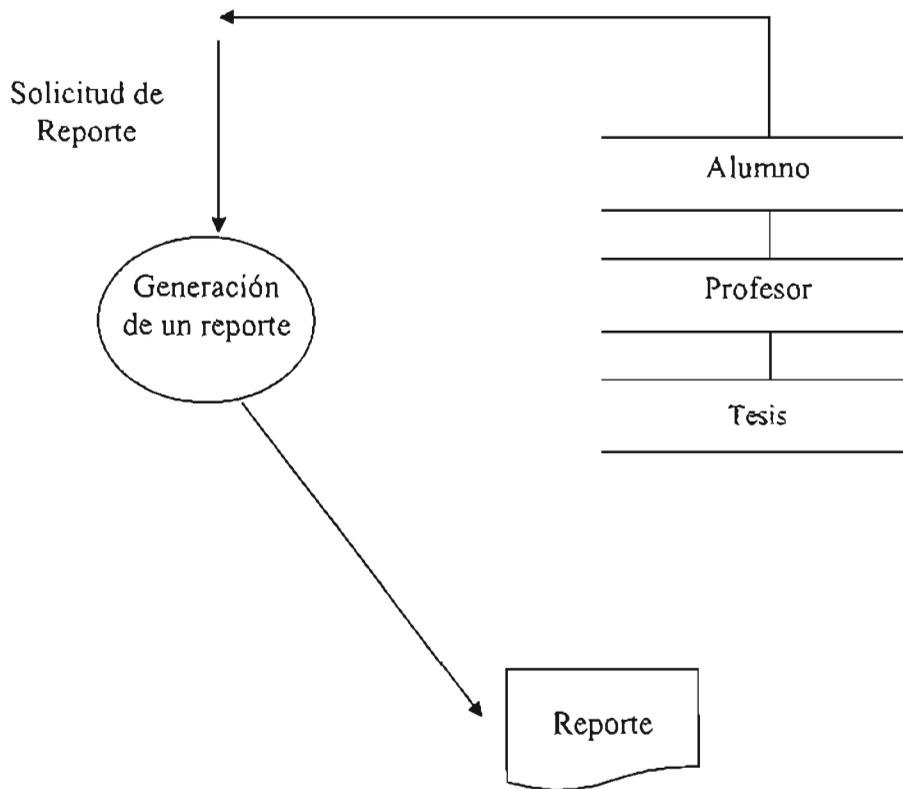


Figura 3.5. Representación gráfica del proceso de Impresión

### 3.3. Modelo Entidad – Relación

#### 3.3.1. Modelo de la Base de Datos

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

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

Según van evolucionando los tiempos, las necesidades de almacenamiento de datos van creciendo y con ellas las necesidades de transformar los mismos datos en información de muy diversa naturaleza.

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se dispone el problema en subproblemas y se resuelve cada uno de estos independientemente, utilizando técnicas específicas.

Es necesario definir el concepto de modelo de datos para poder describir la estructura de una base de datos. Este modelo de datos es un grupo de herramientas conceptuales que nos ayudan a describir los datos, sus relaciones, su semántica y sus limitantes.

La metodología de diseño de base de datos se divide en tres modelos:

**Modelo Conceptual:** El cual parte de las especificaciones de requisitos de usuario. El objetivo del modelo conceptual es describir el contenido de la información que almacenará la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. Es decir que el Modelo Conceptual es el conjunto de todos aquellos requerimientos que el cliente pone de manifiesto al solicitar la elaboración del sistema y del cual se parte para la elaboración del modelo de datos entidad-relación. Véase Figura 3.6.

**Modelo de Datos Entidad-Relación:** Se basa en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y las relaciones entre estos. Se desarrollo para facilitar el diseño de las bases de datos permitiendo las especificaciones de un esquema empresarial, que representa la estructura lógica global de la base de datos. Su objetivo es representar la estructura de los datos y las dependencias de los mismos, garantizando la consistencia y evitando la duplicidad.

**Modelo Físico:** Se trata del almacén de los datos, es la base de datos en si misma, el soporte donde se almacenan los datos y de donde se extraen para convertir los datos en información.

Hasta este momento contamos con el modelo conceptual, que se refiere a los requerimientos del sistema los cuales ya fueron descritos.

La siguiente tarea será la de elaborar el Modelo **Entidad-Relación**



Figura 3.6.

#### Definición del Modelo Entidad-Relación:

El Modelo Entidad-Relación es un modelo que describe la distribución de los datos de los datos almacenados en el sistema. También se le conoce con DER. El principal propósito del DER es representar los objetos de datos y sus relaciones.

Este modelo de datos se basa en una apreciación del mundo real y está diseñado de forma abstracta para poder ser aplicado. Dicho modelo consiste en un conjunto de objetos llamados *entidades*, y las *relaciones* que entre estos existen. Véase Figura 3.11.

#### Tipos de Objetos

##### Entidad

Una entidad puede ser cualquier objeto o evento que a parte de distinguirse de otros por ciertas particularidades (atributos) que lo describen, es también una fuente de datos, véase Figura 3.7. No necesita ser algo "tangible" como lo es el concepto de un libro o una computadora, puede ser también una abstracción como una fiesta o un concepto. Se le llama *conjunto de entidades* al conjunto de entidades del mismo tipo. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior.

### Atributos

Los atributos son características propias de cada una de las entidades, y para cada atributo existe un rango de valores permitidos al cuál se le conoce como *dominio* del atributo. Por ejemplo, el dominio para cierto atributo podría ser el conjunto de todas las cadenas de texto de cierta longitud.

De este modo cada entidad posee un conjunto de atributos los cuales se representan por las parejas: nombre del atributo y valor del dato.

También se puede definir como una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades.

### Relaciones

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior.

Pero este modelo no solo se basa en entidades y sus relaciones; se cuenta además con ciertas condiciones que se deben cumplir para poder llevar a cabo la relación entre las entidades, y una de estas es la cardinalidad de mapeo, la cual expresa el número de entidades con las que puede asociarse otra entidad por medio de un conjunto de relaciones. La cardinalidad de mapeo adecuada estará dada por la manera en la que se haya modelado el sistema.

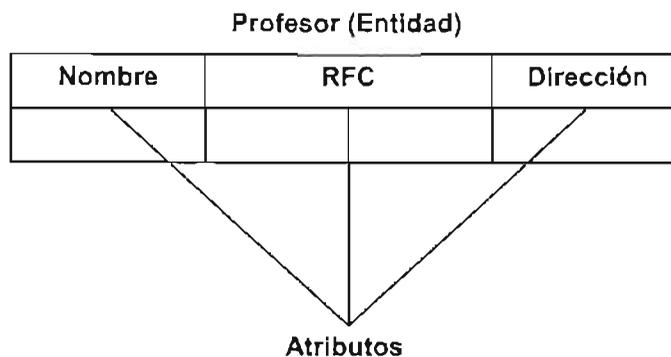
### Uniones

Son las que permiten enlazar las distintas entidades ya sea con sus atributos o con las relaciones que las afectan, se representan por medio de líneas.

Hagamos ahora una analogía con nuestro lenguaje a fin de entender como pasar del modelo Conceptual al modelo Entidad-Relación.

En el lenguaje humano se tiene:

**Sujeto:** Aquel que realiza una acción → **Entidades**  
**Verbo:** La acción que realiza el sujeto → **Relación**  
**Adjetivo:** Lo que se dice del sujeto → **Atributo**



**Figura 3.7. Representación de una entidad y sus atributos**

En resumen el Modelo Entidad-Relación es una conexión finita de tablas que representan nuestra situación. Para elaborar el modelo Entidad-Relación es necesario entonces elaborar las tablas o entidades que representen la situación. El modelo Entidad-Relación para nuestro sistema esta representado en la figura 3.13.

### 3.3.2. Entidades

Las tablas o entidades deberán contar con ciertas características para que las podamos incluir en nuestro modelo de datos Entidad-Relación y que son las siguientes:

#### Llave primaria

Las tablas deberán incluir una columna o grupo de columnas que identifiquen de manera única a cada renglón de la tabla. Esta llave, debe ser no nula (carente de valor), no duplicable y no cambiante.

### Llave foránea

Es una columna o grupo de columnas que es llave primaria en una entidad y atributo en otra. Para que exista este tipo de llave debe existir una relación entre ambas entidades.

- El nombre de la tabla es único dentro del modelo de datos Entidad-Relación
- El nombre de cada columna es único dentro de la tabla
- Cada renglón es único, lo aseguramos con la llave primaria
- No es posible descomponer columnas
- Cada atributo tiene uno y solo un valor

### 3.3.3. Mapeo del modelo Entidad- Relación

**Relación Uno a uno (1:1):** Una entidad esta relacionada con otra sola entidad y esta última entidad esta relacionada únicamente con la entidad anterior, (Fig. 3.8.)



Figura 3.8. Relación uno a uno

**Relación Uno a Muchos (1:M):** Una entidad esta asociada a varias entidades de otro conjunto, pero cada una de estas entidades esta asociada a una sola entidad del primer conjunto, (Fig. 3.9.)

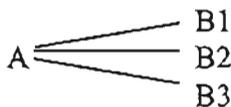


Figura 3.9. Relación uno a muchos

**Relación Muchos a muchos (M: M):** Una entidad esta relacionada con varias entidad de otro conjunto, y una entidad de ese conjunto se puede encontrar asociada a cualquier entidad del primer conjunto, (Fig. 3.10.)

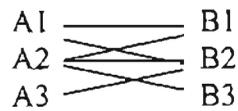


Figura 3.10. Relación muchos a muchos

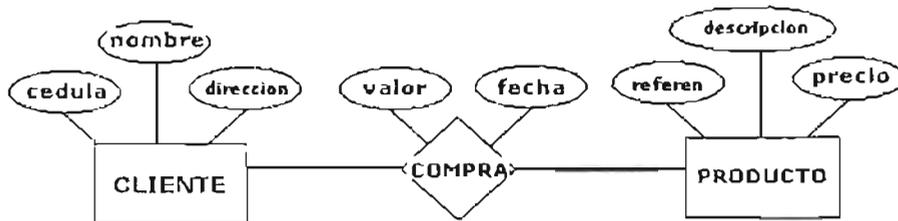


Figura 3.11. Ejemplo simple de un diagrama entidad-relación

Otra forma de representar los conjuntos de entidades y sus atributos consiste en mostrar los atributos en forma de lista dentro de un rectángulo, (Figura 3.12.)

<b>Alumno</b>
Núm. Cuenta
Nombre
RFC

Figura 3.12. Representación de una tabla

### 3.3.4. Normalización

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades.

### Evitando

- **La redundancia de los datos:** Repetición de datos en un sistema.
- **Anomalías de actualización:** Inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- **Anomalías de Borrado:** Pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- **Anomalías de Inserción:** Imposibilidad de adicionar datos en la base debido a la ausencia de otros datos.

### Asegurando

- Integridad entre los datos
- Consistencia de la información

El proceso de normalización nos conduce hasta el modelo Físico de Datos que consta de varias fases denominadas formas normales, estas formas se detallan a continuación.

### **Definición de la clave**

Antes de proceder a la normalización de la tabla lo primero que debemos definir es una clave, esta clave deberá contener un valor único para cada registro (no podrán existir dos valores iguales en toda la tabla) y podrá estar formado por un único campo o por un grupo de campos.

Una vez definida la clave podremos pasar a estudiar la primera forma normal.

### **Primera forma normal (1NF)**

Se dice que una tabla se encuentra en primera forma normal (1NF) si y solo si cada uno de los campos contiene un único valor para un registro determinado.

Primer Nivel de Formalización/Normalización (F/N)

- Eliminar los grupos repetitivos de las tablas individuales. Crear una tabla separada para cada grupo de datos relacionados.
- Identificar cada grupo de datos relacionados con una clave primaria.

### Segunda forma normal (2NF)

La segunda forma normal compara todos y cada uno de los campos de la tabla con la clave definida. Si todo los campos dependen directamente de la clave se dice que la tabla está en segunda forma normal (2NF).

#### Segundo Nivel de Formalización/Normalización (F/N)

- Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
- Relacionar estas tablas mediante una clave externa.

### Tercera forma normal (3NF)

Se dice que una tabla está en tercer forma normal si y solo si los campos de la tabla dependen únicamente de la clave, en otras palabras, los campos de las tablas no dependen unos de otros.

#### Tercer Nivel de Formalización/Normalización (F/N)

- Eliminar aquellos campos que no dependen de la clave.

### Cuarta forma normal (4NF)

Una tabla está en cuarta forma normal si y sólo si para cualquier combinación clave-campo no existen valores duplicados.

#### Cuarto Nivel de Formalización/Normalización (F/N)

- En las relaciones muchos a muchos, las entidades independientes no pueden ser almacenadas en la misma tabla.



### 3.4. Diseño de Base de Datos

La creación o diseño físico de la Base de Datos es el proceso de producir una descripción de la implementación de la Base de Datos en un tipo de almacenamiento. Describe también las relaciones base, las estructuras y los métodos de acceso usados para acceder a los datos de forma efectiva.

El proceso inicial del Diseño Físico es la traducción del modelo de datos lógico en una forma que pueda ser implementada para el manejador de la base de datos.

El siguiente paso es analizar las transacciones que se ejecutaran en la base de datos, elección de la organización basada en el análisis, añadir índices, introducir y controlar la redundancia para mejorar el funcionamiento.

El diseño físico de una base de datos nos permite lo siguiente:

- Facilitar a los usuarios de la base la obtención de los datos
- Conseguir que los datos estén siempre disponibles
- Optimizar el rendimiento de los recursos dedicados al sistema

En el diseño de la base de datos se deben desarrollar los programas y procedimientos necesarios de forma que cuando se vaya cargando la base, los distintos conjuntos de información se puedan ir probando y una vez que algunos ficheros están cargados, comenzar las pruebas y medir su rendimiento.

Para el diseño físico de la base de datos se pueden manejar dos etapas:

- La primera etapa; consiste en adaptar la estructura conceptual a las exigencias y restricciones del modelo de datos propio del manejador de base de datos que vaya a ser utilizado.
- La segunda etapa; es obtener el esquema almacenado mediante especificaciones que tiendan a conseguir la máxima eficiencia.

El punto de partida del diseño físico es el modelo de datos y la documentación que describe el modelo creado, comenzando por refinar el modelo conceptual y después utilizándolo para derivar un grupo de relaciones. El modelo y la relación derivada de este fueron validados utilizando las reglas de normalización.

```
DROP table nivel_Alumnos;
DROP table nivel_Estudio;
DROP table nombramiento;
DROP table plantel;
DROP table profesor;
DROP table telefono;
DROP table tesis;
DROP table tesis_Terminada;
DROP table tipo_Inves;
DROP table tipo_Jefe;
DROP table tipo_Profesor;
```

```
DROP DATABASE TESIS_OSIRIS;
```

```
CREATE DATABASE IF NOT EXISTS TESIS_OSIRIS;
USE TESIS_OSIRIS;
```

```
CREATE TABLE administrador (
    rfc CHAR(10) NOT NULL,
    login VARCHAR(15) NOT NULL,
    password VARCHAR(15) NOT NULL,
    nombre_Admin VARCHAR (80) NOT NULL,
    descripcion VARCHAR (250),
    PRIMARY KEY (rfc)
);
```

```
CREATE TABLE alumno (
    num_Cuenta CHAR (9) NOT NULL,
    rfc_Alumno CHAR (10) NOT NULL,
    nombre_alumno VARCHAR (40) NOT NULL,
    apellido_P VARCHAR (40) NOT NULL,
    apellido_M VARCHAR (40) NOT NULL,
    sexo ENUM ('Hombre','Mujer') DEFAULT 'Hombre' NOT NULL,
    calle VARCHAR (4) NOT NULL,
    numero_Ext INT (4) NOT NULL,
    numero_Int INT (4),
```

### 3.4.1. Restricciones de integridad

En el mundo real existen ciertas restricciones que deben cumplir los elementos en él existente; por ejemplo, una persona solo puede tener un número de identificación (CURP); y una única dirección oficial. Cuando se diseña una base de datos se debe reflejar fielmente la situación que estamos tratando, lo que es lo mismo, reflejar las restricciones existentes en el mundo real.

En general, se puede decir que existen tres tipos de reglas de Integridad:

- **Integridad de dominio:** restringimos los valores que pueden tomar un atributo respecto de su dominio, por ejemplo: Edad > 18 y < 65.
- **Integridad de entidad:** la clave primaria de una entidad no puede tener valores nulos y siempre deberá ser única.
- **Integridad referencial:** cuando existe una relación entre dos tablas, se crea una llave foránea, la cual deberá siempre estar correspondida con la llave primaria de la otra tabla.

### 3.4.2. Código para generar la Base de Datos

```
USE TESIS_OSIRIS;
```

```
DROP table administrador;
```

```
DROP table alumno;
```

```
DROP table area;
```

```
DROP table area_Graduados;
```

```
DROP table beca;
```

```
DROP table campo_Investigacion;
```

```
DROP table carrera;
```

```
DROP table catalogo_Tesis;
```

```
DROP table delegacion;
```

```
DROP table departamento;
```

```
DROP table estado;
```

```
DROP table fecha_Tesis;
```

```
DROP table instituto;
```

```
DROP table jefe;
```

```
DROP table nacionalidad;
```

);

```
CREATE TABLE campo_Investigacion (  
    id_CampoInv CHAR (4) NOT NULL,  
    nombre_campo VARCHAR (80) NOT NULL,  
    descripcion TEXT NOT NULL,  
    PRIMARY KEY (id_CampoInv)
```

);

```
CREATE TABLE carrera (  
    id_Carrera CHAR (4) NOT NULL,  
    nombre_carrera VARCHAR (80) NOT NULL,  
    id_Plantel CHAR(4) NOT NULL,  
    PRIMARY KEY (id_Carrera,id_Plantel),  
    FOREIGN KEY (id_Plantel) REFERENCES plantel
```

);

```
CREATE TABLE catalogo_Tesis (  
    clave INT NOT NULL AUTO_INCREMENT,  
    NoRegistro CHAR (7) NOT NULL,  
    NoCuenta CHAR(9) NOT NULL,  
    Edo_Alumno ENUM ('Alta','Baja') DEFAULT 'Alta',  
    PRIMARY KEY (clave),  
    FOREIGN KEY (NoRegistro, NoCuenta) REFERENCES Tesis
```

);

```
CREATE TABLE delegacion (  
    id_Delegacion INT (2) NOT NULL,  
    nombre_delegacion VARCHAR (40) NOT NULL,  
    PRIMARY KEY (id_Delegacion)
```

);

```
CREATE TABLE departamento (  
    id_Depto CHAR (4) NOT NULL,  
    nombre_depto VARCHAR (80) NOT NULL,  
    numero_Tel INT (8) NOT NULL,  
    descripcion VARCHAR (250),
```

```

colonia VARCHAR (80) NOT NULL,
codigo_postal INT (6) NOT NULL,
email VARCHAR (80) NOT NULL,
Municipio VARCHAR (80) NOT NULL,
id_Estado CHAR (4) NOT NULL,
id_Delegacion CHAR (4) NOT NULL,
id_Nacionalidad CHAR (4) NOT NULL,
telefono VARCHAR (13) NOT NULL,
PRIMARY KEY (num_Cuenta),
FOREIGN KEY (id_Estado) REFERENCES estado,
FOREIGN KEY (id_Delegacion) REFERENCES delegacion,
FOREIGN KEY (id_Nacionalidad) REFERENCES nacionalidad
);

```

```

CREATE TABLE area (
    id_Area INT NOT NULL AUTO_INCREMENT,
    nombre_area VARCHAR (80) NOT NULL,
    descripcion VARCHAR (250) NOT NULL,
    Id_Depto CHAR (4) NOT NULL,
    PRIMARY KEY (id_Area,id_Depto),
    FOREIGN KEY (id_Depto) REFERENCES departamento
);

```

```

CREATE TABLE area_Graduados (
    clave_AGrad CHAR (4) NOT NULL,
    nombre_Graduados VARCHAR (80) NOT NULL,
    PRIMARY KEY (clave_AGrad)
);

```

```

CREATE TABLE beca (
    id_Beca INT NOT NULL AUTO_INCREMENT,
    becado ENUM ('Si','No') DEFAULT 'Si',
    institucion_Beca VARCHAR (80) NULL,
    descripcion VARCHAR (250),
    Num_Cuenta CHAR (9) NOT NULL,
    PRIMARY KEY (id_Beca),
    FOREIGN KEY (Num_Cuenta) REFERENCES alumno
);

```

```
id_Nacionalidad CHAR (2) NOT NULL,  
nombre_nacionalidad VARCHAR (80) NOT NULL,  
PRIMARY KEY (id_Nacionalidad)  
);
```

```
CREATE TABLE nivel_Alumnos (  
id_nivel_A CHAR (9) NOT NULL,  
f_ingreso CHAR (4) NOT NULL,  
f_egreso CHAR (4) NOT NULL,  
promedio CHAR (5) NOT NULL,  
avances_creditos CHAR (5) NOT NULL,  
tipo_nivel ENUM ('Licenciatura','POSGRADO') DEFAULT 'Licenciatura',  
tipo_Posgrado ENUM ('Maestria','Doctorado','Especialidad','NINGUNA') DEFAULT 'NINGUNA',  
id_Carrera CHAR (4) NOT NULL,  
id_Plantel CHAR (4) NOT NULL,  
Edo_Examen ENUM ('APROBADO','NO APROBADO','EN PROCESO') DEFAULT  
'EN PROCESO',  
beca ENUM ('Becado','No Becado'),  
PRIMARY KEY (id_nivel_A),  
FOREIGN KEY (id_Carrera, id_Plantel) REFERENCES carrera  
  
);
```

```
CREATE TABLE nivel_Estudio (  
clave_NEst CHAR (4) NOT NULL,  
nombre_Nivel VARCHAR (80) NOT NULL,  
PRIMARY KEY (clave_NEst)  
);
```

```
CREATE TABLE nombramiento (  
clave_Nombramiento CHAR (4) NOT NULL,  
nombre_nombramiento VARCHAR (80) NOT NULL,  
PRIMARY KEY (clave_Nombramiento)  
);
```

```
CREATE TABLE plantel (  

```

```
id_Jefe CHAR (2) NOT NULL,  
PRIMARY KEY (id_Depto),  
FOREIGN KEY (id_Jefe) REFERENCES Jefe  
);  
  
CREATE TABLE estado (  
    id_Edo INT (2) NOT NULL,  
    nombre_edo VARCHAR (40) NOT NULL,  
    PRIMARY KEY (id_Edo)  
);  
  
CREATE TABLE fecha_tesis (  
    id_FTesis INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    fecha_inicio CHAR(10) NOT NULL,  
    fecha_fin CHAR (10) NULL,  
    fecha_examen CHAR(10) NULL  
);  
  
CREATE TABLE instituto (  
    clave_Inst CHAR (4) NOT NULL,  
    nombre_instituto VARCHAR (100) NOT NULL,  
    Estado_Ubicacion VARCHAR (250) NOT NULL,  
    PRIMARY KEY (clave_Inst)  
);  
  
CREATE TABLE Jefe (  
    id_Jefe INT NOT NULL AUTO_INCREMENT,  
    tipo_Jefe CHAR (2) NOT NULL,  
    rfc_Prof CHAR (10) NOT NULL,  
    password VARCHAR (15) NOT NULL,  
    login VARCHAR (15) NOT NULL,  
    email VARCHAR (80),  
    PRIMARY KEY (id_Jefe),  
    FOREIGN KEY (tipo_Jefe) REFERENCES tipo_Jefe  
);  
  
CREATE TABLE nacionalidad (  

```

```

CREATE TABLE telefono (
    id_Telefono INT (4) NOT NULL,
    casa INT (8) NOT NULL,
    oficina INT (8),
    celular INT (13),
    otro INT (13),
    PRIMARY KEY (id_Telefono)
);

CREATE TABLE tesis (
    id_Tesis CHAR (7) NOT NULL,
    proyecto ENUM ('Tesis','Seminario') DEFAULT 'Tesis' NOT NULL,
    titulo VARCHAR (250) NOT NULL,
    objetivo VARCHAR (250) NOT NULL,
    temario TEXT NOT NULL,
    descripcion TEXT,
    campo_Inves VARCHAR (250),
    tipo_Inves VARCHAR (250),
    financiamiento VARCHAR (250),
    NO_Alumnos int (1) NOT NULL,
    avances CHAR (5) NOT NULL,
    ano VARCHAR (4) NOT NULL,
    semestre VARCHAR (2) NOT NULL,
    nivel ENUM ('Licenciatura','Posgrado') DEFAULT 'Licenciatura' NOT NULL,
    posgrado ENUM ('Ninguna','Maestria','Especialidad','Doctorado') DEFAULT 'Ninguna' NOT NULL,
    id_FTesis int (11) NOT NULL,
    Area VARCHAR (4) NOT NULL,
    Depto VARCHAR (4) NOT NULL,
    rfc VARCHAR(10) NOT NULL,
    PRIMARY KEY (id_Tesis),
    FOREIGN KEY (Depto) REFERENCES departamento,
    FOREIGN KEY (Area) REFERENCES area,
    FOREIGN KEY (rfc) REFERENCES profesor,
    FOREIGN KEY (campo_Inves) REFERENCES campo_Investigacion,
    FOREIGN KEY (tipo_Inves) REFERENCES tipo_Inves,
    FOREIGN KEY (id_FTesis) REFERENCES fecha_tesis
);

```

```

id_Plantel CHAR (4) NOT NULL,
nombre_plantel VARCHAR (80) NOT NULL,
PRIMARY KEY (id_Plantel)

```

```
);
```

```

CREATE TABLE profesor (
    rfc_Prof CHAR (11) NOT NULL,
    nombre_prof VARCHAR (40) NOT NULL,
    apellido_P VARCHAR (80) NOT NULL,
    apellido_M VARCHAR (80) NOT NULL,
    sexo ENUM ('Hombre','Mujer') DEFAULT 'Hombre',
    calle INT (4) NOT NULL,
    numero_Ext INT (4) NOT NULL,
    numero_Int INT (4),
    colonia VARCHAR (80) NOT NULL,
    codigopostal INT (6) NOT NULL,
    email VARCHAR (80),
    Municipio VARCHAR (80) NULL,
    id_Estado CHAR (4) NOT NULL,
    id_Delegacion CHAR (4) NOT NULL,
    id_Nacionalidad CHAR (4) NOT NULL,
    id_Nivel_Estudio CHAR (4) NOT NULL,
    clave_AGrad CHAR (4) NOT NULL,
    id_Nombramiento CHAR (4) NOT NULL,
    id_Tipo_Profesor CHAR (4) NOT NULL,
    telefono CHAR (13) NOT NULL,
    password VARCHAR (15) NOT NULL,
    PRIMARY KEY (rfc_Prof),
    FOREIGN KEY (id_Estado) REFERENCES Estado,
    FOREIGN KEY (id_Delegacion) REFERENCES Delegacion,
    FOREIGN KEY (id_Nacionalidad) REFERENCES Nacionalidad,
    FOREIGN KEY (id_Nivel_Estudio) REFERENCES Nivel_Estudio,
    FOREIGN KEY (clave_AGrad) REFERENCES Area_Graduado,
    FOREIGN KEY (id_Nombramiento) REFERENCES Nombramiento,
    FOREIGN KEY (id_Tipo_Profesor) REFERENCES Tipo_Profesor

```

```
);
```

**ESTA TESIS NO SALE  
DE LA BIBLIOTECA**

```

CREATE TABLE propuestos (
    id_Propuestos INT NOT NULL AUTO_INCREMENT,
    titulo VARCHAR (250) NOT NULL,
    profesor VARCHAR (250) NOT NULL,
    mail VARCHAR (250) NOT NULL,
    telefono VARCHAR (13) NOT NULL,
    num_personas CHAR (1) NOT NULL,
    PRIMARY KEY (id_Propuestos)
);

```

### 3.5. Diccionario de Datos

El Diccionario de Datos (DD) dentro de un Manejador de Bases de Datos contiene el directorio de todos los elementos de Información en la base de datos y también su descripción.

Según Artre “Un diccionario de datos es un deposito central de información acerca de las entidades: los campos de datos que representan a las entidades, las relaciones entre éstas, sus orígenes, significados, usos y formatos de representación.”

La administración de los datos almacenados en un sistema manejador de bases de datos (SMBD) requiere a su vez de un sistema de administración de los elementos que definen a la base de datos y sus relaciones constitutivas. El Diccionario de Datos (DD) contiene la información acerca de los elementos que constituyen la base de datos (tablas, vistas, índices, entre otros), tales como sus atributos, su uso y sus definiciones estándar. Estos son almacenados también en una base de datos, la cual lleva el control de las relaciones usadas por el sistema para la administración de los datos, o sea, el *catálogo del sistema*.

El DD está constituido por el catálogo del sistema, el cual contiene las relaciones que describen la estructura de la base de datos. Como cualquier relación, el catálogo del sistema puede ser consultado por cualquier usuario, y se actualiza por si mismo cuando se crea un nuevo elemento en el sistema, ya sea una tabla, una vista, un índice o inclusive una base de datos. La interacción entre los procesos que constituyen el motor del SMBD y el DD es muy estrecha, ya que es necesaria la consulta y actualización del catálogo del sistema cada vez que se crea o modifica un objeto del sistema.

```
CREATE TABLE tesis_Terminada (  
    id_TesisTer CHAR (4) NOT NULL,  
    rfc_Prof CHAR (4) NOT NULL,  
    num_Cuenta CHAR (9) NOT NULL,  
    registro CHAR (6) NOT NULL,  
    F_Titulo DATE,  
    edo_Alum ENUM ('Alta','Baja') DEFAULT 'Alta' NOT NULL,  
    nivel ENUM ('Licenciatura','Posgrado') DEFAULT 'Licenciatura' NOT NULL,  
    posgrado ENUM('Ninguna','Maestria','Especialidad','Doctorado') DEFAULT 'Ninguna' NOT NULL,  
    estado_Examen ENUM ('Aprobado','No aprobado') DEFAULT 'Aprobado' NOT NULL,  
    ano CHAR (4) NOT NULL,  
    semestre CHAR (2) NOT NULL,  
    PRIMARY KEY (id_TesisTer),  
    FOREIGN KEY (rfc_Prof) REFERENCES profesor,  
    FOREIGN KEY (num_Cuenta) REFERENCES alumno,  
    FOREIGN KEY (registro) REFERENCES tesis  
);
```

```
CREATE TABLE tipo_Inves (  
    id_TipoInv CHAR (4) NOT NULL,  
    nombre_Inves VARCHAR (80) NOT NULL,  
    descripcion TEXT,  
    PRIMARY KEY (id_TipoInv)  
);
```

```
CREATE TABLE tipo_Jefe (  
    tipo_Jefe CHAR (2) NOT NULL,  
    tipo VARCHAR (80) NOT NULL,  
    PRIMARY KEY (tipo_Jefe)  
);
```

```
CREATE TABLE tipo_Profesor (  
    clave_TProf CHAR (4) NOT NULL,  
    nombre_TProf VARCHAR (80) NOT NULL,  
    descripcion VARCHAR (80),  
    PRIMARY KEY (clave_TProf)  
);
```

**TABLA DE AREAS:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Área	1	PK	Id_Area	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN
	3		Descripción	VARCHAR	80	NN
	4	PK,PF	id_Depto	CHAR	4	NN
	5	PF	id_Jefe	CHAR	4	NN

**TABLAS DE NACIONALIDAD:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Nación	1	PK	id_Nacionalidad	CHAR	2	NN
	2		Nombre	VARCHAR	80	NN

**ESTADO:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Estado	1	PK	id_Estado	INT	2	NN
	2		Nombre	VARCHAR	40	NN

**TABLA DE DELEGACIONES:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Delegación	1	PK	id_Delegación	CHAR	4	NN
	2		Nombre	VARCHAR	4	NN

**TABLA DE TELEFONOS:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Telefono	1	PK	id_Telefono	INT	6	NN
	2		casa	INT	8	NN
	3		oficina	INT	8	N
	4		celular	INT	13	N
	5		otro	INT	13	N

**BECA:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NULO
Beca	1	PK	Id_Beca	CHAR	4	NN
	2		Becado (SI o NO)	VARCHAR		NN
	3		Institución_Beca	VARCHAR	80	NN
	4		Descripción	VARCHAR	80	N
	5	PF	Num_Cuenta	CHAR	9	NN

Entre las características representativas del DD deben estar:

- Presentar un rápido acceso a la base de datos
- Capacidad de generación de directorio
- Lenguaje de definición de datos
- Control de acceso y seguridad de datos
- Independencia a los dispositivos
- Capacidades de localización de datos

A continuación se muestran las tablas que conforman nuestro diccionario de datos

**TABLA DE AMINIDSTRADORES:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Administrador	1	PK	RFC	CHAR	10	NN
	2		Login	VARCHAR	15	NN
	3		Password	VARCHAR	15	NN
	4		Nombre	VARCHAR	80	NN
	5		Descripción	VARCHAR	250	NN

**TABLA DE JEFES:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Jefe	1	PK	Id_Jefe	CHAR	2	NN
	2		Password	VARCHAR	15	NN
	3		Login	VARCHAR	15	NN
	4		Email	VARCHAR	80	NN
	5	PF	tipo_Jefe	CHAR	2	NN
	6	PF	RFC_Prof	CHAR	10	NN

**TABLA TIPOS DE JEFE :**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Tipo_Jefe	1	PK	tipo_Jefe	CHAR	2	NN
	2		Tipo	VARCAHR	80	NN

**TABLA DE DEPARTAMENTOS:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Departamento	1	PK	id_Depto	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN
	3		Numero_Tel	INT	8	NN
	4		Descripción	VARCHAR	80	NN
	5	PF	Id_Jefe	CHAR	2	NN

TABLA DE ALUMNOS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Alumno	1	PK	NoCuenta	CHAR	9	NN
	2		RFC	CHAR	10	NN
	3		Nombre	VARCHAR	40	NN
	4		Apellido_P	VARCHAR	40	NN
	5		Apellido_M	VARCHAR	40	NN
	6		Sexo	ENUM("hombre","mujer")		NN
	7		Numero_EX	INT	4	NN
	8		Numero_lint	INT	4	N
	9		Calle	VARCHAR	80	NN
	10		Colonia	VARCHAR	80	NN
	11		Codigopostal	INT	6	NN
	12		Email	VARCHAR	40	N
	13		municipio	VARCHAR	80	N
	14	PF	Id_Estado	CHAR	4	NN
	15	PF	Id_Delegación	CHAR	3	NN
	16	PF	Id_Nacionalidad	CHAR	4	NN
	17		TeleFono	CHAR	8	NN
	18		Edo_Examen	Enum("En proceso","Aprobado","No aprobado")		NN

TABLA FECHAS DE TESIS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Fecha_Tesis	1	PK	Id_FTesis	CHAR	4	NN
	2		F_Inicio	CHAR	10	NN
	3		F_Fin	CHAR	10	NN
	4		F_Examen	CHAR	10	N
	5					

TABLA DE CAMPO DE INVESTIGACIONES:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Campo_Investigación	1	PK	Id_CampoInv	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN
	3		Descripción	TEXT		NN

TABLA DE TIPO DE INVESTIGACIONES:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Tipo_Investigación	1	PK	Id_TipoInv	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN
	3		Descripción	TEXT		NN

## TABLA PLANTEL:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NULO
Carrera	1	PK	Id_Plantel	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN

## TABLA CARRERA:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NULO
Carrera	1	PK	Id_Carrera	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN
	3	PK,PF	Id_Plantel	CHAR	4	NN

## TABLA DE NIVEL\_ALUMNOS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NULO
Alumno_Nivel	1	PK,PF	id_Nivel_A	CHAR	4	NN
	2		Id_Carrera	CHAR	4	NN
	3		Id_Plantel	CHAR	4	NN
	4		num_Cuenta	CHAR	9	N
	5		F_ingreso	DATE		NN
	6		F_egreso	DATE		N
	7		promedio	FLOAT	5	NN
	8		avences credito	FLOAT	5	NN

## TABLA DE AREA GRADUADOS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Area_Graduados	1	PK	Clave_AGradde	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN

## TABLA DE NOMBRAMIENTOS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
NOMBRAMIENTO	1	PK	Clave_Nombramientodes	CHAR	4	NN
	2		Nombre	VARCHAR	80	NN

## TABLA TIPO PROFESORES:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
TIPO_PORF	1	PK	Clave	CHAR	4	NN
	2		Nombre (tipo)	VARCHAR	80	NN
	3		Descripción	VARCHAR	80	N

**TABLA DE CATALOGO TESIS:**

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
CATALOGO_TESIS	1	PK	clave	AUTO_INCREMENT	4	NN
	2	PF	Rfc_prof	CHAR	10	NN
	3	PF	Num_Cuenta	CHAR	9	NN
	4	PF	Id_Tesis	CHAR	7	NN
	5	PF	ID_Area	CHAR	4	NN
	6	PF	Id_Depto	CHAR	4	NN
	7		EdoAlumno	Enum('Alta','Baja')		NN

**TABLA DE TESIS TERMINADAS:**

TABLA	CAMPO	LLAVE	NOMBRE TRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
TESIS_TERMINADA	1	PK	Id_TesisTer	CHAR	4	NN
	2	PF	Rfc_prof	CHAR	10	NN
	3	PF	Num_Cuenta	CHAR	9	NN
	4	PF	Id_Tesis	CHAR	6	NN
	5		F_titulacion	DATE		NN
	6		Nivel (L O P)	ENUM		NN

TABLA DE PROFESORES:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Profesor	1	PK	RFC_Prof	CHAR	10	NN
	2		Nombre	VARCHAR	40	NN
	3		Apellido_P	VARCHAR	40	NN
	4		Apellido_M	VARCHAR	40	NN
	5		Sexo	ENUM("HOMBRE",		NN
	6		Calle	" MUJER")	30	NN
	7		Colonia	VARCHAR	80	NN
	8		Codigopostal	VARCHAR	6	NN
	9		Numero_Ex	VARCHAR	4	NN
	10		Numero_In	int	4	N
	11		municipio	int	30	N
	12		Email	VARCHAR	80	NN
	13		Telefono	VARCHAR	8	NN
	14		password	CHAR	15	NN
	15	PF	Id_Estado	VARCHAR	15	NN
	16	PF	Id_Delegacion	CHAR	4	NN
	17	PF	Id_Nacionalidad	CHAR	4	NN
	18	PF	Clave_Nivel	CHAR	4	NN
	19	PF	Clave_Area_Graduado	CHAR	4	NN
	20	PF	Clave_Nombramiento	CHAR	4	NN
	21	PF	Clave_Tipo_Prof	CHAR	4	NN
		PF	Id_Area	CHAR	4	NN

TABLA TESIS:

TABLA	CAMPO	LLAVE	NOMBRE ATRIBUTO	TIPO	LONGITUD	NO NULO (NN) NULO (N)
Tesis	1	PK	Id_Tesis	CHAR	6	NN
	2		Proyecto	Enum('Tesis', 'Seminario')		NN
	3		Titulo	VARCHAR	250	NN
	4		Objetivo	TETX	250	NN
	5		Temario (Indice)	CHAR		NN
	6		Descripción	CHAR	250	NN
	7		No_Alumnos	INT	1	NN
	8		Financiamiento	VARCHAR	250	NN
	9		Avances	CHAR	4	NN
	10		Año	CHAR	4	NN
	11		Semestre	CHAR	4	NN
	12		Nivel	Enum('Licenciatura'		NN
	13		Posgrado	,'Posgrado')		
	14	PF	Id_CampoInv	Enum('Ninguna', 'Maestría',	4	NN
	15	PF	Id_TipoInv	'Especialidad', 'Doctorado')	4	NN
	16	PF	Id_Fecha	CHAR	4	NN
			CHAR			

#### 4.1. Descripción de los procesos

La descripción de procesos es una declaración formal que usa técnicas y lenguajes que permiten describir actividades importantes que forman parte del sistema.

Antes de comenzar con la descripción de los procesos, es importante mencionar que es un proceso. **“Un proceso es un conjunto ordenado de tareas, una serie de pasos que involucran actividades, restricciones y recursos que producen una determinada salida esperada”.**<sup>14</sup>

Cualquier proceso tiene las siguientes características:

- El proceso establece todas las principales actividades.
- El proceso puede estar compuesto por subprocesos.
- Cada actividad del proceso tiene entradas y salidas.

Es en esta etapa en la cual llevaremos a cabo los procesos necesarios para poder concluir el sistema y que esté finalmente sea útil para los usuarios.

Una vez que tenemos nuestras tablas y las relaciones entre estas podemos empezar a elaborar los procesos que permitirán que interactúen los usuarios con el sistema, de tal manera que estos puedan hacer consultas, inserción, actualización y borrado de datos. Esto dependiendo de los permisos establecidos para cada tipo de usuario. Cada proceso que se lleve a cabo será visto por el usuario como una pantalla con la cual podrá interactuar. A continuación serán descritas las pantallas que representan los procesos más importantes del sistema.<sup>15</sup>

---

<sup>14</sup> Ingeniería de Software, Shari Lawrence Pfleeger, Pág.52

<sup>15</sup> Para mayor información acerca del código de los procesos del sistema consulte el Apéndice II y para ver más detalle de las pantallas consulte el manual del usuario.

# *CAPITULO IV.*

## Desarrollo del Sistema

Es en esta fase es donde nos centraremos en la implementación de la solución como software. Es decir, es ahora donde escribiremos los programas que implementen el diseño y es precisamente ahora que tenemos que sacar beneficio de las características establecidas en la fase de diseño y de la estructura de datos, para hacer la construcción con un lenguaje de programación.

Después de realizado todo el código y de su implementación, es necesaria una etapa de pruebas, mismas que permitirán detectar algunos errores cometidos durante la programación

La fase de desarrollo, consiste en realizar la programación del sistema, que consiste en trasladar el diseño a un lenguaje de programación seleccionado. Desarrollando cada módulo con su respectivo grupo de pantallas (entradas y salidas), y su correspondiente ayuda.

Si el usuario proporcionó la información correcta se le mostrará una pantalla en la cual se encontrarán las siguientes tres opciones (ver Figura 4.3.):

**Datos personales:** Muestra el nombre, RFC, dirección, teléfono y e-mail.

**Datos académicos:** Proporciona el número de cuenta, año de egreso e ingreso, promedio, etc.

**Tesis:** Muestra la información de la tesis que el alumno está elaborando.

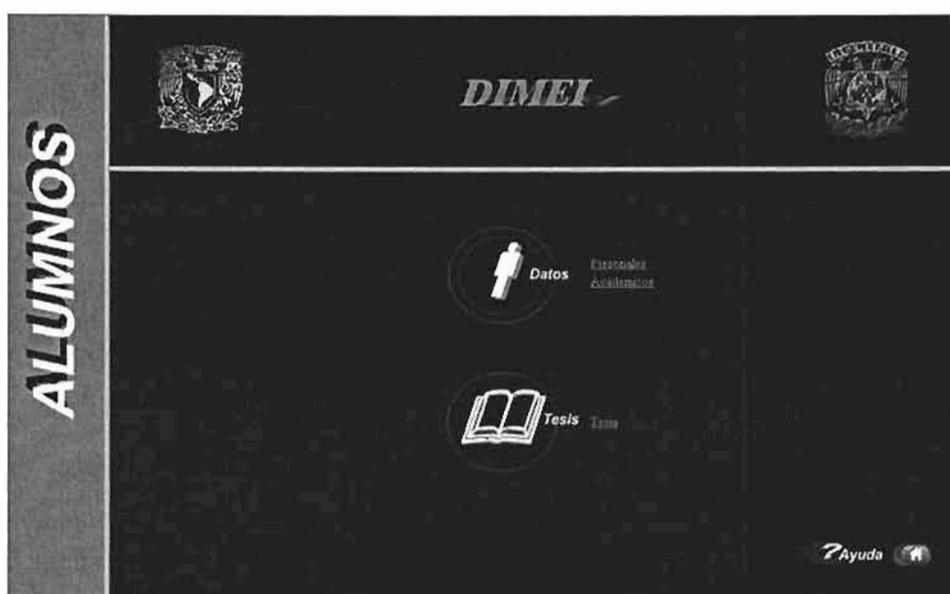


Figura 4.3.

### Usuario de tipo profesor

Al igual que en el módulo anterior en este se pedirá que el profesor escriba información que le de acceso al sistema, dicha información será su RFC y su Password, este último será proporcionado por el administrador del sistema. Véase Figura 4.4.

Al ingresar al sistema el usuario se encontrará con una pantalla en la cual puede escoger entre cinco opciones existentes de acuerdo al tipo de usuario al cual corresponda. Véase Figura 4.1.



Figura 4.1.

#### Usuario de tipo alumno

Al ingresar en este módulo se le pedirá al alumno que escriba su número de cuenta y su RFC, mismos que serán validados por el sistema y verificados en la base de datos, de ser correcta la información se dará acceso a la siguiente pantalla, en caso contrario se mandará un mensaje de error para que se proporcionen los datos correctos. Véase Figura 4.2.

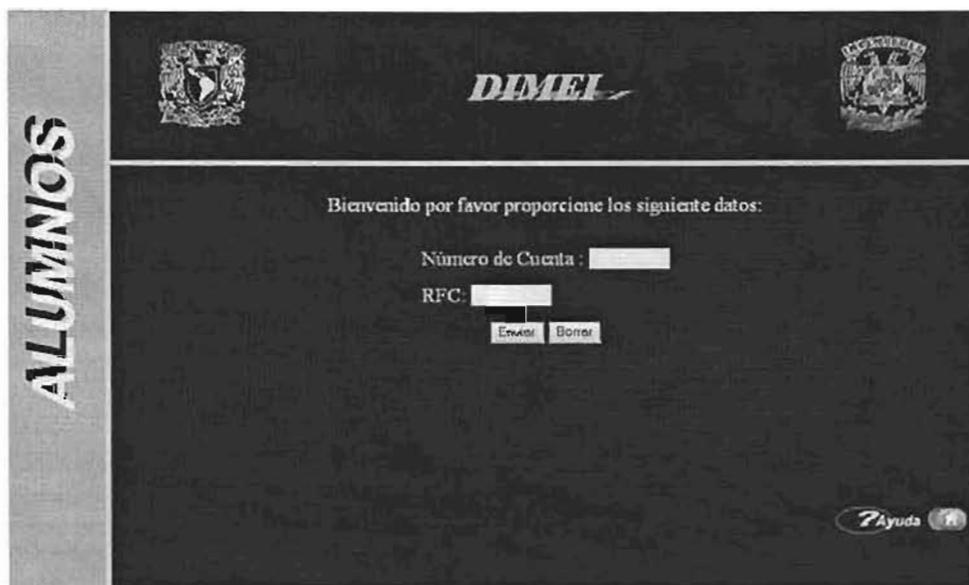


Figura 4.2.

Si el usuario proporcionó la información correcta se le mostrará una pantalla en la cual se encontrarán las siguientes tres opciones (ver Figura 4.3.):

**Datos personales:** Muestra el nombre, RFC, dirección, teléfono y e-mail.

**Datos académicos:** Proporciona el número de cuenta, año de egreso e ingreso, promedio, etc.

**Tesis:** Muestra la información de la tesis que el alumno está elaborando.

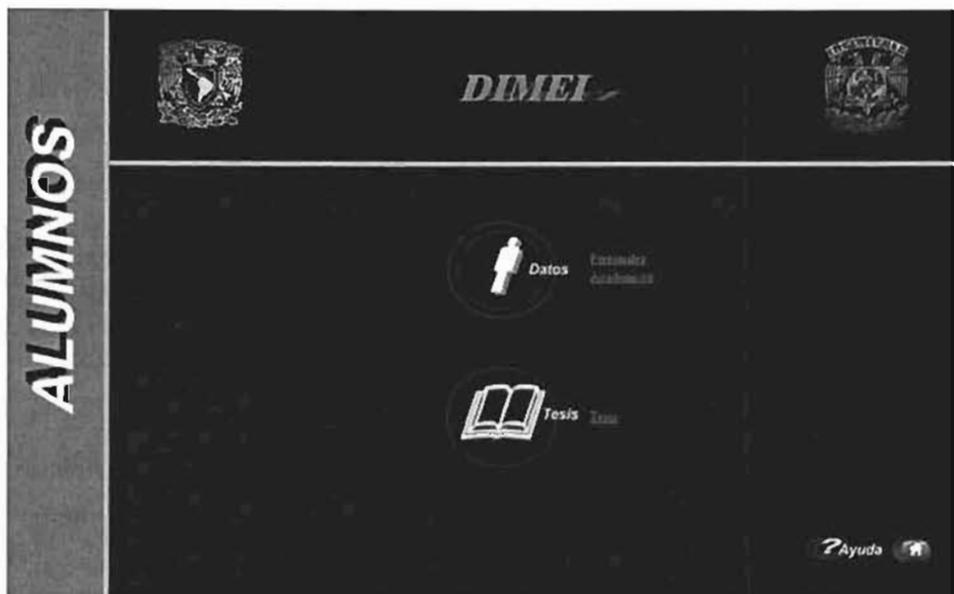


Figura 4.3.

### Usuario de tipo profesor

Al igual que en el módulo anterior en este se pedirá que el profesor escriba información que le de acceso al sistema, dicha información será su RFC y su Password, este último será proporcionado por el administrador del sistema. Véase Figura 4.4.



Figura 4.4.

Una vez verificados sus datos podrá ver la pantalla en la cual puede optar por cualquiera de las siguientes opciones (Ver Figura 4.5):

**Datos personales:** Muestra el nombre, RFC, dirección, teléfono y e-mail

**Datos académicos:** Proporciona el nivel de estudios, área de graduación, nombramiento y tipo de profesor.

**Tesis:** Muestra una lista de las tesis que se han dirigido o se están dirigiendo, pudiendo seleccionar aquella de la cual se desea saber más información.

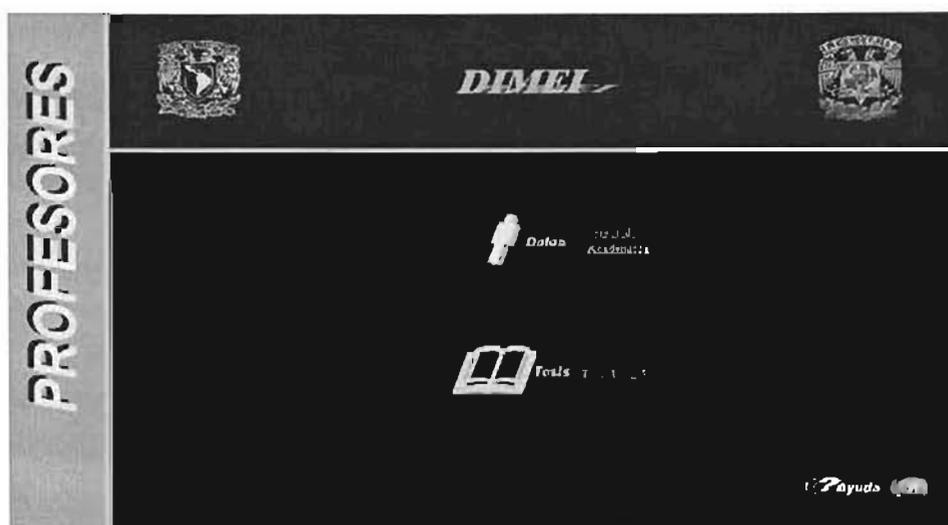


Figura 4.5.

### Usuario de tipo jefe de departamento

Para que un jefe de departamento pueda ingresar al sistema para hacer consultas se requiere que proporcione su RFC, password, login y el nombre del departamento al que pertenece, mismo que podrá ser seleccionado de una lista desplegable (ver Figura 4.6.). Una vez verificada la información se presentará una pantalla para que el usuario pueda realizar consultas acerca de las tesis que se están elaborando en ese departamento. Dichas tesis podrán ser consultadas por el número de cuenta del alumno, el RFC del profesor, el área o mostrar todas las correspondientes al departamento.



JEFE DE DEPTO.

 **DIMEL** 

Bienvenido por favor proporcione los siguiente datos:

RFC:

Login:

Password:

Departamento

[Ayuda](#) 

Figura 4.6.

### Usuario de tipo Administrador

Esté usuario es el que mantendrá constantemente la información actualizada, por ello es necesario que tenga acceso a todo. Para poder ingresar al sistema deberá proporcionar su RFC, login y password (ver Figura 4.7):

Figura 4.7

Una vez que la información de acceso ha sido confirmada, aparecerá una pantalla en la que el administrador podrá seleccionar entre la actividad que desea llevar a cabo. Esta actividad puede ser: una alta, una baja, un cambio o una consulta (ver Figura 4.8.).

Figura 4.8.

Si el administrador decide hacer una alta, podrá escoger entre alumnos, profesores, proyectos de tesis, propuestas de tesis u otros. Por ejemplo, a continuación mostramos el formulario para dar de alta una tesis (ver Figura 4.9.).

The screenshot shows the 'ADMINISTRADOR' interface for DIMEL. The main heading is 'Proporcione los Datos para el registro de Proyectos'. The form includes the following fields:

- Projecto: Tesis (dropdown)
- Departamento: Ingeniería Mecánica (dropdown)
- Área: Manufactura (dropdown)
- Núm. de Alumnos: 1 (dropdown)
- Nivel: Licenciatura (dropdown)
- Proyecto: Ninguno (dropdown)
- Título del Proyecto (\*): [text input]
- Objetivo del Proyecto (\*): [text input]
- Clave del Registro (\*): [text input]
- EFC de Director (\*): [text input]
- Tema: [text input]
- Comentarios: [text input]
- Tipo de Investigación: [text input]
- Campo de Investigación: [text input]
- Financiamiento: [text input]
- Avance: [text input]
- Fechas:
  - Fecha de Registro del Proyecto (\*): 01 Enero 2006 (dropdowns)
  - Fecha Fin del Proyecto (\*): 01 Enero 2006 (dropdowns)

Buttons for 'Enviar' and 'Borrar' are located at the bottom of the form.

Figura 4.9.

En caso de que se desee llevar a cabo una consulta, podrá escoger entre alumnos, profesores, proyectos de tesis, propuestas de tesis u otros. En el caso particular de las consultas para tesis se puede escoger entre varias opciones (ver Figura 4.10.).

The screenshot shows the 'ADMINISTRADOR' interface for DIMEL. The main heading is 'Elija una opción a realizar'. Below this, there is a table with the following structure:

OPCIÓN	MOSTRAR DATOS DE TESIS
<input type="radio"/>	Por Tesis por año o semestre
<input type="radio"/>	Por EFC del Director
<input type="radio"/>	Por Número de cuenta del Alumno
<input type="radio"/>	Por Departamento o Área de estudio
<input type="radio"/>	Por Licenciatura o Postgrado
<input type="radio"/>	Por número de registro

A 'Selección' button is located below the table.

Figura 4.1.0.

Al seleccionar alguna de ellas se nos presentará la información. Por ejemplo al hacer una consulta por el número de registro de una tesis, el sistema nos mostrará una pantalla con la información correspondiente a la tesis con ese número de registro (ver Figura 4.11.).

**ADMINISTRADOR**

**DIMEL**

**Datos de la tesis 0402001**

Número de Registro : 0402001	Proyecto : Tesis	Nivel : Licenciatura	Posgrado :	Año : 2004	Semestre : 1º
Título de la tesis : Prueba					
Objetivo : Prueba					
Fecha de inicio o registro : 01/11/2004	Fecha fin del Proyecto : 31/12/2004	Avances : 0	Número de alumnos : 0		

**Datos del director de tesis**

Director : Angel Gabriel Merced	Departamento : Computación	Correo : amerced@unhcrf.com	RFC : 300776127
---------------------------------	----------------------------	-----------------------------	-----------------

**Datos de los alumnos registrados en la tesis**

Número de cuenta : 00000000	Nombre : Marco Luna Valle	Correo : mluna@unhcrf.com	Estado del alumno : Baja
Carrera : Ingeniería Industrial	Institución : Facultad de Ingeniería	Año de ingreso : 1999	Año de egreso : 2004
Número de cuenta : 00001234	Nombre : Ana Reyes Alvarez	Correo : areyes@unhcrf.com	Estado del alumno : Baja

Figura 4.11.

Cuando lo que se requiere hacer es un cambio de la información ya almacenada en la base de datos, correspondiente a alumnos, profesores o proyectos de Tesis, el administrador deberá proporcionar la información solicitada en una pantalla como la de la Figura 4.12.

**ADMINISTRADOR**

**DIMEL**

Cambios

Escribe el RFC

Busca el RFC del Director de tesis

Borrar Actualizar Profesor

Escribe el número de cuenta

Busca el número de cuenta

borrar Actualizar Alumno

Escribe la clave de la tesis

Busca la clave de la tesis

borrar Actualizar Tesis

Figura 4.12.

Por ejemplo en el caso particular de requerir un cambio de información en el registro de un alumno podemos buscar la información por el RFC, si se encuentra la información requerida en la base de datos se mostrará una pantalla que contiene casillas que hacen referencia a cada campo de la tabla de la base de datos. Véase Figura 4.13.

**ADMINISTRADOR**

**DIMEI**

Selecciona los campos que desea actualizar IIMA770528

Datos personales		
<input type="checkbox"/> Nombre: Angel	<input type="checkbox"/> Apellido Paterno: Indio	<input type="checkbox"/> Apellido Materno: Mercado
<input type="checkbox"/> Sexo: Hombre	<input type="checkbox"/> Nacionalidad: Mexicana	<input type="checkbox"/> RFC: IIMA770528
Datos Académicos		
<input type="checkbox"/> Nivel de estudios: Licenciatura	<input type="checkbox"/> Área de graduación: Ingeniería en Computación	<input type="checkbox"/> Ambiente: Profesor de Tiempo completo
<input type="checkbox"/> Tipo de profesor: Profesor tipo A	<input type="checkbox"/> Password: angelito	<input type="checkbox"/> Email: angelbmercado@terra.com.mx
Domicilio		
<input type="checkbox"/> Calle: 0	<input type="checkbox"/> Número Exterior: 37	<input type="checkbox"/> Número Interior: 0
<input type="checkbox"/> Colonia: Las Cruces	<input type="checkbox"/> C.P.: 10330	<input type="checkbox"/> Estado: Distrito Federal
<input type="checkbox"/> Delegación: Magdalena Contreras	<input type="checkbox"/> Municipio:	<input type="checkbox"/> Teléfono: 54252012

Seleccionar

Ayuda

Figura 4.13.

En esta pantalla se podrá seleccionar aquella información que se necesita cambiar, puede ser desde un solo campo o todos los campos.

## 4.2. Pruebas e implementación

### 4.2.1. Implementación

Esta es la última fase del desarrollo del sistema y es el proceso de instalar el software nuevo. Al hacer la implementación lo primero que tenemos que asegurar es que el sistema sea operacional, es decir que funciona de acuerdo a los requerimientos del análisis y que permitir a los usuarios operarlo.

La implementación del sistema es de gran importancia; ya que aunque el sistema este bien diseñado y desarrollado correctamente su éxito depende de su implantación y ejecución, por lo que es importante capacitar al usuario con respecto a su uso y mantenimiento.

Para instalar el nuevo software en el equipo de cómputo de la DIMEI, se requiere construir un proceso en donde se combinen los elementos que integran al sistema para ejecutarse en una

configuración particular, esto involucra la compilación de elementos o el ligado de programas, donde se instalará el código.

Para planear el proceso de instalación se consideraron algunos factores que a continuación se describen:

- Tener todos los componentes que integran el sistema, además de un proceso de instalación y todas las rutas o archivos que se van a instalar.
- Verificar que se tengan las versiones apropiadas de cada componente requerido para la instalación.
- Disponer de todos los datos requeridos.
- Los archivos de datos que hagan referencia con algún componente del sistema, deben tener el mismo nombre en la máquina donde se ejecutará el sistema.
- Revisar las versiones apropiadas de las herramientas necesarias, ya que las versiones existentes pueden ser incompatibles con la versión en que fue creado el sistema.

#### 4.2.2. Propuesta de implantación

Dado que el objetivo a alcanzar consiste en la automatización del registro en computadora de la información, que se tiene de proyectos de tesis, directores y alumnos en la **Coordinación de titulación de la DIMEI**, y siendo que esta información tendrá que ser manejada a través de Internet, tanto para consultas como para actualización, entonces, partimos de que para realizar el presente trabajo, se aprovechó toda la infraestructura tecnológica de la **UNAM** en cuanto a equipo de cómputo se refiere. Para esto a lo largo de los pasados capítulos se ha propuesto el uso de las tecnologías, tanto en hardware como en software, que más se adecua a las necesidades, para cumplir todas las expectativas que debe tener el sistema; por lo anterior, en términos generales nuestra propuesta es la siguiente:

- 1) Para alcanzar el objetivo, se planteó un sistema que esta bajo la arquitectura Cliente/Servidor, que contará con un origen de datos sobre un equipo que utiliza un sistema operativo Linux y Mysql como Administrador de Base de Datos Relacional y el protocolo TCP/IP de comunicación entre los clientes y el servidor. En cuanto el equipo cliente, se utilizará una interfase gráfica que puede utilizar cualquier navegador (Internet Explorer, Netscape, Mozilla, etc.). En este sistema, cuando el servidor recibe una solicitud de servicio, realiza el proceso y regresa el resultado al

cliente. De esta manera, el sistema se desempeñará como se describe a continuación: El servidor se encarga de ejecutar todo aquellos procesos que para el cliente podrían representar mucho tiempo de proceso, de tal forma que el servidor se encarga de realizar el trabajo pesado y el cliente se dedica a presentar los resultados a los usuarios finales, mediante la aplicación realizada en PHP. La premisa del sistema, está en que se aprovecha lo que ya se tiene, esto a su vez, trae como consecuencia que haya un mejor rendimiento.

- 2) Los siguientes puntos representan las características a detalle que debe cumplir el sistema.
  - a. **Manejo bajo un ambiente Cliente/Servidor.** Esto permite la correcta distribución de las funciones de una máquina potente (servidor) y otra menos potencial (cliente), esto es los datos deben residir en el servidor, quien puede procesarlos con mayor facilidad, mandando al cliente la información ya procesada y éste último, presentarla al usuario final, quien a su vez podrá consultarla en pantalla de acuerdo a sus necesidades.
  - b. **Acceso a la Información vía internet.** El sistema presentaría, una característica muy importante y especial, ésta es que la aplicación, debía correr sobre cualquier computadora cliente conectada a internet, con la única restricción de que el usuario cuenta con permisos para acceder al sistema.
  - c. **Medidas de seguridad en el acceso a la información.** La seguridad dentro del sistema juega un papel muy importante y debido a que el sistema trabaja en internet, la información de la base de datos está expuesta a todos los usuarios de internet a nivel mundial, por lo tanto no se descarta la idea de que alguien con malas intenciones intente filtrarse. Es por eso que el sistema cuenta con seguridad de acceso, controlando el nivel de acceso a los usuarios válidos. Esto es para poder entrar al sistema, se pide una clave y una contraseña, y dependiendo del tipo de clave es como el usuario va a poder hacer movimientos dentro del sistema.
  - d. **Medidas de seguridad para la integridad de la información.** En los niveles de integridad y consistencia, se protege la información a ser consultada o modificada; además de que el sistema se comporte en forma esperada, estos mecanismos suelen ser transparentes a los usuarios, ya que desde el diseño de la base de datos, se estructuro de tal manera, que fuera estable y lógica, para evitar redundancia de la información e inconstancia en los datos.

- e. **Actualización en línea.** Dado que el sistema puede ser accesado desde Internet, se hace indispensable que la actualización se lleve a cabo desde cualquier lugar, no importando donde se encuentre el usuario. A esto responde el sistema en forma satisfactoria, siempre y cuando se cumpla con las autenticaciones necesarias para poder acceder a la base de datos.

Realizados los pasos anteriores e implantado el sistema de información lo siguiente es asegurarse el buen funcionamiento del sistema. Para lo cual se llevan a cabo una serie de pruebas, siendo este nuestro siguiente tema.

### **4.3. Pruebas**

Durante esta fase, el sistema se emplea de manera experimental para asegurarse que el software no tenga fallas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan con entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados.

Cuando se desarrolla un sistema debe verificarse que cada parte cumpla con las especificaciones establecidas y validarse el sistema contra los requerimientos del usuario. La validación emplea un proceso de prueba. Las pruebas son un proceso necesario para la validación y para establecer la confiabilidad del producto y, con ello, contribuir a su calidad.

Las pruebas comienzan a nivel de módulo y trabajan hacia fuera, es decir hacia la integridad del sistema. Diferentes tipos de pruebas son apropiadas en cada caso para diferentes momentos. La prueba la lleva acabo el desarrollador del sistema y en el caso de grandes proyectos la realiza un grupo de prueba independiente.

#### **4.3.1 Niveles de Pruebas**

Existen diversos alcances para las pruebas de un sistema, que permiten agruparlas por nivel. Aún cuando no hay un acuerdo universal, los niveles más comunes son tres y se muestran en la Figura 4.14.

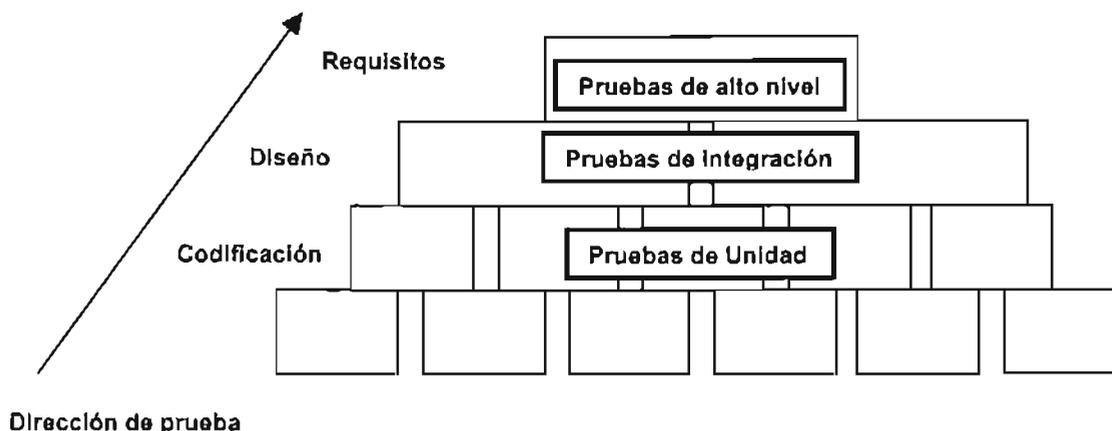


Fig. 4.14. Pasos en la prueba de Sistemas \*

**Pruebas de unidad:** Donde se prueban partes significativas del producto por separado. La unidad puede ser una subrutina, un módulo, una clase o un componente.

**Pruebas de Integración:** Es donde se va comprobando la interacción de unidades ya probadas. Es decir, la integración de todos los módulos para formar el sistema completo.

**Pruebas de Alto nivel:** Después de integrado todo el sistema se realizan un conjunto de pruebas de alto nivel, en donde se deben comprobar los criterios de validación. La prueba de validación debe proporcionar la seguridad de que el sistema satisface todos los requerimientos funcionales, de comportamiento y de requerimiento.

#### 4.3.2. Pruebas realizadas al sistema

##### a) Pruebas de unidad

La primera prueba realizada fue la prueba de unidad que se realiza sobre los procedimientos contenidos dentro de un módulo, cada uno de manera independiente, a fin de localizar posibles errores de codificación y lógica que estén contenidos en su código. Es en este momento cuando se detecta si está bien generada una llamada a un proceso almacenado en el servidor. Así como los datos que este último regresa.

\* Apuntes de Ingeniería de programación, Ing. Orlando Zaldívar Zamorategul, Pág. 152

En el caso de nuestro sistema, los primeros procesos que se probaron fueron los de acceso al sistema, para después dar paso a los de consulta, es decir todos aquellos que mostraban información directamente en la pantalla del usuario; posteriormente, se trabajo con aquellos que guardaban la información.

#### **b) Pruebas de integración**

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

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

#### **c) Pruebas de volumen**

Con las pruebas de volumen, se asegura el funcionamiento adecuado del sistema en condiciones de gran demanda de recursos en cantidad y frecuencia. Para este tipo de pruebas, tuvimos que trabajar con volúmenes de información cercanos a los que se piensa se pueden tener. Lo anterior se realizó en la **Unidad de Servicios de Cómputo Académico (UNICA)**. También se dio de alta información en cada uno de los módulos que conforman la aplicación, de igual forma, se realizaron consultas de determinada información del sistema.

#### **d) Pruebas de validación**

En esta prueba lo que se hizo fue validar la información correcta de cada uno de los módulos, es decir, se verifico que la información que se esta ingresando sea la que sistema realmente esta solicitando en cada uno de estos módulos y no la que el usuario cree que deba ir.

Por otra parte en el diseño mismo de las ventanas de aplicación, se proporciona una ayuda en línea, donde el usuario puede ver la forma en se debe escribir la información, especificando el tipo de dato que deberá aceptar el campo, por ejemplo, caracteres alfanuméricos, numéricos, fecha, etc., por lo tanto, el sistema mismo no permitirá que en un campo de fecha, se teclee un dato diferente.

Finalmente podemos decir que el objetivo de las pruebas es valorar y mejorar la calidad, siguiendo la idea de que la mejor manera de minimizar el número de errores en un programa es

controlarlos y eliminarlos durante el desarrollo del sistema mismo, y como hemos mencionado la confiabilidad de un sistema implica un conjunto de actividades que incluyen diversas pruebas que garantizan el apego a las especificaciones de requisitos, principios generales de facilidad en el manejo del sistema, estándares en el lenguaje de programación, documentación y expectativas del usuario.

# *CAPITULO V.*

## Mantenimiento

Esta fase tiene como objetivo corregir el mal funcionamiento de un sistema (mantenimiento correctivo) modificar el sistema como consecuencia de cambios externos, pero que tienen un efecto en nuestro sistema, por ejemplo, modificaciones realizadas que surgen a raíz de nuevas tecnologías (mantenimiento adaptativo) o bien la realización de mejoras funcionales no previstas en la fase de diseño funcional (mantenimiento perfectivo). Por lo que podemos decir que el mantenimiento de un sistema son todas aquellas modificaciones que se hacen luego de haber concluido el desarrollo y de poner en marcha el sistema.

Ahora que se ha concluido la realización del sistema y de que hemos superado las pruebas e implementación solo nos resta hablar del mantenimiento que permita asegurar el buen funcionamiento y la protección de la información que el sistema maneja. En este capítulo se contemplan revisiones periódicas al sistema, medidas preventivas, correctivas y nuevas tecnologías.

## **5.1. Revisiones periódicas**

El monitoreo permanente del sistema asegura que las necesidades de mantenimiento sean identificadas y satisfechas cuando resulte necesario. Cuando el sistema es de uso prolongado, se puede establecer un mecanismo para recibir retroalimentación de los usuarios como una forma efectiva para determinar las necesidades de mantenimiento y modificación.

A los sistemas se les debe de dar mantenimiento para asegurar que continúen operando en el nivel mostrado durante la época de prueba. Si los sistemas se deterioran, existe el riesgo de que no se desempeñen conforme a los estándares requeridos.

### **5.1.1. Resaldos de la base de datos**

El presente sistema almacena información importante de la DIMEI. Por este motivo, se llevaran a cabo respaldos por parte del administrador, asegurado de esta forma que la Base de Datos trabaje de manera eficiente. Estos mismos respaldos serán de gran utilidad para llevar una recuperación de información en caso de una contingencia.

### **5.1.2. Cambios en la lógica de programación**

Si requiere hacer un cambio en la lógica de programación se puede hacer un respaldo de la carpeta que contiene todos los archivos del sistema, lo que será de gran ayuda para el programador, dando como resultado que la tarea de mantenimiento no se vuelva un problema.

## **5.2. Mantenimiento correctivo, preventivo, adaptativo y perfectivo**

### **5.2.1 Mantenimiento correctivo**

Este tipo de mantenimiento responde a problemas que son resultado de defectos. A medida que ocurren las fallas, se ponen a consideración y se busca la causa de la falla, haciéndose las correcciones necesarias. A menudo este tipo de reparaciones es temporal, como para mantener el sistema funcionando, pero no el arreglo óptimo.

### 5.2.2. Mantenimiento preventivo

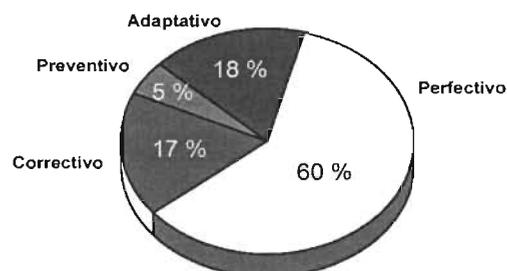
El mantenimiento preventivo involucra la modificación de algún aspecto del sistema a fin de prevenir las fallas. Por lo general el mantenimiento preventivo se realiza a partir del momento en el que se encuentra un defecto real o potencia, como todavía no se ha manifestado como falla, lo cual permite encarar las acciones necesarias para corregir el defecto antes de que se produzca el daño.

### 5.2.3. Mantenimiento adaptativo

A veces, la introducción de un cambio en una porción del sistema requiere la realización de cambios en otras de sus partes. El mantenimiento adaptativo es la aplicación de estos cambios secundarios. El mantenimiento adaptativo se hace para agregar un parámetro extra y no para corregir los defectos, es decir permite adaptar el sistema a medida que evoluciona. El mantenimiento adaptativo puede realizarse debido a cambios en el hardware o en el ambiente.

### 5.2.4. Mantenimiento perfectivo

A medida que se realiza el mantenimiento de un sistema, se examinan los documentos, el código y las pruebas, buscando oportunidades para la mejora. Por ejemplo, cuando se agregan funciones a un sistema. El mantenimiento perfectivo implica hacer cambios para mejorar algún aspecto del sistema, incluso cuando los cambios no son sugeridos por los defectos. Los cambios de la documentación para clasificar los puntos, el cambio de la sesión de pruebas para mejorar la cobertura de prueba, y las modificaciones de código y del diseño para reforzar la legibilidad son ejemplos de mantenimiento perfectivo. En la Figura 5.1. se observa la forma en la que se encuentra distribuido el mantenimiento para un sistema.



**Figura 5.1. Distribución de los esfuerzos de mantenimiento\***

\* Ingeniería de Software, Shari Lawrence Pfleeger, Pág. 549

### 5.3 Nuevas Tecnologías

Es importante destacar la función que tiene el avance de la tecnología informática, sobre todo en este siglo que a comenzado. Las nuevas tecnologías nos permitirán mantener un sistema de forma funcional para los usuarios durante más tiempo. Por ejemplo, podemos mencionar los avances en cuanto al almacenamiento de grandes cantidades volúmenes de datos y del desarrollo de interfaces. A continuación mencionaremos algunas de las nuevas tecnologías a las cuales e puede adaptar el sistema realizado en esté trabajo.

#### 5.3.1. Java

Java es un lenguaje de programación con el que podemos realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido que cada vez cobra más importancia tanto en el ámbito de Internet como en la información en general.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de las plataformas. Eso quiere decir que si hacemos un programa es Java podrá funcionar en cualquier computadora del mercado.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso por medio de computadoras distintas. Con Java podemos programar páginas Web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que deseemos hacer con acceso a través Web se puede hacer utilizando Java.

#### 5.3.2. ASP

ASP (Active Server Page) es una tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script.

Un lenguaje de servidor como esté es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página por Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente.

El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT, aunque también se puede utilizar en un PC con Windows 98 si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux podemos utilizar las ASP si instalamos un componente denominado Chilisoft.

Actualmente se ha presentado ya la segunda versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro.

### **5.3.3. Perl**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de archivos. Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows.

Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como Javascript o ASP. Esto quiere decir que el código de los scripts en Perl no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde Perl podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código Perl.

# *CAPITULO VI.*

## Capacitación

Cuando se ha elaborado un nuevo sistema, siempre es necesario capacitar y enseñar a los usuarios sobre el manejo de éste. El programa de capacitación se elabora desde las fases de diseño y codificación de un sistema. Esto permite ir estructurando el nivel de conocimiento que se debe aplicar al programa y depende de este el grado de complejidad de la capacitación.

La capacitación varía dependiendo del sistema y las diferencias que existen entre los usuarios que lo utilizarán; ya que las funciones que cada uno de estos desempeña son diferentes.

Para estructurar el programa de capacitación se considera cada nivel y las funciones de los usuarios, así como las tareas que realiza cada uno de ellos. En este punto es muy importante distinguir entre los usuarios finales y administrativos del sistema. Los usuarios finales requieren la capacitación para las actividades que ayuden a desempeñar sus labores.

En este capítulo tomaremos en consideración dos aspectos importantes, el entrenamiento y la documentación. Por lo que a medida que se diseñó el sistema se planificaron y se desarrollaron las ayudas que contribuirán a que el usuario aprenda a utilizar el sistema. El sistema irá acompañado por la documentación que los usuarios consultarán para resolver problemas u obtener mayor información.

## **6.1. Capacitación de los usuarios del sistema**

La capacitación de los usuarios la basamos primordialmente sobre las funciones principales, sobre todo aquellas a las que el usuario necesita acceder. Por ejemplo, un alumno que consulta los datos sobre su proyecto de Tesis.

La capacitación puede hacerse de diversas maneras, pero sin importar como se brinde, debe proporcionar en todo momento la información a los usuarios y no sólo cuando se haga la entrega del sistema por primera vez.

### **6.1.1. Sesiones teórico-prácticas**

Consideramos esta forma de entrenamiento como una de las más importantes, ya que las demostraciones agregan personalización a la capacitación y los usuarios responden positivamente. Además las sesiones involucran múltiples formas de reforzar lo que los usuarios están aprendiendo. Escuchar y observar como trabaja una función ayuda a recordar siempre las funciones y las técnicas.

Por lo anteriormente expuesto se decidió llevar a cabo diferentes sesiones con la Coordinación de Titulación de la DIMEI con el fin de explicar el funcionamiento y los alcances del sistema.

### **6.1.2. Documentación**

Los documentos contiene toda la información que se necesita para utilizar el sistema. Estos pueden ser manuales separados o disponibles en línea.

#### **Manual del Usuario**

El manual del usuario es una guía que muestra como utilizar el sistema. Esta guía esta integrada por información descriptiva del sistema, la cual explica su operación y las funciones que realiza. Dentro de la información proporcionada por la guía también se debe incluir información orientada hacia el administrador.

La información que contiene la guía proporciona una visión precisa del sistema. Dicha guía esta estructurada de tal forma que el usuario pueda ocuparla cada vez que tenga una duda

## 6.2. Ayuda en línea

La ayuda en línea hace que el entrenamiento resulte fácil, ya que el usuario puede explorar información adicional acerca del propósito o utilización de una función, sin ocupar tiempo en la búsqueda a través de los documentos en papel.

Esta guía es una interfaz gráfica para los usuarios, en el que el usuario puede consultar el tema en el cual existen dudas. Este es un tipo de manual mejor conocido como ayuda en línea, en el que se realizan consultas en cualquier momento durante la ejecución del programa y sobre las dudas que surgen sobre su operación.

La información que contiene una ayuda en línea, es información descriptiva y concreta del sistema, la cual explica la operación y funciones que realiza. Es importante que describa las funciones que ofrece el programa.

Esta ayuda debe estar construida de tal manera que los temas sean presentados con una estructura fácil de entender para el usuario. A continuación se describen algunas de las características que puede contener.

- Una interfaz gráfica que habilita el usuario para seleccionar un tema.
- Implementación de definiciones.

En este tipo de ayuda es importante no perder de vista que no se debe incluir demasiada información de carácter técnico. La elaboración de este tipo de ayuda durante el desarrollo del sistema ayuda a realizar las pruebas al mismo tiempo que se detectan errores que pueden existir en el desarrollo del sistema.

# *Conclusiones*

Al llegar al final del presente trabajo podemos decir que el análisis, desarrollo, pruebas y mantenimiento de un sistema, requiere un enfoque sistemático que permita asegurar su aceptación por parte de los usuarios. Donde las necesidades y limitaciones del usuario deben estar determinadas y claramente establecidas para lograr un análisis confiable del problema a resolver; en el diseño del producto es necesario tener presente a los usuarios ya que ellos son quienes trabajarán directamente con el sistema, las pruebas que se realicen a lo largo del desarrollo garantizan el buen funcionamiento y calidad del sistema. Por último las tareas de mantenimiento incluirán solicitudes de análisis de cambio, rediseño y modificación.

Para lograr el desarrollo y mantenimiento de un producto de software, es necesario utilizar técnicas de solución de problemas junto con las notaciones adecuadas. Para el caso de nuestro proyecto, se realizó como primer paso, una descripción de la situación por resolver considerando las restricciones del problema para justificar una estrategia de solución sistematizada que realice las funciones requeridas.

Cuando ya se tienen los requerimientos y las funciones que realizará el sistema, el siguiente paso consiste en especificar la manera en que esto será estructurado; en este punto es necesario emplear herramientas que nos permitan trabajar con los requisitos, independientemente del lenguaje de programación que se empleará en una etapa posterior. Tales herramientas incluyen a los diagramas de flujo de datos que permiten especificar las actividades de los procesos y la transferencia de datos entre ellos de una manera gráfica. También es empleado el diccionario de datos, que contiene un listado organizado de todos los elementos de datos involucrados en el sistema permitiendo una comprensión de las entradas, salidas, almacenamientos y pasos intermedios desarrollados por los procesos.

El desarrollo de un sistema de programación incluye la determinación del tipo de sistema a desarrollar, en nuestro caso, se trata de un sistema de procesamiento de datos, que se apoya en una base de datos para el almacenamiento de información, dentro de este ámbito existen terminologías y notaciones particulares que permiten manejar las relaciones entre los datos y entidades que requiere el sistema para su funcionamiento, tales relaciones se expresan gráficamente mediante los diagramas Entidad-Relación.

Después del análisis y diseño, se puede empezar a trabajar con un lenguaje de programación para traducir lo desarrollado en instrucciones para la computadora; las características disponibles en el lenguaje ejercen influencias sobre la estructura del sistema.

Las actividades que sucederán después de la entrega del producto caen dentro de lo que es el mantenimiento, tales actividades incluirán, cuando así sea requerido, mejoras en el producto, adaptaciones a nuevos sistemas operativos y/o nuevas máquinas.

Para la **“Coordinación de titulación de la DIMEI”** el presente trabajo, significa contar con un sistema que integra toda la información necesaria de tesis, directores y sus respectivos proyectos, arrojándonos información relevante de cada uno de ellos, además de contar con un sistema seguro y eficiente para el control y seguimiento éstos.

Finalmente, podemos decir que el objetivo fundamental de este trabajo fue proporcionar un sistema que facilitará a los usuarios llevar a cabo algunas de sus tareas, atendiendo a la calidad del producto final, todo esto empleando los métodos, herramientas, técnicas y conocimientos aprendidos a lo largo de la carrera, la cual nos proporcionó lo necesario para poder resolver un problema.

## Ciclo de vida del Software

Por ciclo de vida, se entiende la sucesión de etapas por las que pasa el software desde que un nuevo proyecto es concebido hasta que se deja de usarse. Cada una de estas etapas lleva asociada una serie de tareas que deben realizarse, y una serie de documentos que serán la salida de cada una de estas fases y servirán de entrada en la fase siguiente. El ciclo de vida comprende cuatro fases:

**Concepción:** Define los alcances del proyecto.

**Elaboración:** Define un plan del proyecto, especifica las características y fundamenta la arquitectura.

**Construcción:** Crea el producto.

**Transición:** Transfiere el producto a los usuarios.

Existen diversos modelos de ciclo de vida, es decir, diversas formas de ver el proceso de desarrollo de software, y cada uno de ellos va asociado a un paradigma de la Ingeniería del software, dicho de otra manera, a una serie de métodos, herramientas y procedimientos que debemos usar a lo largo de un proyecto. La elección de un método u otro se realiza de acuerdo con la naturaleza del proyecto y de la aplicación.

Las principales diferencias entre distintos modelos de ciclo de vida están en el alcance del ciclo dependiendo de hasta donde llegue el proyecto correspondiente; ya que un proyecto puede comprender un simple estudio de viabilidad o su desarrollo completo, y por otro lado las características de las fases de cada modelo, es decir la forma en que están estructuradas, ya que pueden ser lineales, con prototipo o en espiral. Los diferentes modelos de ciclo de vida también son conocidos como metodologías.

## Metodología

Una metodología es un conjunto de pasos que describen como construir técnicamente un software, siendo una versión ampliada y detallada de un ciclo de vida completo y que incluye lo siguiente:

- Reglas, procedimientos, métodos y herramientas.
- Funciones individuales y en grupo por cada tarea.
- Productos resultantes.
- Normas de calidad.

Definido lo que es una metodología, describimos a continuación aquellas que son más utilizadas para el desarrollo de Software.

### **Ciclo de vida clásico**

Es la más utilizada, siempre que es posible, por ser la más sencilla. Consiste en descomponer la actividad general del proyecto en fases que se aplican en forma lineal, es decir cada una se realiza solo una vez. Como es un ciclo lineal es fácil dividir las tareas de manera que no se necesite el resultado de la tarea siguiente. Véase Figura 1.

En esta cada fase requiere información de entrada, procesos y resultados que serán utilizados en la siguiente fase. El ciclo de vida clásico está compuesto por las siguientes fases:

- Planificación del sistema
- Análisis
- Diseño
- Codificación
- Pruebas
- Implantación
- Mantenimiento

### **Planeación del Sistema**

Es la etapa en la que se considera si el sistema solicitado es factible de llevarse a cabo, considerando para ello los principales propósitos, los objetivos, problemas, necesidades y restricciones. Para saber si el sistema es factible de desarrollarse es necesario realizar un estudio de factibilidad en el que se consideran los siguientes tres aspectos:

**Factibilidad Técnica:** Estudia si el proyecto puede desarrollarse con el software existente o es necesario utilizar una nueva tecnología.

**Factibilidad Económica:** Investiga si los costos se justifican con los beneficios que se obtienen.

**Factibilidad Operacional:** Investiga si el sistema será utilizado, como para obtener beneficio.

## **Análisis**

En esta fase se hace la recopilación de los requisitos necesarios para el desarrollo del sistema, los cuales son indispensable para comprender mejor el problema a resolver y de esta manera también determinar los requerimientos de hardware.

## **Diseño**

El diseño de un sistema de información responde a la forma en la que el sistema cumplirá con los requerimientos identificados durante la fase de análisis. Por lo que en esta fase se traducen los requisitos en una representación del software que pueda ser establecida de forma que obtenga la calidad requerida antes de que comience la codificación. Al igual que los requisitos, el diseño se documenta y forma parte de la configuración del software.

El diseño es una tarea que puede ser más sencilla si se elabora un esquema del formato o algunas de las pantallas que se esperan aparezcan cuando el sistema este terminado, dicho esquema puede ser elaborado en papel o en la pantalla de alguna terminal.

En esta fase también son indicados cuales serán los datos de entrada, cuales de los datos serán calculados y cuales deberán ser almacenados, los procedimientos que indican como procesar los datos y como producir las salidas, así como la elaboración los documentos que contienen las especificaciones de diseño en donde se representa a los procedimientos mediante diagramas, tablas y símbolos especiales.

En resumen podemos decir que el diseño del sistema es un proceso que se enfoca sobre cuatro atributos distintos del programa:

- Estructura de los datos
- Arquitectura del software
- El detalle procedimental
- La caracterización de la interfaz

## **Codificación**

Es la fase en la que se traduce el diseño en una forma legible para la máquina. De aquí la importancia de que el diseño se realice de una manera detallada, para que la codificación puede llevarse a cabo de manera casi mecánicamente.

## Pruebas

Durante esta fase el sistema puede ser empleado de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo a las especificaciones, y en la forma en que los usuarios lo necesitan. La prueba se centra en la lógica interna del software, asegurándose de que todas las sentencias sean probadas, alimentando sus entradas con conjuntos de datos de prueba para su procedimiento y después analizando los resultados.

## Mantenimiento

El mantenimiento de software aplica cada uno de los pasos procedentes del ciclo de vida al programa existente en vez de a uno nuevo.

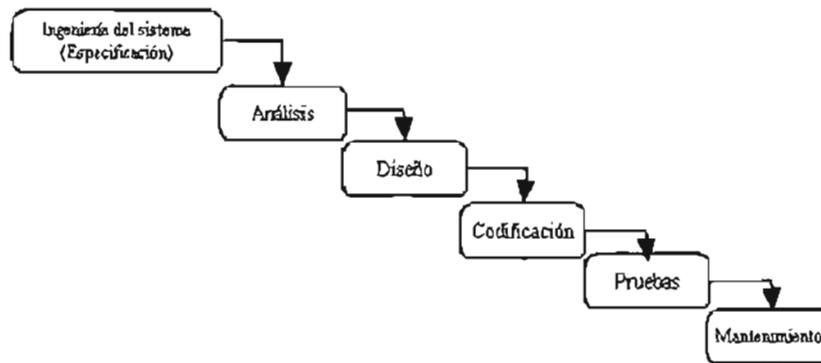


Figura 1. Diagrama del Ciclo de vida Clásico

## Ciclo de vida clásico modificado

Fue creado para permitir la retroalimentación entre fases, es un modelo interactivo y no lineal, que facilita la terminación de metas y tareas. En este modelo es normal congelar partes del desarrollo después de cierto número de iteraciones. Véase Figura 2.

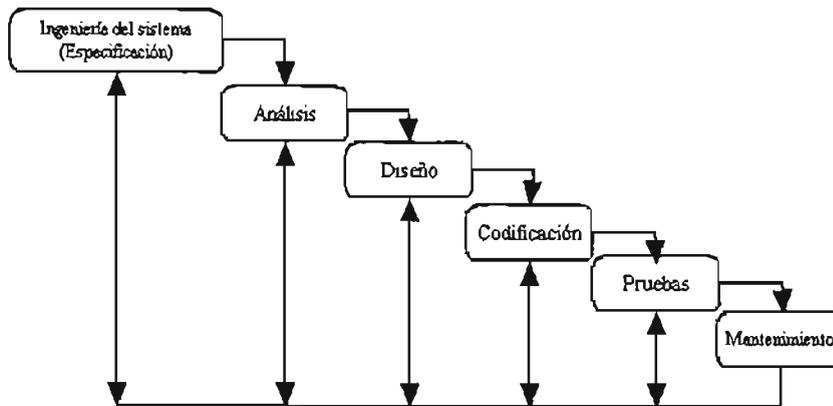


Figura 2. Diagrama del Ciclo de vida Clásico modificado

### Construcción de prototipos

Como en el ciclo de vida clásico se comienza con la recolección de los requisitos, a continuación se produce un diseño rápido, el cual se enfoca sobre la representación de los aspectos del software visibles al usuario. El diseño rápido conduce a la construcción del prototipo que será evaluado por el usuario, y que será utilizado para refinar los requisitos del software a desarrollar. Se produce un proceso interactivo en el que el prototipo se va afinando hasta satisfacer las necesidades del cliente, al mismo tiempo que va ayudando al desarrollo. Véase Figura 3.

Un prototipo no es más que la representación o modelo del producto de programación que incorpora componentes del producto real, pero que tiene un funcionamiento limitado en cuanto a sus capacidades y confiabilidad. Sin embargo hay varias razones para desarrollarlo; una de ellas es ilustrar el formato de los datos de entrada, mensajes y otro tipo de información importante para el cliente.

Es importante que nunca se pierda de vista que el desarrollo de un prototipo solo sirve como un mecanismo de definición de los requerimientos del sistema. La construcción de prototipos es un modelo que facilita la creación del sistema a construir mediante una de las tres formas siguientes:

1. Un prototipo en papel o un modelo basado en computadora que describa la interacción hombre-máquina.
2. Un prototipo que implemente algunos subconjuntos de la función requerida del programa deseado.

3. Un programa existente que ejecute parte o toda la función deseada, pero que tenga otras características que deben ser mejoradas en el nuevo trabajo de desarrollo.

La construcción final del producto puede llevarse a cabo de dos maneras:

**Prototipos evolutivos:** Toma el prototipo y lo refina.

**Prototipos Revolucionarios:** Desecha el prototipo, la especificación se complementa y se utiliza con el método de cascada o ciclo de vida clásico.

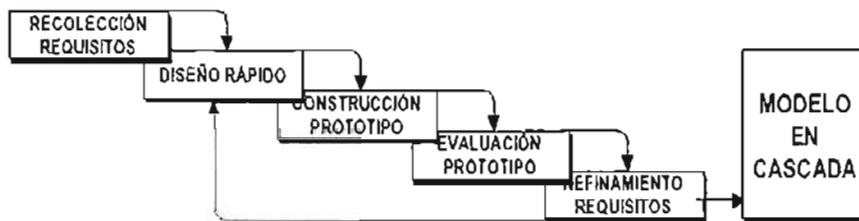


Figura 3. Diagrama del modelo basado en la construcción de prototipos

### Modelo Espiral

Es un modelo en el que la idea básica es el desarrollo evolutivo, usando el modelo clásico para cada etapa. Es un modelo que está orientado a evitar riesgos de trabajo; puesto que no define en detalle el sistema completo a la primera. Los desarrolladores deberán solamente definir las más altas prioridades, implementarlas y entonces obtener una aprobación de los usuarios. Con este conocimiento, deberán entonces retroceder o volver al punto de partida para definir e implementar más y mejores partes. Véase Figura 4.

El modelo espiral mejora el modelo clásico enfatizando la naturaleza iterativa del proceso de diseño, introduciendo un ciclo de prototipo iterativo, en el que se construye una versión inicial del producto, sin implementar completamente la funcionalidad del sistema y su totalidad. Este primer producto ayuda a los analistas a establecer las necesidades del cliente facilitando la captación de los requerimientos reduciendo el riesgo de parcheo del producto final, además de permitir al usuario ver

funcionando una versión de lo que será el programa sin asumir que dicha versión no es robusta ni completa.

Para el desarrollo del producto inicial puede utilizarse el modelo de construcción de prototipos, dichos prototipos pueden construir de acuerdo a las siguientes clases:

**Vertical:** Desarrolla completamente algunas facetas del producto.

**Horizontal:** Desarrolla parcialmente todas las facetas del producto.

**Evolutivo:** La versión final es el producto ya construido.

**Desechable:** Se usa solo para la captación de requerimientos y funcionalidad.

Como se puede observar es un método iterativo, que como ya fue mencionado, toma las mejores características del método clásico y el de prototipos, agregando un nuevo elemento conocido como “análisis de riesgo”. Si el análisis de riesgos indica una incertidumbre en los requisitos, se pueden crear prototipos en el cuadrante de Ingeniería para dar asistencia tanto al desarrollador como al cliente.

El cliente evalúa el trabajo de Ingeniería y si así lo concederá sugiere modificaciones. En base a los comentarios del cliente, se produce la siguiente fase de planeación y de análisis de riesgo. Cada iteración se vuelve una decisión para tomar o no. Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.

Sin embargo en la mayoría de los casos, el proyecto sigue y, cada vez que la espiral se encuentra más hacia fuera, se convierte en un modelo más completo y al final, un sistema operacional. Cada vuelta alrededor de la espiral requiere Ingeniería, que se puede llevar a cabo mediante el método clásico de o el de prototipo. Mientras más se aleja el desarrollo del centro, aumentan las actividades de Ingeniería, por eso es considerado como un método de enfoque evolutivo o incremental, el cual se explica a continuación:

### **Desarrollo Incremental**

El desarrollo incremental tiene por objetivo identificar elementos de funcionalidad separados en los requerimientos iniciales, dar prioridad y crear un plan de desarrollo utilizando diferentes modelos del ciclo de vida para los diferentes incrementos, de tal manera que en cada giro se construye un nuevo modelo del sistema completo. Este modelo puede combinarse con otros modelos del proceso de desarrollo. El desarrollo incremental es el mejor modelo para el desarrollo de grandes sistemas; ya que por su complejidad no se recomienda para el desarrollo de sistemas pequeños.

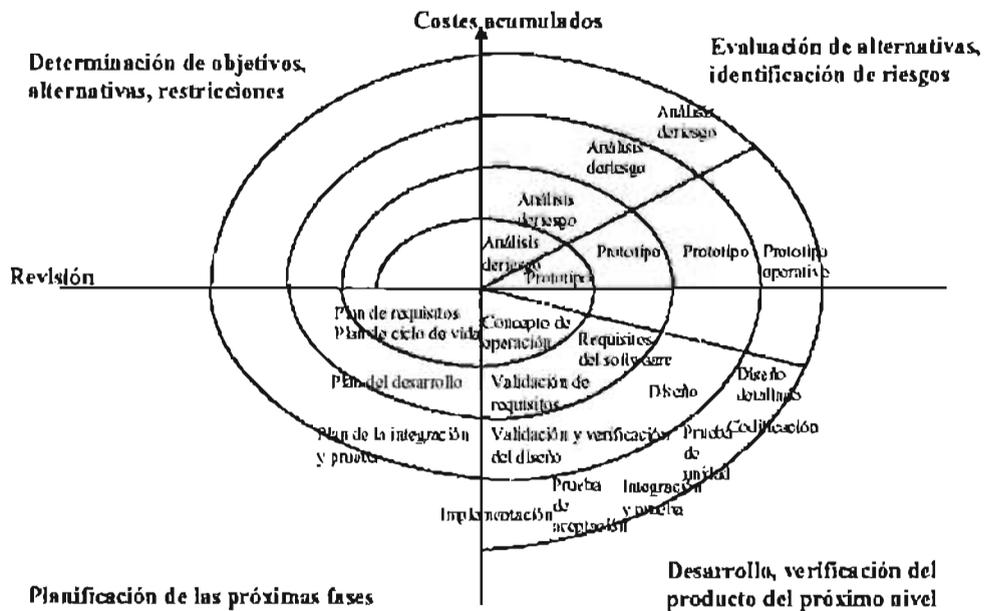


Figura 4. Diagrama del modelo Espiral

Después de realizada esta investigación acerca de las metodologías existentes para el desarrollo de sistemas se decidió realizar el desarrollo del presente sistema con el modelo de vida clásico.

## Código de procesos

## Página principal

```

<HTML>
<HEAD>
<meta http-equiv=Content-Type content="text/html; charset=ISO-8859-1">
<TITLE>Index</TITLE>
</HEAD>
<BODY bgcolor="#000033">
<!-- URL's used in the movie-->
<A HREF=Jefe_1.html></A><A HREF=Admin_1.html></A><A HREF=Frames_Propuesta.html></A>
<A HREF=Alumnos_1.html </A> <A HREF=Profesor_1.html></A> <!-- text used in the movie-->
<!--División de Ingeniería Mecánica e Industrial-->
<OBJECT classid = "clsid: D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
WIDTH="1000" HEIGHT="600" Id="Index" ALIGN="">
<PARAM NAME=movie VALUE="Index.swf"> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033> <EMBED src="Index.swf" quality=high bgcolor=#000033 WIDTH="1000"
HEIGHT="570" NAME="Index" ALIGN=""
TYPE="application/x-shockwave-flash" PLUGINS PAGE="http://www.macromedia.com/go/getflashplayer"></EMBED>
</OBJECT>
</BODY>
</HTML>

```

## Código para entrar a la sesión de un alumno

```

<html><head><title>FORMULARIO ALUMNO</title></head>
<body BgColor="#000033" Text="#FFFFFF" alink="#999900" blink="#ccc09">
<? php

include ("curp.inc");
include ("conec.php");

function for_Alumno ($texto, $Cuenta, $RFC)
{
echo" <br>
<FONT Size='5'><center>Bienvenido por favor proporcione los siguiente datos:</FONT>
<br><br><CENTER> $texto </CENTER><br>
<table><tr><td>
<FORM Action='forAlum1.php' Method='post'> ";
printf ("<font Size='5' Color=#CC9900>Número de Cuenta :</font><input Type='texto' Name='Cuenta' Size='10' Maxlength='110'
value=%s">
<br>
<font Size='5' Color=#CC9900>RFC :</font><input Type='texto' Name='RFC' Size='10' Maxlength='10'
value=%s'", $Cuenta, $RFC);
echo"<br><br><center>
<INPUT type = submit value = 'Enviar' name='enviar'>
<INPUT type=submit value='Borrar' name='borrar'>
</FORM>
</td></tr></table> ";
echo"
<table Width=100% ><tr><td Align=Right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=550 HEIGHT=180 Id=ayuda_A ALIGN=>
<PARAM NAME=movie VALUE=ayuda_A.swf> <PARAM NAME=quality VALUE=high> <PARAM NAME=bgcolor
VALUE=#000033> <EMBED src=ayuda_A.swf quality=high bgcolor=#000033 WIDTH=550 HEIGHT=180
NAME=ayuda_A ALIGN=
TYPE=application/x-shockwave-flash PLUGINS PAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</td></tr></table>";
}
if ( !$enviar) // verificamos si no se ha pulsado el botón 'Enviar datos'
{
for_Alumno ("", "", ""); // formulario sin valores iniciales
}

```

```

else
{
if ( !empty($RFC) && !empty($Cuenta) ) // verificamos si los campos tienen datos
{ //echo "<CENTER>Valores en los Campos </CENTER>";
// Los campos son validados, nos conectamos a la base de datos para comprobar los valores.
if( numero_cuenta($Cuenta) != "Verifica Numero de Cuenta" && valida_RFC($RFC) != "Verifique el RFC" )
{
$conexion = Conectarse (); // funcion para conectarse a la base de datos
$sql = "select * from alumno where num_Cuenta='$Cuenta' and rfc_Alumno='$RFC'";
$resultado = mysql_query ($sql, $conexion);
$filas = mysql_num_rows ($resultado); // el número de filas en la consulta
if ($filas == 1) // Se realizo la conexion a la base de datos el usuario existe
{
while ($row = mysql_fetch_array ($resultado)) // numero de elementos que tiene la tabla
{
echo "Los valores de la tabla son <br><br>
El Numero de Cuenta<FONT BGCOLOR='\BLUE'\>$row [num_Cuenta] </FONT><br>
El RFC $row [rfc_Alumno]";

// Validamos campos del formulario
echo "<br><center>Bien venido ala aplicación ";
// header ("Location: opcines_alumno.html");
echo "<br><a href='opcines_alumno.php? Cuenta=$Cuenta' > Entrar </a><br>";
}
}
}
else
{// No se encontraron los valores en la base
for_Alumno ("Datos validos, pero su N&uacute;mero de Cuenta o RFC no coinciden con la
Base de Datos ", $Cuenta, $RFC);
}
}
else // Campos no validados se muestra el formulario nuevamente
{ //echo "<CENTER><font Color='#CC9900' >Campos no Validos </font> </CENTER>";

{ //echo "<CENTER><font Color='#CC9900' >Campos no Validos </font> </CENTER>";
if (valida_RFC ($RFC) == "Verifica el RFC")
{
printf ("<CENTER> %s </CENTER> ", valida_RFC ($RFC));
$RFC="";
}

else if (numero_cuenta ($Cuenta) == "Valida Numero Cuenta ")
{
printf ("<CENTER> %s </CENTER>", numero_cuenta ($Cuenta));
$Cuenta = "";
}

for_Alumno ("Verifica los datos RFC o N&uacute;mero de cuenta ", $Cuenta, $RFC);
}
}
else
{
for_Alumno ("Valores incompletos en los campos", $Cuenta, $RFC);
}
}

?>
</BODY>
</HTML>

```

**Código para entrar a la sesión de profesor**

```

<HTML><HEAD><TITLE>FORMULARIO PROFESORES</TITLE></HEAD>
<BODY BgColor='#000033' Text='#FFFFFF'>
<? php

```

```

include ("curp.inc");
include ("conec.php");
function for_Prof ($Prof,$texto )
{ echo "
<br>
<FONT Size='5'><center>Bienvenido por favor proporcione los siguiente datos:</FONT>
<br><br><CENTER> $texto </CENTER>
<br> <table><tr><td>
<FORM Action='forProf1.php' Method='post'>;

printf (*<FONT Size='5' Color='#CC9900'>RFC : </FONT> <INPUT Type='texto' Name='Prof [RFC]' Size='10'
Maxlength='10' value='%s'>
<br><br>
<FONT Size='5' Color='#CC9900'>Password :< /FONT><INPUT Type='password' Name='Prof[Password]'
Size='15' Maxlength='15' value='%s'>
<br><br> ", $Prof [RFC], $Prof[Password] );

echo"<center><INPUT type = submit value = 'Enviar' name='enviar'>
<INPUT type = submit value = 'Borrar' name='borrar' ></center>

</FORM></td></tr></table>";
echo"
<table Width=100%>
<tr><td Align=right>
<OBJECT classId=clsid: D27CDB6E-AE6D-11cf-9688-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=550 HEIGHT=120 id=ayuda_D >
<PARAM NAME=movie VALUE=ayuda_D.swf><PARAM NAME=quality VALUE=high><PARAM
NAME=bgcolor VALUE=#000033><EMBED src=ayuda_D.swf quality=hgh bgcolor=#000033 WIDTH=550
HEIGHT=120 NAME=ayuda_D
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</td></tr>
</table> ";
}

if ($borrar)
{
for_Prof ("","Limpio campos<br>");
}

else if (! $enviar) // verificamos si no se ha pulsado el botón 'Enviar datos'
{
for_Prof ("","<br><br>");
}
else
{
if (! empty ($Prof [RFC]) && ! empty ($Prof[Password]) ) // verificamos si los campos tienen datos
{
/* echo "<CENTER>Valores en los Campos </CENTER>"; Validamos campos del formulario */

if (valida_RFC ($Prof [RFC])) = "Verifique el RFC"
{
$conexion = Conectarse (); // función para conectarse a la base de datos
$sql = "select * from profesor where rfc_Prof='$Prof [RFC]' ";
$resultado = mysql_query ($sql, $conexion);
$filas = mysql_num_rows ($resultado); // el número de filas en la consulta
if ($filas == 1) // Se realizó la conexión a la base de datos el usuario existe
{
while ($row = mysql_fetch_array ($resultado)) // numero de elementos que tiene la tabla
{
//echo "Los valores de la tabla son <br><br>El login<FONT BGCOLOR='BLUE'>$row [login] </FONT><br>
El password $row [password]";
if ($Prof [Password] == $row [password]) // Verifico que el password coincide
{
echo "Entramos a la aplicación";

// Validamos campos del formulario
echo "<br><center>Bien venido a la aplicación ";

```

```

// header ("Location: opcines_alumno.html");
echo "<br><a href='Opcines_Profesor.php? RFC=$Prof [RFC]' > Entrar </a></br>";
}

else
{$texto="Su PASSWORD no es correcto ";
$Prof [Password] ="";
for_Prof ($Prof, $texto);
}
}
}
else
{$texto= "Su RFC <FONT COLOR='RED'\> $Prof [RFC] </FONT> no existe en la Base da Datos ";
$Prof [RFC] ="";
$Prof [Password] ="";
for_Prof ($Prof, $texto);
}
}
else
{$texto ="<CENTER>Su RFC <FONT COLOR='RED'\> $Prof [RFC] </FONT> no es valido </CENTER><br>";
$Prof [RFC] ="";
$Prof [Password] ="";
for_Prof ($Prof, $texto);
}
}
else
{
for_Prof ($Prof,"<CENTER> Faltan Valores en uno de los campos </CENTER><br>");
}
}

?>
</BODY>
</HTML>

```

#### Código para entrar a la sesión de Jefe de Departamento

```

<HTML><HEAD><TITLE>FORMULARIO DE JEFES</TITLE></HEAD>
<BODY BgColor='#000033' Text='#FFFFFF'>
<?php
include ("curp.inc");
include ("conec.php");

/* Este es el formulario de la pagina */

function select ($tabla)
{
$conexion = Conectarse ();
$base = "OSIRIS";
$conconsulta = "SELECT * FROM $tabla";
$resultado = mysql_query ($conconsulta);
$lineas = mysql_num_rows ($resultado);
$columnas = mysql_num_fields ($resultado);
return array ($resultado, $lineas, $columnas);
}

function elegir_select ($opcion, $Enviar, $tabla)
{
/* Se obtienen valores de columna, línea y resultado de la función select por medio de
llist (numero de valores regresados) */

list ($resultado, $lineas, $columnas) = select ($tabla);
Echo "<TD><SELECT NAME='dato {$opcion}' ALIGN='LEFT'>
<OPTION VALUE='0'>Seleccionar opción"; // valor pordefault en el select

/* Aquí es donde se obtienen todos los datos de la tabla seleccionada y son los valores
desplegados en el select, la llave de cada tabla es almacenada en VALUE y el segundo campo

```

```

que contiene la descripción es el valor */

for ($i=0; $i < $lineas; $i++)
{
  $id=mysql_result ($resultado, $i, 0); // clave del registro
  $valor = mysql_result ($resultado, $i, 1); //valor del registro (nombre)
  echo "<OPTION VALUE=$id>$valor ";
}
}
function Jefe($Jefe,$texto)
{
  echo "
  <br>
  <FONT Size='5'><center>Bienvenido por favor proporcione los siguiente datos:</FONT>
  <CENTER>$texto</CENTER>
  <table><tr><td>
  <FORM Action='forJefe1.php' Method='post'>";

  printf ("<FONT Size='5' Color=#CC9900>RFC: </FONT> <INPUT Type='texto' Name='Jefe [RFC]'
  Size='10' Maxlength='10' value='%s'>
  <br><br>
  <FONT Size='5' Color=#CC9900>Login: </FONT><INPUT Type='texto' Name='Jefe [Login]'
  Size='15' Maxlength='15' value='%s'>
  <br><br>
  <FONT Size='5' Color=#CC9900>Password :< /FONT><INPUT Type='password' Name='Jefe [Password]'
  Size='15' Maxlength='15' value='%s'>
  <br><br></td></tr>
  ", $Jefe [RFC], $Jefe [Login], $Jefe [Password]);
  echo"</td></tr></table>
  <table><tr><td>";
  echo " <FONT Size='5' Color=#CC9900>Departamento</FONT>:";
  echo " <input type='text' value='%s'>";
  echo " <input type='button' value='Enviar'>";
  echo " <input type='button' value='Borrar'>";
  echo "</td></tr></table>";

  echo "
  <table Width=100%><tr><td align=right>
  <OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
  codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
  WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
  <PARAM NAME=movie VALUE=ayuda_H.swf><PARAM NAME=quality VALUE=high>
  <PARAM NAME=bgcolor VALUE=#000033>
  <EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
  TYPE=application/x-shockwave-flash PLUGINSOURCE=http://www.macromedia.com/go/getflashplayer></EMBED>
  </OBJECT>
  </td></tr></table>
  ";
}

if ($borrar)
{
  Jefe ("", "Limpio campos");
}

else if (! $enviar) // verificamos si no se ha pulsado el boton "Enviar datos"
{
  Jefe ("", "<br>");
}
else
{
  if(!empty($Jefe[RFC]) && !empty($Jefe[Login]) && !empty($Jefe[Password]) && !empty($dato[id_Depto]))
  //verificamos si los campos tienen datos
  {
    if ( valida_RFC ($Jefe [RFC])!= "Verifique el RFC" )
    // Si los campos son validos es prudente hacer la conexion con la base de datos y el valor de la curp

    $conexion = Conectarse(); // funcion para conectarse a la base de datos
    $sql = "select * from jefe where rfc_Prof='$Jefe[RFC]' ";
    $resultado = mysql_query($sql, $conexion);
    $filas = mysql_num_rows($resultado); // el número de filas en la consulta
  }
}

```

```

if ($filas == 1) // Se realizo la conexion a la base de datos el usuario existe
{
    while ($row = mysql_fetch_array ($resultado)) // número de elementos que tiene la tabla
    {
        //echo "Los valores de la tabla son <br><br>El login<FONT BGCOLOR=\`BLUE\`>$row [login]
        </FONT><br>El password $row [password]";
        if( ($Jefe[Login] == $row[login] ) && ($Jefe[Password] == $row[password]) )
        // Verifico que el login y pasword coincidan
        {
            echo "Entramos ala aplicación Depto $dato [Id_Depto]";
        }
        else
        { $texto="Su LOGIN <FONT COLOR=\`RED\`> $Jefe [Login] </FONT> O PASSWORD no coinciden";
          $Jefe [Password] ="";
          Jefe ($Jefe, $texto);
        }
    }
}
else
{$texto= "Su RFC <FONT COLOR=\`RED\`> $Jefe [RFC] </FONT> no existe en la Base da Datos ";
 $Jefe [RFC] ="";
 $Jefe [Password] ="";
 Jefe ($Jefe, $texto);
}
}
else
{ $texto= " Verifique su RFC <FONT COLOR=\`RED\`> $Jefe [RFC] </FONT> ";
 $Jefe [RFC] ="";
 $Jefe [Password] ="";
 Jefe ($Jefe,$texto);
}
}
else
{
    Jefe ($Jefe,"<CENTER> Valores incompletos en los campos </CENTER> ");
}
}
?>
</BODY>
</HTML>

```

#### Código para entrar a la sesión de Administrador

```

<HTML><HEAD><TITLE>FORMULARIO ADMINISTRADOR</TITLE></HEAD>
<BODY BgColor=#000033' Text=#FFFFFF' VLING="#CC9900" LING="#CC9900">
<?php
include ("conec.php"); // archivo utilizado para conectarnos a la base de datos
//include("funciones.php");
include("curp.inc");
//include("fun_for.inc");

//funciones
function for_adm ($RFC, $Login, $Password,$texto)
{ echo"<br>
<FONT Size='5'><center>Bienvenido por favor proporcione los siguiente datos:</FONT>
<br><br><CENTER> $texto </CENTER><br>
<table><tr><td>
<FORM Action='forAdm1.php' Method='post'> ";
printf (" <FONT Size='5' Color=#CC9900>RFC : </FONT><INPUT Type='texto' Name='RFC' Size='11'
Maxlength='10' value='%s'>
<br><br>
<FONT Size='5' Color=#CC9900>Login :</FONT><INPUT Type='texto' Name='Login' Size='11
Maxlength='10' value='%s'>
<br><br>
<FONT Size='5' Color=#CC9900>Password :</FONT><INPUT Type='password' Name='Password' Size='11'

```

```

Maxlength='10' value='%s'>
", $RFC, $Login,$Password);
echo"<br><br><center>
<INPUT type = submit value = 'Enviar' name='enviar'>
<INPUT type = submit value = 'Borrar' name='borrar'></center>
</FORM></td></tr></table>
";
}

function Intentos ($numero)
{
echo "<CENTER><FONT COLOR='WHITE'><H3> INTENTO FALLIDO</H3></FONT>";
$intentos = $numero +1;
echo "<HR ALIGN='CENTER' WIDTH='50 %'><BR>
<FONT COLOR='WHITE'>Ha hecho $intentos Intentos </FONT><BR><BR>
REGRESAR AL MENÚ INICIAL<BR><BR> <A HREF='forAdm1.php'>REGRESAR</A></FONT><BR><BR>
</CENTER><HR ALIGN='CENTER' WIDTH=50%><BR>";
}

if($borrar)
{
//echo "Borramos valores en campos";
for_adm ("","","","Limpio campos");
echo"
<center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033>
<EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}

else if ( !$enviar) // verificamos si no se ha pulsado el botón 'Enviar datos'
{ ECHO "<BR>";
for_adm ($RFC, $Login, $Password, " ");
echo" <center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf><PARAM NAME=quality VALUE=high><PARAM
NAME=bgcolor VALUE=#000033><EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650
HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}

else
{
$numero = contador();
// echo " valor "+ $numero;
if ( $numero >= 4)
(Intentos ($numero) );
else {
if( !empty($RFC) && !empty($Login) && !empty($Password)) // verificamos si los campos tienen datos
{if ( valida_RFC ($RFC))= "Verifique el RFC" ) // Validamos los campos del formulario
$RFC = valida_RFC ($RFC);

//echo "la curp: $RFC El nombre es $nombre ";
if ( $RFC!= "Verifique el RFC")
{ // Si los campos son validos es prudente hacer la conexión con la base de datos y el valor de la curp
$conexion = Conectarse (); // función para conectarse a la base de datos
$sql = "select * from administrador where rfc='$RFC'";
}
}
}
}
}

```

```

$resultado = mysql_query ($sql, $conexion);
$filas = mysql_num_rows ($resultado); // el número de filas en la consulta
if ($filas == 1) // Se realizó la conexión a la base de datos el usuario existe
{
    while ($row = mysql_fetch_array ($resultado)) // numero de elementos que tiene la tabla
    { // echo "Los valores de la tabla son <br><br>El login <FONT BGCOLOR=#"BLUE"#$row [login]
      </FONT> <br>El password $row [password]";
      if (($Login == $row [login]) && ($Password == $row [password])) // Verifico que el login y password coincidan
      {
          include ("Cambios2.php");
      }
    }
}

else { // El password o Login es incorrecto
for_admin ($RFC,"", "", "El password o Login es incorrecto");
echo "
<center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033>
<EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSOURCE=http://www.macromedia.com/go/getflashplayer> </EMBED>
</OBJECT>
</center>
";
}
}
mysql_free_result ($resultado);
}
else
{ // El usuario no existe
For_admin ("RFC", $Login, "", "Verifique su RFC <font color=#009955> RFC </font>");
echo "
<center><table Width=100%><tr><td align=right>
<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033>
<EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSOURCE=http://www.macromedia.com/go/getflashplayer> </EMBED>
</OBJECT>
</center> ";
}
}
else { // RFC INVÁLIDA
for_admin ($RFC, Login, $Password, "El usuario no existe verifique el RFC");
echo "<center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high> <PARAM
NAME=bgcolor VALUE=#000033> <EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650
HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSOURCE=http://www.macromedia.com/go/getflashplayer>
</EMBED>
</OBJECT>
</center> ";
}
}
else
{ // formulario con datos incompletos
for_admin ($RFC, $Login, $Password, "Valores incompletos en los campos");
}
}
}

```





```

    echo "Edo. $estado ";
}
echo "
</font></td>
</tr>
<tr>
<td Bgcolor=#CC9900><font size=4 color=#000033><b>Tel&ecute;fono</font></b></td>
<td><font size=4>$fila[telefono]</td>
</tr>
<tr><td Bgcolor=#CC9900><font size=4 color=#000033><b>Email</b></font></td>
<td><font size=4>$fila [email]</td>
</tr>
</table></center></center>";
Cerrar($conexion);
?>
<br>
<A Href=Index.html target=_top><IMG Src=home_0.gif align=right ></A>
<?php ECHO " <A Href='Opcines_Profesor.php?RFC=$RFC' > <IMG Src=flecha_0.gif align=right ></A> ";?>

</body>
</html>

```

**Código para las consultas del Administrador**

```

<HTML><HEAD><TITLE>FORMULARIO ADMINISTRADOR</TITLE></HEAD>
<BODY BgColor=#000033' Text=#FFFFFF' VLING=#CC9900' LING=#CC9900">
<?php
include ("conec.php"); // archivo utilizado para conectamos a la base de datos
//include ("funciones.php");
include ("curp.inc");
//include ("fun_for.inc");

//funciones
function for_adm($RFC,$Login,$Password,$texto)
{echo"<br>
<FONT Size='5'><center>Bienvenido por favor proporcione los siguiente datos:</FONT>
<br><br><CENTER> $texto </CENTER><br>
<table><tr><td>
<FORM Action='forAdm1.php' Method='post' > ";
printf (" <FONT Size='5' Color=#CC9900'>RFC:
</FONT><INPUT Type='texto' Name='RFC' Size='11' Maxlength='10' value='%s'>
<br><br>
<FONT Size='5' Color=#CC9900'>Login:
</FONT><INPUT Type='texto' Name='Login' Size='11 Maxlength='10' value='%s'>
<br><br>
<FONT Size='5' Color=#CC9900'>Password:
</FONT><INPUT Type='password' Name='Password' Size='11' Maxlength='10' value='%s'>
", $RFC, $Login, $Password);
echo"<br><br><center>
<INPUT type = submit value = 'Enviar' name='enviar'>
<INPUT type = submit value = 'Borrar' name='borrar'></center>
</FORM></td></tr></table>
";
}

funcion Intentos ($numero)
{
    echo "<CENTER><FONT COLOR='WHITE'><H3> INTENTO FALLIDO</H3></FONT>";
    $intentos = $numero +1;
    echo "<HR ALIGN='CENTER' WIDTH='50%'><BR>
<FONT COLOR='WHITE'>Ha hecho $intentos intentos </FONT><BR><BR>
REGRESAR AL MENÚ INICIAL<BR><BR> <A HREF='forAdm1.php'>REGRESAR</A></FONT><BR><BR>
</CENTER><HR ALIGN='CENTER' WIDTH=50%><BR>";
}
If($borrar)
{//echo "Borramos valores en campos";

```

```

for_adm ("", "", "Limpio campos");
echo"
<center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033>
<EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}

else if ( !$enviar) // verificamos si no se ha pulsado el botón 'Enviar datos'
{ ECHO "<BR>";
for_adm ($RFC, $Login, $Password, " ");
echo" <center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid:D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf><PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033> <EMBED src=ayuda_H.swf quality=high bgcolor=#000033
WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}

else
{
    $numero = contador();
    // echo " valor "+ $numero;
    if ($numero >= 4)
    { IntenOlos($numero); }
    else {
        if( !empty($RFC) && !empty($Login) && !empty($Password)) // verificamos si los campos tienen datos
        { if ( valida_RFC ($RFC)!="Verifique el RFC" ) // Validamos los campos del formulario
            $RFC = valida_RFC ($RFC);

            //echo "la curp: $RFC El nombre es $nombre ";
            if ( $RFC!="Verifique el RFC")
            // Si los campos son validos es prudente hacer la conexión con la base de datos y el valor de la curp
            $conexion = Conectarse (); // funcion para conectarse a la base de datos
            $sql = "select * from administrador where rfc='$RFC'";
            $resultado = mysql_query ($sql, $conexion);
            $filas = mysql_num_rows ($resultado); // el número de filas en la consulta
            if ($filas == 1) // Se realizo la conexión a la base de datos el usuario existe
            {
                while ($row = mysql_fetch_array ($resultado)) // numero de elementos que tiene la tabla
                {
                    // echo "Los valores de la tabla son <br><br>El login
                    <FONT BGCOLOR="BLUE">$row [login] </FONT> <br>El password $row [password]";
                    if (($Login == $row [login]) && ($Password == $row[password] ))
                    // Verifico que el login y password coincidan
                    {
                        include ("Cambios2.php");
                    }
                    else { // El password Login es incorrecto
                        for_adm ($RFC, "", "", "El password o Login es incorrecto");
                    }
                }
            }
        }
    }
}

echo"
<center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf><PARAM NAME=quality VALUE=high>

```

```

<PARAM NAME=bgcolor VALUE=#000033> <EMBED src=ayuda_H.swf quality=high bgcolor=#000033
WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</Center>
";
}
} mysql_free_result ($resultado);
}
else
{//El usuario no existe
for_adm ("$RFC", $Login, "", "Verifique su RFC <font color=#009955> $RFC </font>");
echo"
<center><table Width=100%><tr><td align=right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#000033><EMBED src=ayuda_H.swf quality=high bgcolor=#000033
WIDTH=650 HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}
}
else { //RFC INVÁLIDO

for_adm ($RFC, Login, $Password, "El usuario no existe verifique el RFC");
echo" <center>
<table Width=100%><tr><td align=right>
<OBJECT classid=clsid: D27CDB6E-AE6D-11cf-96B8-444553540000
codebase=http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0
WIDTH=650 HEIGHT=95 id=ayuda_H ALIGN=>
<PARAM NAME=movie VALUE=ayuda_H.swf> <PARAM NAME=quality VALUE=high> <PARAM
NAME=bgcolor VALUE=#000033><EMBED src=ayuda_H.swf quality=high bgcolor=#000033 WIDTH=650
HEIGHT=95 NAME=ayuda_H ALIGN=
TYPE=application/x-shockwave-flash PLUGINSPPAGE=http://www.macromedia.com/go/getflashplayer></EMBED>
</OBJECT>
</center> ";
}
}
else
{// Formulario con datos incompletos

for_adm ($RFC, $Login, $Password, "Valores incompletos en los campos");
}
}
}
?>
</BODY>
</HTML>

```

#### Código del proceso de Altas

```

<HTML><HEAD><TITLE>FORMULARIO</TITLE></HEAD>
<BODY BgColor=#000033 Text=#FFFFFF Link=#CC9900 VLink=#CC9900 ALink=#CC9900>
<script language=javascript>
function ventanaAlumno (URL) {
    window.open (URL, "Alumno", "width=500, height=600, scrollbars=YES")
}
</script>
<script language=javascript>
function ventanaProfesor (URL) {
    window.open (URL, "Profesor", "width=500, height=600, scrollbars=YES")
}
</script>

```

```

<script language=javascript>
function ventanaProyecto (URL) {
    window.open (URL,"Proyecto","width=600, height=666666600, scrollbars=YES")
}
</script>
<script language=javascript>
function ventanaAltaArea (URL) {
    window.open (URL,"AltaArea","width=500, height=410, scrollbars=NO")
}
</script>
<script language=javascript>
function ventanaAltaDepartamento (URL) {
window.open (URL,"AltaDepartamento","width=500, height=500, scrollbars=NO")
}
</script>
<script language=javascript>
function ventanaMostrarArea (URL) {
window.open (URL,"MostrarArea","width=500, height=400, scrollbars=YES")
}
</script>
<script language=javascript>
function ventanaMostrarDepartamento (URL) {
window.open (URL,"MostrarDepartamento","width=600, height=500, scrollbars=YES")
}
</script>
<script language=javascript>
function ventanaMostrarOtros (URL) {
window.open (URL,"MostrarOtros","width=500, height=500, scrollbars=YES")
}
</script>

<?php
include ("Funciones_Formulario2.php");
//include ("curp.inc");
include ("RegFormularios.php");
include ("funciones.php");
error_reporting (E_ error); //No muestra los errores ni las variables que cometen esos errores

/* Esta función muestra los valores que se almacenaran en diversas lablas.
Se pretende mostrar el formulario principal para guardar proyectos de TESIS O SEMINARIOS
Para los niveles de LICENCIATURA O POSGRADO De la DIME
Se obtienen valores del la TESIS.... del DIRECTOR, ALUMNOS,
DEPARTAMENTOS AL QUE PERTENECE EL PROYECTO
y otros datos importantes */

function Tesis ($pagina, $texto, $Tesis, $Info)
{include ("arreglos.php"); //Se incluyen los arreglos para utilizar en este formulario
// error_reporting (E_ ERROR);
//Nivel de errores críticos para reportar errores y no mostrar en pantalla mensaje de error
$conexion = Conectarse (); //Conexion a la BD

/* Los valores del arreglo $Info se extraen con list ()... y mostramos las variables que
se utilizaran en esta función... */

list($Proy,$sDepto,$sArea,$NivelAI,$NivelPos,$NumAI,$FDiaR,$FMesR,$FAnoR,$FDiaI,$FMesI,
$FAnoI, $FDiaF, $FMesF, $FAnoF) = $Info;

ECHO <FORM NAME=datos ACTION='$pagina' METHOD='GET' >
<TABLE border='3' BgColor='#CC9900' Width=100%>
<TR><TD><FONT Color='#000033' Size='4'><B><center>$texto</center></B> </FONT></TD></TR>
</TABLE>;

/* Declaramos el select para el tipo de Proyecto TESIS O SEMINARIO utilizamos
un arreglo $Proyecto para almacenar las opciones, de ahí la utilización del foreach()
Donde obtenemos la clave del arreglo y el valor que contiene el arreglo.
Se obtiene el valore seleccionado en el () a través de una comparación entre la clave $idProy del arreglo

```

y la variable del select \$Proy (esto en código php).  
 En esta función se utilizan de la misma forma los select para obtener el valor que se ha seleccionado. \*/

```

ECHO " <TABLE Cellspacing=10 >
<TR><TD>Proyecto
<select Name='Proy' OnChange='document.datos.submit ();>";
foreach ($Proyecto as $idProy => $CampoProy)
{ECHO "<OPTION VALUE='$idProy' ";
if ($idProy == $Proy)
ECHO "SELECTED";
ECHO " > $CampoProy</OPTION>";}
ECHO " </SELECT></TD>

<TD>Departamento";
/* La primera vez que se ejecuta no se ha elegido " Departamento" ni " Área ",
Mostramos el primer valor de cada tabla en Área y Departamento en los select...
estos valores están en la base de datos, así desplegamos el select de todos
los departamentos y las áreas que contiene cada departamento respectivamente
de forma automática */

$conulta = "SELECT * FROM departamento";
$Depto = mysql_query ($conulta); //Se obtienen todos los departamentos de la BD
$lineaD = mysql_num_rows ($Depto);

ECHO "<SELECT NAME='sDepto' OnChange='document.datos.sArea.selectedIndex=1; document.datos.submit ();>";
for ($i=0; $i < $lineaD; $i++)
{
    $id=mysql_result ($Depto, $i, 0); // clave del registro
    $Nombre_Depto = mysql_result ($Depto, $i, 1); //valor del registro (nombre)
    ECHO"<OPTION VALUE='$id'";
    if ($id == $sDepto)
    ECHO " SELECTED ";
    ECHO">$Nombre_Depto</OPTION>";
}
ECHO "</SELECT></TD>";
ECHO"<TD Align=right >Area";

/* Ahora se obtiene la información de las Áreas para el departamento elegido
Se selecciona de la base de datos la tabla de áreas con el valor obtenido del select
de Departamento $sDepto */

$con_aux = "select * from area where id_Depto=$sDepto";
$Area = mysql_query ($con_aux); //Se obtienen las areas de la base de datos
$lineaA = mysql_num_rows ($Area);
if ($lineaA) //Si la consulta es exitosa y existen valores en el Area se genera el resto
{ECHO "<SELECT NAME='sArea' document. $datos.submit ();>";
for ($i=0; $i < $lineaA; $i++)
{ $id=mysql_result ($Area, $i, 0); // clave del registro
$Nombre_Area = mysql_result ($Area, $i, 1); //valor del registro (nombre)
ECHO "<OPTION VALUE='$id' ";
if ($i == $sArea)
ECHO "SELECTED ";
ECHO " > $Nombre_Area </OPTION>";
}
ECHO "</SELECT></TD></TR>";
}
else //Si la consulta no es exitosa y existen mostramos el select sin valores en el Area
{
?>
<TD><SELECT NAME='sArea' onChange='document.datos.submit ()>
<OPTION VALUE='0' SELECTED>El Departamento no tiene Areas definidas </OPTION>
</SELECT></TD></TR>
<?php
}

/* Mostramos el nivel del proyecto LICENCIATURA O POSGRADO con un arreglo $Tipo
Dependiendo el seleccionado, en el select de Posgrado se muestran las opciones para
cada nivel, facilitando al usuario las opciones disponibles en estos campos
que se almacenarán en la base de datos */
    
```

```

// Mostramos el valor para saber cuantos alumnos serán registrados para este proyecto
echo " <TD> N&uacute; m. de Alumnos
<select Name='NumAl' onChange='document.datos.submit ();>";
foreach ($NumAlumno as $idNumA => $CampoNumA)
{echo "<OPTION VALUE='$CampoNumA' ";
if ($CampoNumA == $NumAl)
echo "SELECTED";
echo " > $CampoNumA</OPTION>";}
echo "</SELECT></TD>";
ECHO"<TD Align=CENTER >Nivel
<select Name='NivelAl' onChange='document.datos.submit ();>";
foreach ($Tipo as $idN => $CampoN)
{ECHO "<OPTION VALUE='$idN' ";
if ($idN == $NivelAl)
ECHO "SELECTED";
ECHO " > $CampoN</OPTION>"; }
ECHO "</SELECT>
Posgrado
<SELECT Name='NivelPos' onChange='document.datos.submit ();>";
foreach ($NivelAl as $idNPA => $CampoNP)
{echo "<OPTION VALUE=$idNPA ";
if ($idNPA==$NivelPos)
echo" SELECTED ";
echo "> $CampoNP</OPTION>";}
echo "</SELECT></TD></tr>";
ECHO"</TR></TABLE>";

/* Mostramos los valores importantes para la tesis.....algunos son opcionales
otros son Indispensables (*). */

printf ("<center>Título del Proyecto (*) </TD><TD>
<INPUT Type='Text' Name='Tesis [Nombre]' Size='100' Maxlength='240' value='%s'>
<br><br>Objetivo del Proyecto (*) </TD><TD>
<INPUT Type='Text' Name='Tesis [Objetivo]' Size='100' Maxlength='240' value='%s'>

</center><br><center>
<TABLE Cellspacing=5 >
<TR><TD> Clave Del Registro (*) </TD><TD>
<INPUT Type='Text' Name='Tesis [Clave]' SIZE='8' Maxlength='7' value='%s'></TD>
<TD Align=right > RFC de Director (*) </TD><TD>
<INPUT Type='Text' Name='Tesis [RFC]' Size='10' Maxlength='10' value='%s'></TD>
</TR>
<TR><TD Align=right> Temario</TD><TD>
<TEXTAREA Name='Tesis [Temario]' Cols='20' Rows='3'></TEXTAREA></TD>
<TD Align=right> Comentarios</TD><TD>
<TEXTAREA Name='Tesis [Comentario]' Cols='20' Rows='3' ></TEXTAREA></TD>
</TR>
<TR><TD>Tipo de Investigación</TD><TD>
<INPUT Type='Text' Name='Tesis [tipo]' Size='20' value='%s'></TD>
<TD>Campo de Investigación</TD><TD>
<INPUT Type='Text' Name='Tesis [Investigacion]' Size='20' value='%s'></TD>
</TR>
<TR><TD Align=right>Financiamiento</TD><TD>
<INPUT Type='Text' Name='Tesis [Financiamiento]' Size='20' value='%s'></TD>
<TD Align=right>Avance</TD><TD>
<INPUT Type='Text' Name='Tesis [Avance]' Size='6' Maxlength='5' value='%s'></TD>
</TR>
</TABLE></center>", $Tesis [Nombre], $Tesis [Objetivo], $Tesis [Clave], $Tesis [RFC],
$Tesis [tipo], $Tesis [Investigacion], $Tesis [Financiamiento], $Tesis [Avance]);

ECHO"<TABLE BORDER='3' BGCOLOR='#CC9900' WIDTH=100%>
<tr><TD><FONT Color='#000033' Size='4'><B>Fechas</B></FONT></TD></TR>
</TABLE>";

// Esta es la parte para mostrar las fechas que son importantes para el registro de las tesis
ECHO "<TABLE Cellspacing=10 ><TR>
<TD ALIGN='LEFT'> Fecha de Registro del Proyecto (*)
<SELECT Name='FDIaR' >";

```

```

foreach ($Dia as $IdDR => $CampoDR)
{ECHO "<OPTION VALUE=$CampoDR";
if ($CampoDR == $FDIaR)
ECHO "SELECTED";
ECHO " > $CampoDR</OPTION>";}
ECHO "</SELECT>
<SELECT Name='FMesR' >";
foreach ($Mes as $IdMR => $CampoMR)
{ECHO "<OPTION VALUE=$IdMR' ";
if ($IdMR== $FMesR)
ECHO "SELECTED";
ECHO " > $CampoMR</OPTION>";}
ECHO "</SELECT>
<SELECT Name='FAnoR' >";
foreach ($Ano as $IdAR => $CampoAR)
{ECHO "<OPTION VALUE=$IdAR' ";
if ($IdAR == $FAnoR)
ECHO "SELECTED";
ECHO " > $CampoAR</OPTION>";}
ECHO "</SELECT></TD>
<TD ALIGN='CENTER'>Fecha Fin del Proyecto (*)
<SELECT Name='FDIaF' >";
foreach ($Dia as $IdDF => $CampoDF)
{ECHO "<OPTION VALUE=$CampoDF";
if ($CampoDF == $FDIaF)
ECHO " SELECTED ";
ECHO " > $CampoDF</OPTION>";}
ECHO "</SELECT>
<SELECT Name='FMesF' >";
foreach ($Mes as $IdMF => $CampoMF)
{ECHO "<OPTION VALUE=$IdMF' ";
if ($IdMF == $FMesF)
ECHO "SELECTED";
ECHO " > $CampoMF</OPTION>";}
ECHO "</SELECT>
<SELECT Name='FAnoF' >";
foreach ($Ano as $IdAF => $CampoAF)
{ECHO "<OPTION VALUE=$IdAF' ";
If ($IdAF == $FAnoF)
ECHO "SELECTED ";
ECHO " > $CampoAF</OPTION>";}
ECHO "</SELECT></TD></TR>

</tr></table>
<center>
<table><tr><TD><center>
<INPUT type =submit NAME='Enviar' value='Enviar' >
<INPUT type=SUBMIT value='Borrar' name='borrar' >

<center> </TD>
</tr></table>
</center>
<br><br>
<A Href=javascript: ventanaAltaArea ('CopiaForAreas.php')><B><FONT SIZE=4>
Muestra el Formulario para dar de Alta una Área </font></b></A><br><br>
<A Href=javascript: ventanaAltaDepartamento ('CopiaForDepartamentos.php')>
<B><FONT SIZE=4>Muestra el Formulario para dar de Alta un Departamento </font></b></A>

<table Width=100%>
<TR><TD Allgn=right >
<A Href=javascript: ventanaProyecto ('Aproyectos.html')><IMG Src=ayuda_Proyecto.gif></A>
<A Href=Cambios2.php><IMG Src=flecha_0.gif Alt="Regreso a Altas"></A>
<A Href=index.html target=_top><IMG Src=home_0.gif></A>
</TD></TR></TABLE>";
ECHO"</TABLE></FORM>
";
}

```

```

/* Inicia la aplicación en esta parte */

IF ($borra)
{$Info = array ("$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR",
"$FAnoR","$FDial","$FMesI","$FAnol","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Limpia datos","", $Info);
}
else IF ($Actualizar)
{$Info = array ("$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR","$FAnoR",
"$FDial","$FMesI","$FAnol","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Limpia datos", $Tesis, $Info);
}

else if (!$Enviar)
{// Inicializamos las variables cuando es la primera vez que se ve el formulario
if (!isset ($Proy)) $Proy = "Tesis";
if (!isset ($sDepto)) $sDepto = 1;
if (!isset ($sArea)) $sArea = 0;
if (!isset ($NivelAI)) $NivelAI = "Licenciatura";
if (!isset ($NivelPos)) $NivelPos = "Ninguno";
if (!isset ($NumAI)) $NumAI = "0";
if (!isset ($FDiaR)) $FDiaR = "0";
if (!isset ($FMesR)) $FMesR = "0";
if (!isset ($FAnoR)) $FAnoR = "0";
if (!isset ($FDial)) $FDial = "0";
if (!isset ($FMesI)) $FMesI = "0";
if (!isset ($FAnol)) $FAnol = "0";
if (!isset ($FDiaF)) $FDiaF = "0";
if (!isset ($FMesF)) $FMesF = "0";
if (!isset ($FAnoF)) $FAnoF = "0";

/* Almacenamos todas las variables en un arreglo para manipular la información más fácilmente
a través de él evitando tener muchos valores que complican el entender las funciones */

$Info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR",
"$FAnoR","$FDial","$FMesI","$FAnol","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Proporcione los Datos para el registro de Proyectos","", $Info);
}
else {
if (empty ($Proy) || empty ($sDepto) || empty ($NivelAI) || empty ($NivelPos) ||
empty ($NumAI) || empty ($Tesis [Clave]) ||
empty ($Tesis [Nombre]) || empty ($Tesis [Clave]) || empty ($Tesis [RFC])
)
{Info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR","$FAnoR",
"$FDial","$FMesI","$FAnol","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Faltan datos en campos", $Tesis, $Info);
}
else
{/* Primero verificamos que la clave no este registrada */
$conexion = Conectarse ();
$sql = "SELECT * from catalogo_Tesis where NoRegistro = '$Tesis [Clave]'";
// $sql = "SELECT * from catalogo_tesis where idTesis = '$Tesis [Clave]'";
$resultado = mysql_query ($sql, $conexion);
$filas = mysql_num_rows ($resultado);

//echo " FILAS: $filas ";
Cerrar ($conexion);
if ( registro ($Tesis [Clave])!="Registro Invalido" && $filas == 0 )
{
if ((valida_RFC ($Tesis [RFC]))!= "Verifique el RFC") //verifico que el RFC sea valido
{
if (valida_fecha ($FDiaR, $FMesR, $FAnoR)!= "Esa fecha no existe" &&
valida_fecha ($FDiaF, $FMesF, $FAnoF)!= "Esa fecha no existe")
{

/*echo "<CENTER><FONT COLOR=GRENEN>Datos validos</FONT></CENTER>";
Procedo a verificar que el valor del RFC se encuentre alojado en la tabla de PROFESOR
si es así muestro el valor, en caso contrario, tendré que dar una alta de profesor en la tabla */
$conexion = Conectarse (); // función para conectarse a la base de datos
$sql = "select * from profesor where rfc_prof = '$Tesis [RFC]'";

```

```

$resultado = mysql_query ($sql, $conexion);
$filas = mysql_num_rows ($resultado); // el número de filas en la consulta
if ($filas == 1)
{echo "<TABLE><TH BGCOLOR=#CC9900>EI PROFESOR con RFC
<FONT COLOR=blue> $Tesis [RFC] </FONT> esta registrado </TH></TABLE>";
if ($guardar) // Se guardan datos de Números de cuenta
(//Procedemos a dar de alta a los alumnos
//Informacion= array ($Tesis [RFC], $Tesis [NumAl], $enviar);
//NumerosCuenta ($Informacion);
$VALOR=RepetidoNumeros ($Numero);
If (CadenasEMPTY ($Numero) && $VALOR==1)
{
IF (CadenasNumerosCuenta ($Numero))
{for ($i=0; $i<= count ($Numero); $i++)
echo " <center>Numero de cuenta valido $Numero {$i} </center>";*/
// Funcion para verificar si el alumno registrado en la base de datos
list ($Registrado, $SinRegistro, $valor)=BaseAlumno ($Numero, $conexion);
echo "<CENTER><TABLE ><TR><TD>";
if (count ($Registrado) > 0)
{echo "<CENTER><TABLE BORDER=1 >
<TH BGCOLOR=#9999CC>ALUMNOS REGISTRADOS ".count ($Registrado)."
</TH><TH>Número de Cuenta</TH><TH><TH><TH>MOSTRAR</TH>";
foreach ($Registrado as $id1 => $Registro)
echo "<TR BGCOLOR=#CC9900><TD ALIGN=CENTER>Alumno $id1 </TD><TD>$Registro<TD></TR>";
ECHO"<TD ALIGN=CENTER> <A Href=javascript: ventanaMostrarArea ('Otros.php?tabla=area&Enviar=Enviar')>
<FONT COLOR=Black>Áreas</FONT> </A> ó
<A Href=javascript: ventanaMostrarDepartamento ('Otros.php?tabla=departamento&Enviar=Enviar')>
<FONT COLOR=Black> Departamentos </FONT> </A>
<BR>
<A Href=javascript: ventanaMostrarOtros ('Otros.php?')>
<FONT COLOR=Black> Otras Tablas </FONT> </A> </TD> ";
ECHO "</TABLE></CENTER>";
$Finicio="$FDiaR-$FMesR-$FAnoR";
$FFin="$FDiaF-$FMesF-$FAnoF";
$ano= substr ($Tesis [Clave], 0, 2);
if( substr($ano,0,1) == 0 || substr($ano,0,1) == 1 )
$ano = "20$ano";
else
$ano = "19$ano";
$semestre =substr ($Tesis [Clave], 0, 2).substr ($Tesis [Clave], 2, 2);

// Normalizamos campos
$Tesis [Nombre] = normaliza ($Tesis [Nombre]);
$Tesis [Objetivo]= normaliza ($Tesis [Objetivo]);
$Tesis [Temario]= normaliza ($Tesis [Temario]);
$Tesis [tipo] = normaliza ($Tesis [tipo]);
$Tesis [Investigacion] = normaliza ($Tesis [Investigacion]);
$Tesis [Financiamiento] = normaliza ($Tesis [Financiamiento]);

/*Mostramos el formularios de los datos del proyecto */
ECHO"<CENTER><TABLE BORDER=1 Width=95%>

<TR BGCOLOR=#9999CC>
<TH><FONT COLOR=WHITE>Tipo de Proyecto </FONT></TH>
<TH><FONT COLOR=WHITE>Nivel </FONT></TH>
<TH><FONT COLOR=WHITE>Posgrado </FONT></TH>
<TH><FONT COLOR=WHITE>Calve de registro</FONT></TH>
<TH><FONT COLOR=WHITE>Número de alumnos</FONT></TH>
<TH><FONT COLOR=WHITE>RFC del Director</FONT></TH>
</TR><TR BGCOLOR=#CC9900>
<TD align=center> $Proy</TD>
<TD align=center> $NivelAl</TD>
<TD align=center> $NivelPos</TD>
<TD align=center> $Tesis [Clave] </TD>
<TD align=center>. Count ($Numero). "</TD>
<TD align=center> $Tesis [RFC] </TD>
</TR><TR BGCOLOR=#9999CC>
<TH ><FONT COLOR=WHITE>Clave del Departamento </FONT></TH>
<TH ><FONT COLOR=WHITE>Clave del Área </FONT></TH>

```

```

<TH ><FONT COLOR=WHITE>Fecha de Registro </FONT></TH>
<TH ><FONT COLOR=WHITE>Fecha de Fin </FONT></TH>
<TH ><FONT COLOR=WHITE>Año</FONT></TH>
<TH ><FONT COLOR=WHITE>Semestre</FONT></TH>
</TR><TR BGCOLOR="\#CC9900" >
<TD align=center> $sDepto</TD>
<TD align=center> $sArea</TD>
<TD align=center>$FInicio</TD>
<TD align=center>$FFin</TD>
<TD align=center> $ano</TD>
<TD align=center> $semestre</TD>
</TR><TR BGCOLOR="\#9999CC" >
<TH colspan=6><FONT COLOR=WHITE>Titulo de Tesis </FONT></TH>
</TR><TR BGCOLOR="\#CC9900" >
<TD align=center colspan=6> $Tesis [Nombre] </TD>
</TR>
</TR><TR BGCOLOR="\#9999CC" >
<TH colspan=6><FONT COLOR=WHITE>Objetivo de la Tesis </FONT></TH>
</TR><TR BGCOLOR="\#CC9900" >
<TD align=center colspan=6> $Tesis [Objetivo]</TD>
</TR>
<TR BGCOLOR="\#9999CC" >
<TH colspan=6><FONT COLOR=WHITE>Temario de la Tesis </FONT></TH>
</TR><TR BGCOLOR="\#CC9900" >
<TD align=center colspan=6> $Tesis [Temario]</TD>
</TR>
<TR>
</TR><TR BGCOLOR="\#9999CC" >
<TH colspan=3><FONT COLOR=WHITE>Tipo de Investigacion </FONT></TH>
<TH colspan=3><FONT COLOR=WHITE>Campo de Investigación </FONT></TH>
</TR><TR BGCOLOR="\#CC9900" >
<TD align=center colspan=3> $Tesis [tipo] </TD>
<TD align=center colspan=3> $Tesis [Investigacion] </TD>
</TR>
<TR BGCOLOR="\#9999CC" >
<TD ALIGN=CENTER COLSPAN=2> Financiamiento </TD>
<TD ALIGN=CENTER colspan=2 BGCOLOR="\#CC9900" > $Tesis {Financiamiento} </TD </TD>
<TD ALIGN=CENTER > Avances </TD>
<TD ALIGN=CENTER BGCOLOR="\#CC9900" > $Tesis [Avance] </TD </TD>
</TR>
</TABLE></CENTER>";
/*Una vez que no hay alumnos sin registrar podemos enviar a la base la información */
if (count ($SinRegistro) == 0)
{foreach ($Numero as $clave => $valor) //Creamos la cadena de los números de cuenta a enviar
{$texto.="Numero [$clave]=$valor&";}
ECHO"
<A href='AltaTesis.php?
Proy=$Proy&Depto=$sDepto&Area=$sArea&Nivel=$NivelAI&Posgrado= $NivelPos&NoAlumnos=$NumAI&
Fecha=$FInicio&FechaF=$FFin&Año=$ano&Semestre=$semestre&Clave=$Tesis [Clave] &Titulo=$Tesis
[Nombre]&
Objetivo=$Tesis [Objetivo] &Temario=$Tesis [Temario] &Descripcion=$Tesis [Comentario] &tipoInves=$Tesis
[tipo] &campoInves=$Tesis [Investigacion] &financiamiento= $Tesis [Financiamiento] &rfc=$Tesis [RFC]
&avance=$Tesis [Avance] &$texto">
<IMG Src=flecha_1.gif align=center ></A>";
echo "Enviamos Información";
}
}
echo "</TD><TD>";
if (count ($SinRegistro) > 0)
{
echo " <PRE><FONT SIZE=3 bgcolor="\#cc9900" >
En esta parte registramos a los alumnos.
Seleccione la liga para registrar al alumno.
Una vez que ha registrado a un alumno Actualice el formulario.
Cuando todos los alumnos estén dados de alta se podrá registrar la tesis
</FONT></PRE><br><br> ";
}
}

```

```

echo "<TABLE BORDER=1 ><TH BGCOLOR=#9999CC1">
ALUMNOS SIN REGISTRO ".count ($SinRegistro)."</TH>
<TH>Número de Cuenta</TH>";
foreach ($SinRegistro as $id2 => $SinRegistro)
echo "<TR BGCOLOR=#CC99001"><TD ALIGN=CENTER>Alumno $id2</TD>
<TD> $SinRegistro</TD></TR>";
ECHO "</TABLE>";
$info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR",
"$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
AltaAlumno ($info, $Tesis, $Numero, $SinRegistro);
}
echo "</TD></TR></TABLE></CENTER> ";

}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR",
"$FMesR","$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
RegistroAlumnos ($Tesis,"NUMERO DE CUENTA INVALIDO.
Los alumnos REGISTRA SON $Tesis [NumAI]", $Numero, $info);
}
}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR",
"$FMesR","$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
RegistroAlumnos ($Tesis,"Faltan Datos O un Número de Cuenta esta repetida ", $Numero, $info);
}
}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR",
"$FMesR","$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
RegistroAlumnos ($Tesis,"Los alumnos a REGISTRAR SON $NumAI:", $Numero, $info);
}
}

else
{Cerrar ($conexion);
$info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR",
"$FMesR","$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
AltaProf ($info, $Tesis);
}
}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR",
"$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Verifica la fecha de registro o de fin", $Tesis, $info);
}
}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR",
"$FAnoR","$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Verifica el RFC del Director de tesis", $Tesis, $info);
}
}
else
($info = array ("$Proy","$sDepto","$sArea","$NivelAI","$NivelPos","$NumAI","$FDiaR","$FMesR","$FAnoR",
"$FDial","$FMesI","$FAnoI","$FDiaF","$FMesF","$FAnoF");
Tesis ("ForProyectos.php","Verificar la Clave de Registro (ver ayuda) ó esa
Clave de registro ya existe". $Tesis, $info);
}
}
}
?>
</BODY>
</HTML>

```

## Código para hacer cambios

```

<HTML><TITLE> FORMULARIO PAR ACTUALIZAR DATOS DEL ALUMNO </TITLE></HEAD>
<BODY BgColor="#000033" Text="#CC9900">
<CENTER>
<?php
include ("funciones.php");
include ("conec.php");

$arreglo = array ("nombre_alumno" => "Nombre: ",
"apellido_P" => "Apellido Paterno: ",
"apellido_M" => "Apellido Materno: ",
"sexo" => "Sexo: ",
"nombre_nacionalidad" => "Nacionalidad:",
"rfc_alumno"=> "RFC: ",
"tipo_nivel" => "Nivel de estudios: ",
"tipo_Posgrado"=> "Posgrado: ",
"nombre_plantel"=> "Plantel: ",
"nombre_carrera"=> "Carrera: ",
"f_ingreso"=> "Fecha de Ingreso: ",
"f_egreso"=> "Fecha de Egreso: ",
"promedio"=> "Promedio: ",
"edo_examen" => "Estado del examen: ",
"avances_creditos" => "Avance en créditos:",
"beca"=> "Beca: ",
"email"=> "Email: ",
"calle" => "Calle: ",
"numero_Ext" => "Número Exterior: ",
"numero_Int" => "Número Interior: ",
"colonla" => "Colonia: ",
"codigo_postal" => "C.P: ",
"nombre_edo" => "Estado: ",
"nombre_delegacion" => "Delegación: ",
"Municipio" => "Municipio: ",
"telefono" => "Teléfono:"
);

funcion CadenasEMPTY ($Numero)
{ $cont=0;
//Primero establezco que existe valor en el campo utilizo 1 para cierto o falso
for ($i=1; $i<= count ($Numero) ; $i++)
{list ($valor) = each ($Numero);
if ( !empty ($Numero[$valor]) )
$valor {$i} = 1; // Existe valor
else
{ $valor {$i} = 0; //No existe valor
$cont++;
}
}
//Ahora verifico que todos los valores coincidan con la bandera 1 que establece que el valor es correcto
if ($cont >=1)
Return $salida=0; //Falta valor en los campos
else
return $salida=1; //Valor en los campos
}

funcion Mostra_Campos ($Num_Cuenta, $a1, $TEXTO, $arreglo)
{
//echo "Ha seleccionado un campo mínimo "+$resultado;
echo "<CENTER><FORM METHOD=\GET" ACTION=\ActualizarAlumnos.php">
<font color=\#cc9900\ SIZE=4>$TEXTO </font><BR><BR>
<TABLE BORDER=\2">
<TH GCOLOR=\#CC9900\><FONT COLOR=\BLACK\>CAMPO</FON></TH><TH
BGCOLOR=\#CC9900\><FONT COLOR=\BLACK\>ANOTE SU NUEVO VALOR</FONT></TH>";

While ($res=each ($a1))

```

```

(*Se pone las ligas para mostrar Información que se debe proporcionar el las cajas de texto *)
$! = $res [0]; //obtenemos la clave

if ($! == "nombre_nacionalidad")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B>Lista de Naciones:
</B></FONT></TD><TD><B> LIGA</B></TD></TR>";

else if ($! == "tipo_nivel")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B>El Nivel puede ser:
</B></FONT></TD><TD><B>
'Licenciatural' ó 'Posgrado' </B></TD></TR>";

else if ($! == "tipo_Posgrado")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B>El Nivel en Posgrado puede ser:
</B></FONT></TD><TD>
<B> 'Maestria','Especialidad' ó 'Doctorado' </B></TD></TR>";

else if ($! == "nombre_plantele")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B> Lista de Planteles
</B></FONT></TD><TD><B> LIGA </B></TD></TR>";

else if ($! == "nombre_carrera")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B> Lista de Carreras:
</B></FONT></TD><TD><B> LIGA</B></TD></TR>";

else if ($! == "Edo_examen")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033><B>
El Estado del Examen puede ser: </B></FONT></TD><TD>
<B> 'En Proceso', 'Aprobado' ó 'No Aprobado' </B></TD></TR>";
else if ($! == "sexo")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033>
<B> EL sexo puede ser: </B></FONT></TD><TD><B>'Hombre'
ó 'Mujer' </B></TD></TR>";

else if ($! == "beca")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033>
<B> EL alumno es becado: </B></FONT></TD><TD><B>
'Becado' ó 'No Becado'</TD></TR>";

else if ($! == "nombre_edo")
echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033>
<B> EL estado puede ser : </B></FONT></TD><TD><B>LIGA</B></TD></TR>";
else if ($! == "nombre_delegacion")

echo "<TR><TD BGCOLOR=#CC9900><FONT COLOR=#000033>
<B> La delegación puede ser: </B></FONT></TD><TD>
<B>LIGA</B></TD></TR>";
//Se muestra la caja de texto con el valor anterior, de cada campo seleccionado

printf ( "<TR><TD BGCOLOR=#043023> <B><FONT COLOR=#CC9900>$arreglo [$!] </FONT></B></TD>
<TD BGCOLOR=#CC9900><INPUT TYPE='TEXT' NAME='Alumno [$]'"
VALUE=%s' SIZE=40> </TD></TR>". $res [1]);
}
echo "<TR><TD BGCOLOR=#CC9900>
<INPUT TYPE='HIDDEN' NAME='Enviar' VALUE='Enviar'"
<INPUT TYPE='SUBMIT' NAME='ACTUALIZA' VALUE='Actualizar datos'"></TD>
<TD BGCOLOR=#CC9900><INPUT TYPE='RESET' NAME='LIMPIA'"
VALUE='Borra cambios realizados'"></TD></TABLE></FORM></CENTER>";
}

function Normalizamos_valores ($Alumno)
{
while (llst ($clave, $valor) = each ($Alumno))
{
if ($clave=="rfc_alumno")
$Dato [$clave]= strtoupper ($valor);
else if ($clave == "email")
$Dato [$clave] = strtolower ($valor);
}
}

```

```

else
$Dato [$clave]= normaliza ($valor);
}
return $Dato;
}

/* Verifico con esta funcion si los valores son validos para las cadenas de caracteres */
funcion cadenas_Validas ($valor)
{
    if (cadena ($valor) == "No es una cadena de caracteres" )
        return $bandera = 0;
    else
        return $bandera=1;
}

/* Verifico si los valores numéricos son correctos */

funcion numeros_Validos ($clave, $valor)
{
    if (es_numero ($valor) == "No es un número")
        return $bandera=0;
    else
        return $bandera=1;
}

/* Validamos los campos con esta funcion. */
funcion valido_camposActualizoAlumno ($Alumno)
{
    // $bandera =1; // campo valido
    $contador=0;
    /* Verificamos que los valores en los campos sean correctos */
    while (list ($clave, $valor)= each ($Alumno))
        {
            if ($clave=="nombre_alumno" || $clave == "apellido_P" || $clave
            == "apellido_M" || $clave == "sexo" || $clave=="nombre_nacionalidad"
            || $clave=="tipo_nivel" || $clave=="tipo_Posgrado" || $clave=="nombre_plantel" || $clave == "nombre_carrera" ||
            $clave=="calle" || $clave=="colonia" || $clave == "nombre_edo" || $clave=="Municipio" || $clave == "nombre_delegacion"
            || $clave == "Edo_examen" || $clave=="beca" )

                {
                    $bandera = cadenas_Validas ($valor);
                    if ($bandera == 0)
                    {
                        $contador++;
                        $texto [$clave]=$clave;
                    }
                    else if ($clave=="f_ingreso" || $clave=="f_egreso" || $clave=="numero_Ext" || $clave=="numero_Int"
                    || $clave=="codigo_postal" ||
                    $clave=="telefono")
                    {
                        $bandera= numeros_Validos ($clave, $valor);
                        if ($bandera == 0)
                        {
                            $contador++;
                            $texto [$clave]=$clave;
                        }
                    }
                    else if ($clave == "rfc_alumno")
                    {
                        if (valida_RFC ($valor) == "Verifique el RFC" )
                        {
                            $contador++;
                            $texto [$clave]="RFC";
                        }
                    }
                    else if ($clave=="email")
                    {
                        if (valido_email ($valor) == "No es una dirección de correo electrónico" )
                        {
                            $contador++;
                            $texto [$clave]="Email";
                        }
                    }
                    else if ($clave == "promedio")
                    {
                        if (promedio ($valor) == "No es un promedio valido")
                        {
                            $contador++;
                            $texto [$clave]="Promedio";
                        }
                    }
                }
        }
}

```

```

        else if ($clave == "avances_creditos")
        {
if (es_numero ($dato [avances_creditos]) == "No es un número" &&
(strlen ($dato [avances_creditos]) == 3 ||
(strlen ($dato [avances_creditos]) == 2) )
($contador++; $texto {$clave}="Avances en créditos";
)
}
}
}
$datos= array ($contador, $texto);
return $datos;
}

/* funcion cadenaDeSexo ($clave, $valor)
{
switch ($valor)
{
case 'Hombre':
$dato= $valor;
break;
case 'Mujer':
$dato=$valor;
break;
default
$dato="No escribió un valor de sexo";
break;
}
return $dato;
}

*/

funcion VerificaLosDatos ($Alumno)
{ $contador=0;

while (list ($clave,$valor) = each($Alumno) )
{
switch ($clave)
{
case 'sexo':
$dato = cadenasDeSexo ($valor);
if ($dato == "No escribió un valor de sexo correctamente")
{ $contador++;
$Cadena {$clave} = "$clave";
}
break;

case 'nombre_nacionalidad':
$dato = cadenaDeNaciones ($valor);
if ($dato == "No escribió el valor de la nación correctamente")
{$contador++;
$Cadena {$clave} = "$clave";
}
break;

case 'nivel':
$dato = cadenaDeNivel ($clave);
if ($dato == "No escribió un valor de nivel correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'Posgrado':
$dato = cadenaDePosgrado ($valor);
if ($dato == "No escribió un valor de Posgrado correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
}
}
}

```

```

break;
case 'nombre_Plantel':
$dato = cadenaDePlanteles ($valor);
if ($dato == "No escribió un valor de plantel correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'nombre_carrera':
$dato = cadenaDeCarreras ($valor);
if ($dato == "No escribió un valor de carrera correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'edo_Examen':
$dato= cadenaDeExamen ($valor);
if ($dato == "No escribió un valor de examen correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'beca':
$dato= cadenaDeBeca ($valor);
if ($dato == "No escribió un valor de beca correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'nombre_estado':
$dato= cadenaDeEstados ($valor);
if ($dato == "No escribió un valor de estado correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;

case 'nombre_delegacion':
$dato= cadenaDeDelegaciones ($valor);
if ($dato == "No escribió un valor de la delegación correctamente")
{$contador++;
$Cadena {$clave}= "$clave";
}
break;
} //cierra switch

} //cierra while
$dato = array ($contador, $Cadena);
return $dato;
}

/* Esta funcion regresa la clave de las tablas, delegación, nacionalidad, estado etc
tabla nombre de la tabla a consultar
*/

function RegresaClaves ($tabla, $id, $columna, $valor)
{
$conexion = Conectarse ();
$sql = "SELECT $id from $tabla where $columna='$valor' ";
$resultado = mysql_query ($sql);
$id=mysql_result ($resultado, 0, 0);
Cerrar ($conexion);
return $id;
}

```

```

funcion ActualizarConsultas ($Alumno)
{
  while (list ($clave,$valor)= each ($Alumno))
  {
    switch ($clave)
    {
      case "nombre_alumno": case "apellido_P": case "apellido_M": case "sexo": case "calle": case "numero_int":
      case "numero_Ex": case "colonia": case "codigo_postal": case "email": case "Municipio": case "id_Estado":
      case "Id_Delegacion": case "id_Nacionalidad": case "telefono":

        break;

      case "id_nivel_A": case "f_ingreso": case "f_egreso": case "promedio": case "avances_creditos":
      case "tipo_nivel": case "tipo_Posgrado": case "id_Carrera": case "id_Plantel": case "Edo_Examen":
      case "beca":
        break;
    }
  }
}

$bandera = CadenasEMPTY ($Alumno);
if ($bandera == 1)
{
  $Alumno = Normalizamos_valores ($Alumno);
  list ($validos, $texto)=valido_camposActualizoAlumno ($Alumno);
  if ($validos == 0)
  {
    // Echo "<CENTER><font color='\"#cc9900\"'> Datos validos $validos </font>";
    //$contador = 0;
    // list ($contador, $Cadena)=VerificaLosDatos ($Alumno);
    if ($contador == 0)
    {
      echo "<font color='\"#cc9900\"'> Datos validos $contador</font>";
      echo "<TABLE BORDER=1 BGCOLOR='\"#cc9900\"'>
      <TH><B><FONT
      COLOR='\"#000033\"'>Campo</FONT></B></TH><TH><B><FONT
      COLOR='\"#000033\"'>VALORES $validos</FONT></B></TH>";
      for ($i=0; $i< count ($Alumno); $i++)
      {list ($clave, $valor) = each ($Alumno);
      echo"<TR><TD><B><FONT COLOR='\"#000033\"'>$clave </FONT></B></TD>
      <TD><B><FONT COLOR='\"#000033\"'>$valor</FONT></B></TD>
      </TR>";
      }
      echo "<TR><TD><B><FONT COLOR='\"#000033\"'> Salvamos datos con la flecha </font></TD>
      <TD align=center><A Href='GuardaAlumnoDatos.php? Cuenta=$Cuenta'><IMG Src=flecha_1.gif align=center ></A>
      </TABLE></CENTER>";
      $clave= RegresaClaves ("delegacion","Id_Delegacion","nombre_delegacion","Iztapalapa");
      echo "<font color='\"#cccc33\"'>valor de la clave es$clave </FONT>";
      }
    else // Algunos campos no son lo que se piden
    {
      echo "<font BGCOLOR=RED>LOS VALORES QUE NO SON CORRECTOS </FONT>";
    }
  }
  else
  {
    $campos =Faltan_datos ($texto);
    Mostra_Campos ($Num_Cuenta, $Alumno,"<br><font color='\"#cc9900\"'> $campos</FONT> ", $arreglo);
  }
}
else
{
  echo "<b><FONT COLOR='\"#000033\"'>$texto</FONT></B>";
  ECHO "<TABLE><TR><TD>";
  Mostra_Campos ($Num_Cuenta, $Alumno,"Fallan datos en los campos", $arreglo);
  ECHO "</TD><TD>";
  echo "<TABLE BORDER=1 BGCOLOR='\"#cc9900\"'>
  <TH><B><FONT
  COLOR='\"#000033\"'>Campo</FONT></B></TH><TH><B>
  <FONT COLOR='\"#000033\"'>VALORES</FONT></B></TH>";
}

```

```
for ($i=0; $i< count ($Alumno); $i++)
{list ($clave, $valor) = each ($Alumno);
echo"<TR><TD><B><FONT COLOR=#000033> $arreglo [$clave] </FONT></B></TD>";
if ($valor == "")
$valor="<B><FONT COLOR=BLUE>Sin valor </FONT></B>";
ECHO " <TD><B><FONT COLOR=#000033> $valor </FONT></B></TD>
</TR>";
}
echo "</TABLE>";
ECHO "</TD></TR></TABLE>";
}
?>
</BODY></HTML>
```

# *Bibliografía*

Ingeniería de Software

Roger S. Pressman

Mc Graw Hill

España 2002

Ingeniería de Software

Shari Lawrence Pfleeger

Prentice Hall

Buenos Aires, Argentina 2002

Apuntes de Ingeniería de programación

Ing. Orlando Zaldívar Zamorategui

México D.F. 2000

Aprendiendo MySQL

Mark Maslakowski y Tony Butcher

Prentice Hall

México 2001

Guía Práctica para usuarios de PHP 4

Esteban Trigos García

Anaya Multimedia

Madrid, España 2001

**Referencias:**

<http://www.econ.uba.ar/www/departamentos/sistemas/>

<http://www.aceproject.org/main/espanol/et/et105c.htm>

[http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900\\_MTFernandez.pdf](http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_MTFernandez.pdf)

[http://www.gestiopolis.com/recursos2/documentos/fulldocs/fin/modproolga\\_archivos/frame.htm](http://www.gestiopolis.com/recursos2/documentos/fulldocs/fin/modproolga_archivos/frame.htm)

<http://elies.rediris.es/elies9/4-2-3.htm>

<http://www.monografias.com/trabajos7/bada/bada.shtml>

[http://www.odatech.com.ar/papers/Prueba\\_de\\_Integracion\\_para\\_Componentes\\_Software.pdf](http://www.odatech.com.ar/papers/Prueba_de_Integracion_para_Componentes_Software.pdf)

[http://mail.udlap.mx/~pgomez/ing\\_sw/proceso\\_de\\_pruebas.ppt](http://mail.udlap.mx/~pgomez/ing_sw/proceso_de_pruebas.ppt)

<http://www.itver.edu.mx/so1/clas.html>

[http://server2.southlink.com.ar/vap/sistemas\\_operativos.htm](http://server2.southlink.com.ar/vap/sistemas_operativos.htm)

<http://148.202.148.5/cursos/cc321/fundamentos/temario.htm>

*Manual del usuario*



Al ingresar en sistema el usuario se encontrará con la pantalla principal la cual cuenta con cinco opciones diferentes. Las cinco opciones que se encuentran en esta página corresponden a los diferentes tipos de usuario que pueden ingresar. El usuario deberá escoger la opción que le corresponde.



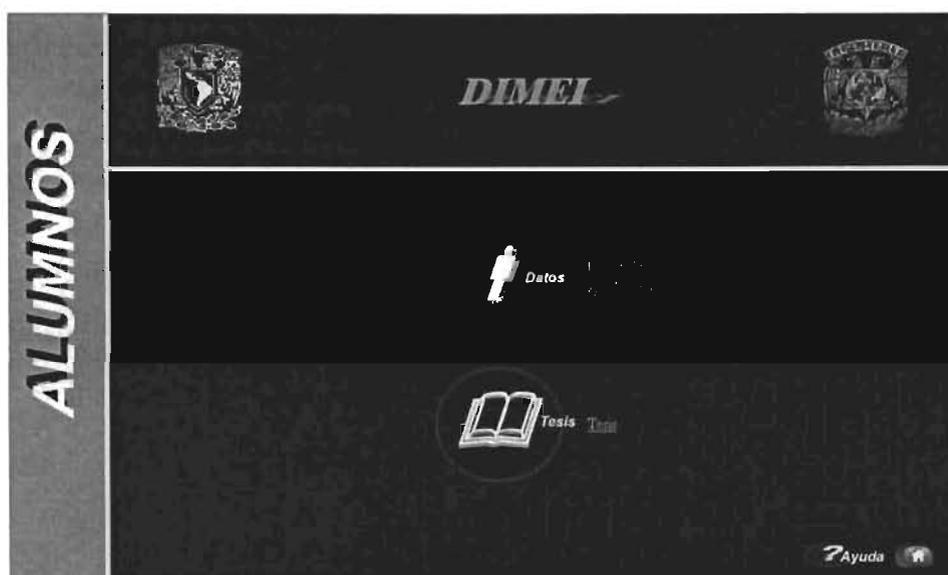
A continuación se explicarán a detalle cada una de las opciones presentadas en la pantalla principal.

### Opción Alumnos

Al dar clic en esta opción se mostrará la pantalla siguiente:



Esta pantalla nos presenta la información que el usuario necesita proporcionar al sistema para ingresar, en el tipo de información solicitada podemos encontrar, el RFC y número de cuenta. Para el número de cuenta agregue un cero en la parte más significativa si su generación es menor al 2000, además deberá omitir el guión que separa al dígito verificador. Por ejemplo para el número de cuenta 9223722-1 se tendría que escribir 092237221. En el caso del RFC escribirlo sin guiones entre las letras y los números, por ejemplo XATU780323. Cuando esta información ha sido verificada y aceptada por el sistema se muestra una nueva pantalla en la que el usuario puede seleccionar entre ver sus datos personales, académicos o la tesis que está elaborando.



Si el usuario selecciona ver sus datos académicos, encontrará información como el nombre, RFC, nacionalidad, etc.

The screenshot displays the 'ALUMNOS' section of the DIMET system. It features the system logo and two institutional emblems at the top. The main content is a table titled 'Datos personales' (Personal Data) with the following information:

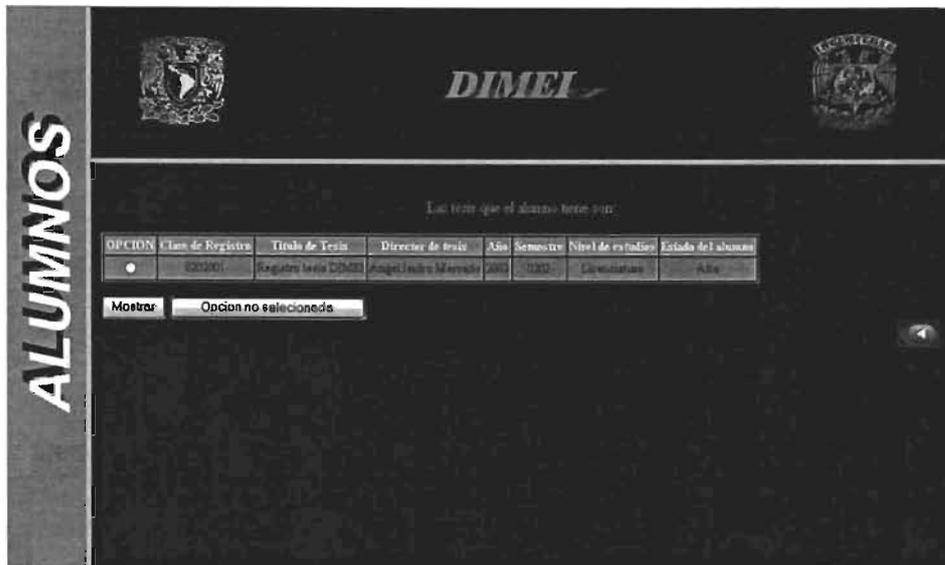
Datos personales	
Nombre	Xochitl Ghorety Reyes Aguirre
RFC	REAX770529
Sexo	MUJER
Nacionalidad	Mexicana
Dirección	Calle Rio Nazas No. # 17 Col. Valle de San Lorenzo C.P. 9970 Del. Iztapalapa Edo. Distrito Federal
Teléfono	58457500
Email	ghoretyr@mexico.com

En el caso de los datos académicos se mostrará una pantalla en la que los datos a mostrar son: el número de cuenta, año de ingreso a la licenciatura, año de egreso, Promedio, etc.

The screenshot displays the 'ALUMNOS' section of the DIMET system. It features the system logo and two institutional emblems at the top. The main content is a table titled 'Datos académicos' (Academic Data) with the following information:

Datos académicos	
Núm. de cuenta	093196452
Año de ingreso	1997
Año de egreso	2004
Promedio	8.0
Avance en créditos	100
Nivel	Licenciatura
Carrera	Ingeniería Mecánica
Plantel	Facultad de Ingeniería
Examen	EN PROCESO

La opción de tesis lleva al usuario a una pantalla en la que se muestra la tesis que tiene dada de alta, en caso de estar cursando una maestría, doctorado o especialidad, también se mostrarán éstas. De tal manera que el alumno puede escoger ver cualquiera de ellas.



Después de seleccionar la Tesis que desea ver de clic en el botón de mostrar, el cual regresará una pantalla que muestra toda la información que fue recaudada por la Coordinación de Titulación al momento de ser registrada.

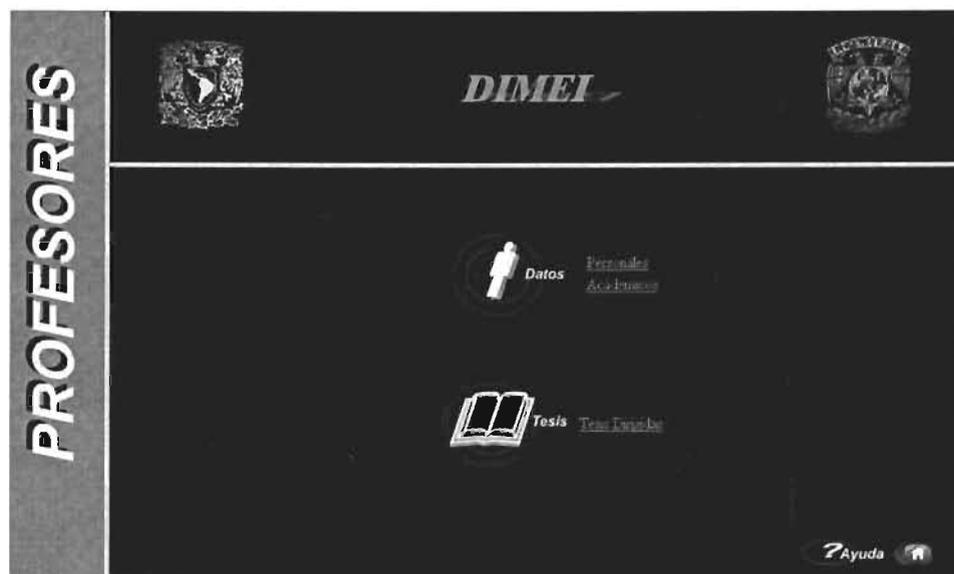


### Opción Profesor

Cuando se da clic en esta opción la información solicitada por el sistema es el RFC y su password. En cuyo caso el RFC debe ser escrito sin guiones entre las letras y los números. El password es una clave proporcionada por el profesor, cuando se da de alta en el sistema, para poder acceder a este.



Quando la información a sido verificada y aceptada por el sistema se presenta otra pantalla en la cual aparecen las opciones de ver datos personales, académicos y tesis dirigidas.



Los datos que se presentan al seleccionar datos personales son el nombre, RFC, dirección, etc. Los cuales se presentan de la siguiente forma:

The screenshot shows the 'DIMEI' interface with a vertical sidebar labeled 'PROFESORES'. The main content area is titled 'Datos personales' and contains a table with the following information:

Nombre	Isidro Mercado
RFC	ima770528
Sexo	Hombre
Nacionalidad	Mexicana
Dirección	Calle Cantera # 37 Col. Las Cruces C.P. 103330 Del. Magdalena Contreras
Teléfono	54252012
Email	angelitomercado@terra.com.mx

En cuanto a la información de carácter académico se encuentra el nivel de estudios, área de graduación nombramiento y tipo de profesor.

The screenshot shows the 'DIMEI' interface with a vertical sidebar labeled 'PROFESORES'. The main content area is titled 'Datos académicos' and contains a table with the following information:

RFC	ima770528
Nivel de estudios	Licenciatura
Área de Graduación	Mecánica
Nombramiento	Profesor de Asignatura
Tipo de Profesor	Profesor tipo A

Cuando se ha seleccionado la opción de Tesis dirigida En cuanto a la opción de tesis dirigidas primero se mostrará una pantalla que contiene todas las tesis. Para ver mayor información acerca de alguna en especial, selecciónela y de clic en el botón de mostrar, hecho esto el sistema mandará otra pantalla que contiene toda la información de la tesis.

**PROFESORES**




**DIMEI**

OPCION	Clave de Registro	Título de Tesis	Director de tesis	Año	Semestre
•	0202001	Registro tesis DIMEI	Angel Isidro Mercado	2002	0202
•	0201002	Sistema Para Protección De Redes	Angel Isidro Mercado	2002	0201
•	0402001	Frutas	Angel Isidro Mercado	2004	0402

Mostrar

**PROFESORES**



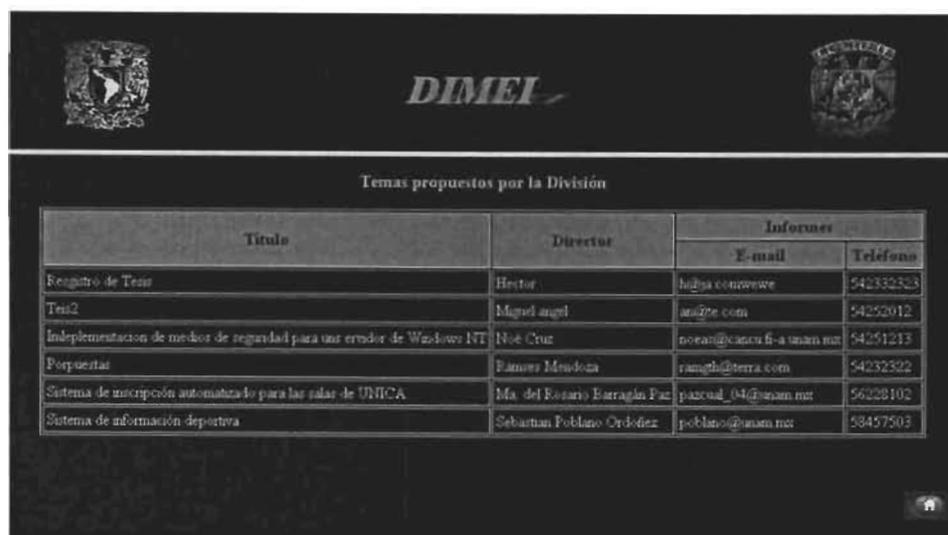

**DIMEI**

Tesis Dirigida

Registro	0202001
Proyecto	Tesis
Alumno o alumnos quienes realizan	Angel Isidro Mercado 093042912
Fecha de Inicio	13-02-2002
Fecha de Finalización	13-02-2004
Avances	0 %
Título	Registro tesis DIMEI
Objetivo	Objetivo Registrar alumnos de DIMEI
Temario	El temario
Descripción	Una descripcion
Departamento	Ingeniería en Termoenergía y Mejoramiento Ambiental
Área	Manufactura
Campo de Investigación	Campo de investigación

### Opción Tesis propuestas por la División

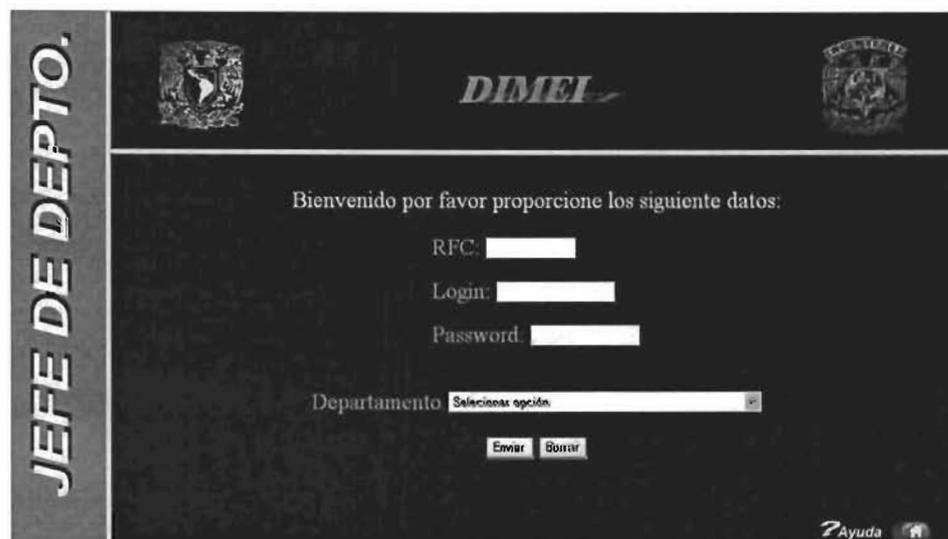
En esta opción todos aquellos alumnos que están en busca de un tema de tesis encontrarán temas propuestos por profesores de la División. Encontrando además del nombre del proyecto, el nombre del profesor que la propone y una forma de ponerse en contacto con éste para adquirir mayor información.



Titulo	Director	Informes	
		E-mail	Teléfono
Registro de Tesis	Hector	hija@conveve	542332323
Tesis2	Miguel Angel	ma@ze.com	54252012
Implementación de medios de seguridad para un servidor de Windows NT	Noé Cruz	noe@cantu5-a.unam.mx	54251213
Propuestas	Ramón Mendoza	ram@terra.com	54232322
Sistema de inscripción automatizado para las salas de UNICA	Ma. del Rosario Barragán Paz	pazrosal_04@unam.mx	56228102
Sistema de información deportiva	Sebastián Poblano Ordoñez	poblano@unam.mx	53457503

### Opción Jefes de Departamento

Para que el jefe de departamento pueda ingresar es necesario que el usuario proporcionar RFC, login, password y departamento del cual es jefe.



**JEFE DE DEPTO.**

Bienvenido por favor proporcione los siguiente datos:

RFC:

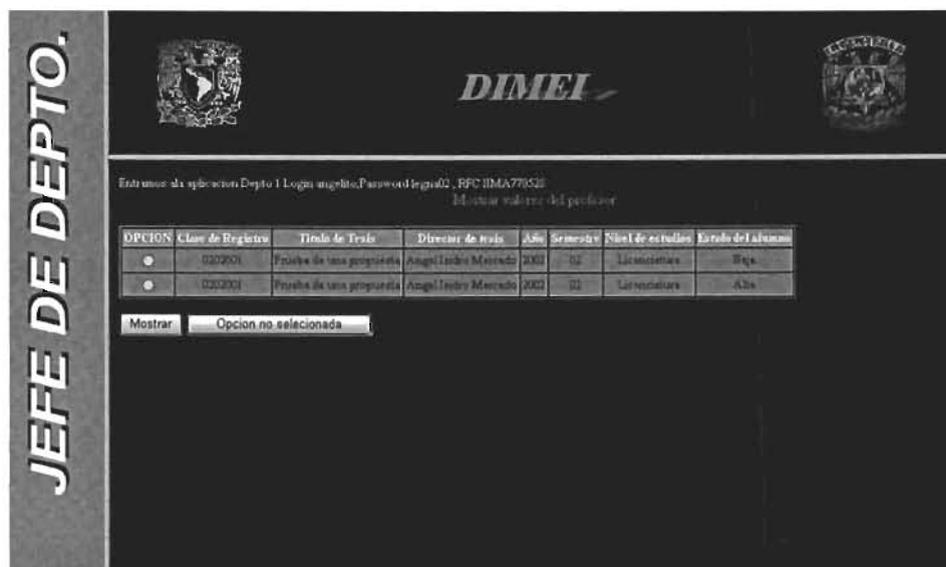
Login:

Password:

Departamento:

[Ayuda](#)

Una vez que los datos del jefe de departamento son identificados como correctos, se muestra una pantalla en la cual aparece una lista que contiene todas las tesis pertenecientes al departamento, pudiendo este seleccionar cualquiera de ellas para ver más información.

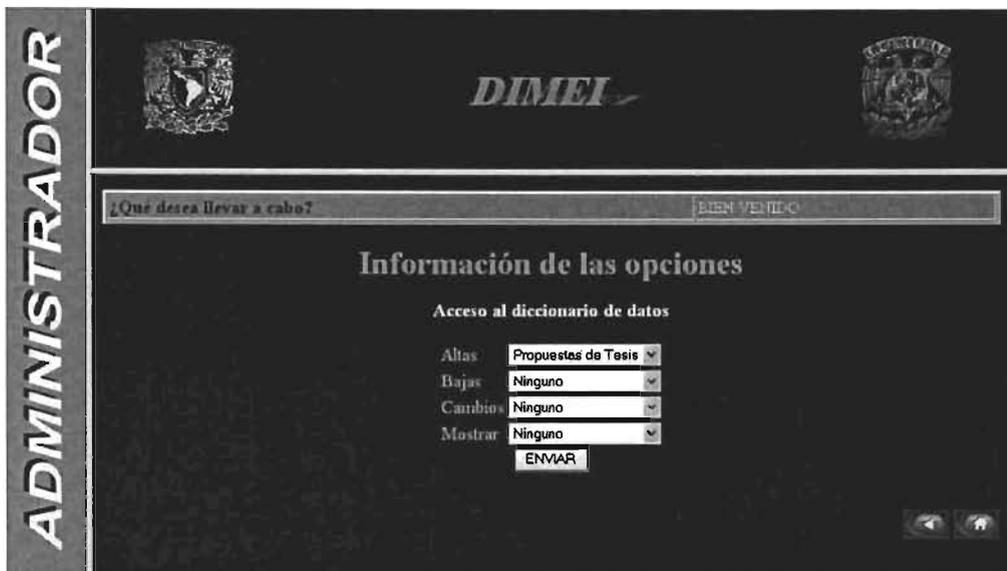


## Administrador

Esté usuario podrá entrar al sistema proporcionando la siguiente información: El RFC sin guiones, Password y login que es la palabra clave que identifica al usuario.



Una vez que la identificación del administrador a sido verificada el sistema muestra una pantalla en la que se puede seleccionar entre hacer altas, bajas, cambios y mostrar información.



Al desplegar la opción de altas se puede seleccionar entre dar de alta alumno, profesores, proyectos de tesis, propuestas de tesis y otros.

**Opción Alta de alumnos**

Si se desea ingresar un alumno seleccione esta opción y a continuación se mostrará un formulario en el que deberá ingresar primero los datos personales del alumno.



Antes de comenzar a llenar este formulario es importante que verifique que existe la nacionalidad a la cual pertenece el alumno. En caso de no encontrarse en la lista dicha nacionalidad, en la parte

inferior encontrará una liga, que proporciona el formulario para darla de alta. Una vez hecho esto de un clic en el botón de actualizar para poder ingresar los demás datos.

La información que se requiere es la siguiente y tendrá que ingresarla como a continuación se explica:

Nombre	Información de la casilla
Núm. Cuenta	En el Núm. de Cuenta agregué un cero si su generación es menor a 2000, omita el guión. Ejemplo, 094170010.
RFC	Digitarlo sin el guión entre las letras y los números. Por ejemplo REAX780213.
Nombre	Digitarlo en mayúsculas y minúsculas.
Apellido Paterno	Digitarlo en mayúsculas y minúsculas.
Apellido Materno	Digitarlo en mayúsculas y minúsculas.
Sexo	Seleccionar el correspondiente de la lista.
Calle	Digitarlo en mayúsculas y minúsculas.
Núm. Exterior	Digitar el número.
Núm. Interior	Digitar el número. En caso de no tener número dejar la casilla vacía.
Colonia	Digitarlo en mayúsculas y minúsculas.
Código Postal	Digitar el número.
E-mail	Digitarlo en minúsculas.
Municipio	Digitarlo en mayúsculas y minúsculas.
Estado	Seleccionar el indicado de la lista.
Delegación	Seleccionar la indicada de la lista.
Nacionalidad	Seleccionar la indicada de la lista.

Núm. Telefónico	Digitarlo sin espacios o guiones. Ejemplo, 56430986.
-----------------	--

Una vez que la información ha sido ingresada de clic en el botón de guardar datos y si la información ha sido aceptada por el sistema aparecerá una pantalla con un mensaje de que la información es válida y que ha sido realizada la alta.

ADMINISTRADOR

**DIMEL**

Se insertan valores en la tabla de alumnos

CAMPO	DATO
Número de cuenta (*)	093386762
RFC (*)	AAER770410
Nombre (*)	Reynaldo
Apellido Paterno (*)	Alamirano
Apellido Materno (*)	Estrada
Sexo (*)	Masculino
Edad (*)	Me voy
Número Exterior (*)	217
Número Interior	
Colonia (*)	Esperanza
Código Postal (*)	57800
Email (*)	reynaldos@hotmail.com

ADMINISTRADOR

**DIMEL**

Alta realizada de forma correcta  
 Los datos personales del Alumno se han almacenado.  
 Atención: revise la fecha ubicada debajo de los datos personales.  
 Para continuar con el registro de los Datos Académicos del Alumno.

Número de cuenta (*)	093386762
RFC (*)	AAER770410
Nombre (*)	Reynaldo
Apellido Paterno (*)	Alamirano
Apellido Materno (*)	Estrada
Sexo (*)	Masculino
Edad (*)	Me Voy
Número Exterior (*)	217
Número Interior	0
Colonia (*)	Esperanza
Código Postal (*)	57800
Email (*)	reynaldos@hotmail.com
Municipio	Texcoco de Mora
Estado (*)	Estado de México

Una vez hecho esto se dará paso a dar de alta la información académica y para ello en la parte inferior de esta pantalla aparece una flecha a la cual se deberá dar un clic para que mande la siguiente parte del formulario, donde la información será ingresada de acuerdo con lo siguiente:

En caso de que la carrera y el plantel de procedencia no se encuentren en la lista en la parte de abajo se encuentran dos ligas que proporcionaran los formularios correspondientes para dar de alta.

Nombre de la casilla	Información de la casilla
Núm.Cta.	En el Núm.Cta. Se agrega un cero si su generación es menor a 2000, omite el guión. Ejemplo, 094170010.
Año de Ingreso	Digitar el número con sus cuatro dígitos.
Año de Egreso	Digitar el número con sus cuatro dígitos.
Promedio	Promedio Actual del estudiante. Si el promedio es por ejemplo 8.1 deberá ser ingresado como 8.10
Avance en créditos	Colocar el porcentaje.
Nivel de Estudios	Seleccionar Licenciatura o Posgrado según corresponda.
Nivel de Posgrado	Seleccionar entre maestría, doctorado o especialidad.
Carrera	Escoja la carrera de procedencia.
Plantel	Institución donde se realizaron los estudios de Licenciatura.
Examen	Seleccione de la lista " En proceso, Aprobado o No Aprobado".
Beca	Seleccione una opción de la lista.

ADMINISTRADOR

DIMEI

de insertar valores en la tabla de datos siguientes

CAMPO	VALOR
Número de cuenta (*)	093366762
Año de ingreso (*)	1997
Año de egreso (*)	2004
Pronto de actual (*)	8.1
Avance en créditos	100
Nivel de estudios (*)	Licenciatura
Nivel de Postgrado	NINGUNA
Carrera (*)	Ingeniería Computación
Facultad (*)	Facultad de Ingeniería
Estado de Examen	En Proceso
Titula o no	No Becado

Limpiar campos    Guardar datos    Actualizar

Una vez ingresa la información el sistema muestra un mensaje de que la información fue aceptada.

ADMINISTRADOR

DIMEI

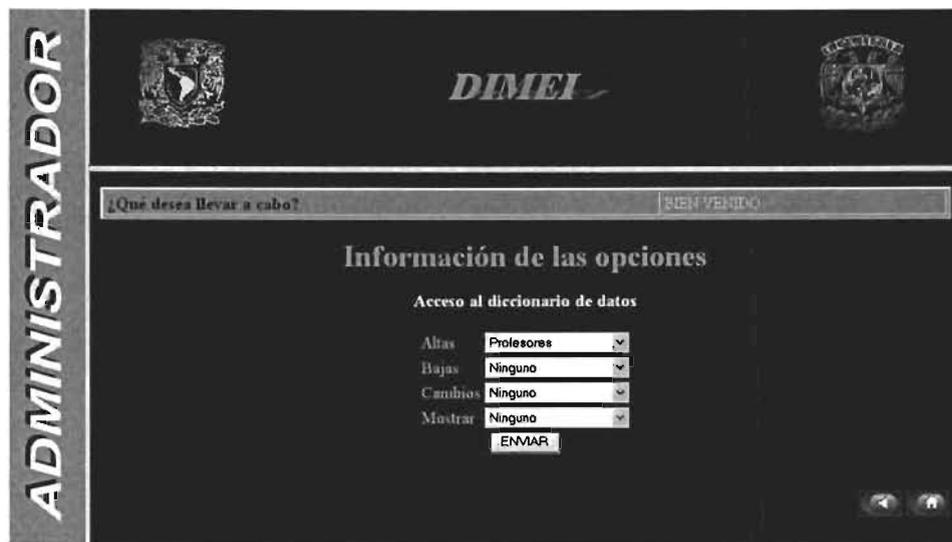
Campos validos

Número de cuenta (*)	094641925
Año de ingreso (*)	1999
Año de egreso (*)	2003
Pronto de actual (*)	9
Avance en créditos	100
Nivel de estudios (*)	Licenciatura
Nivel de Postgrado	Ninguna
Carrera (*)	?
Facultad (*)	?
Estado de Examen	AFROBADO
Titula o no	BECCADO

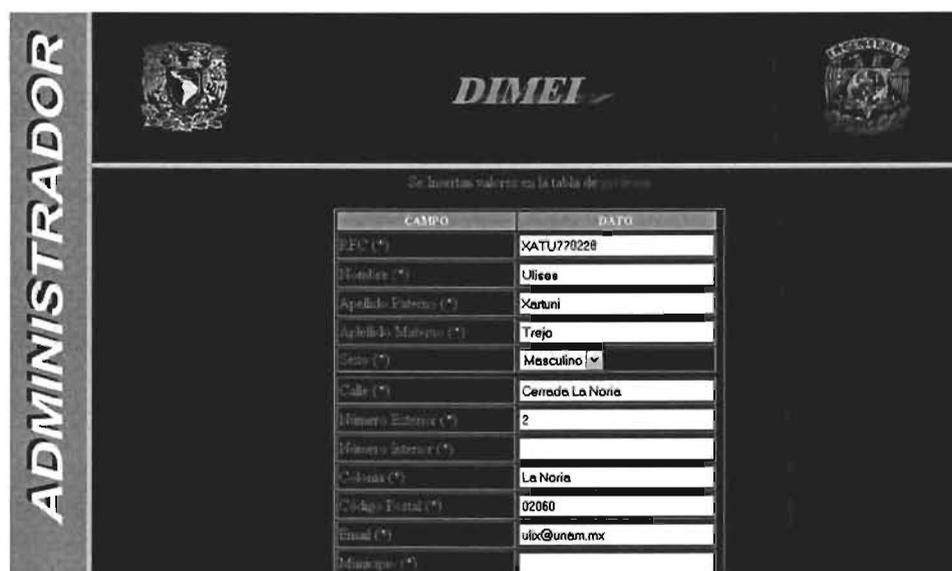
Alta realizada

### Opción Alta de profesores

Si se desea ingresar un profesor seleccione esta opción y de clic en el botón de enviar los datos académicos.



A continuación se proporcionará un formulario en el que deberá ingresar primero los datos personales.



La información deberá ser ingresada de acuerdo a las siguientes especificaciones:

Nombre de la casilla	Información de la casilla
RFC	Digitarlo sin el guión entre las letras y los números. Por ejemplo REAX780213.

Nombre	Digitarlo en mayúsculas y minúsculas.
Apellido Paterno	Digitarlo en mayúsculas y minúsculas.
Apellido Materno	Digitarlo en mayúsculas y minúsculas.
Sexo	Seleccionar el correspondiente de la lista.
Calle	Digitarlo en mayúsculas y minúsculas.
Número Exterior	Digitar el número.
Número Interior	Digitar el número.
Colonia	Digitarlo en mayúsculas y minúsculas.
Código Postal	Digitar el número.
Correo Electrónico	Digitarlo en minúsculas.
Municipio	Digitarlo en mayúsculas y minúsculas.
Estado	Seleccionar el indicado de la lista.
Delegación	Seleccionar la indicada de la lista.
Nacionalidad	Seleccionar la indicada de la lista.
Nivel de estudios	Seleccionar una opción de la lista.
Área de graduación	Seleccionar la indicada de la lista.
Nombramiento	Seleccionar el indicado de la lista.
Tipo de Profesor	Seleccionar de la lista.
Número Telefónico	Digitarlo sin espacios o guiones. Ejemplo, 56430986.

Una vez ingresada la información de clic en el botón de guardar datos. Si la información fue ingresada de manera correcta el sistema mostrara una pantalla en la que se muestra la leyenda de confirma que los datos son correctos y ya se encuentran en la base de datos.

## Opción Alta de Proyectos

Seleccione la opción de dar de alta un proyecto de Tesis y de clic en el botón enviar para que el sistema envíe un formulario para agregar la información.

Antes de comenzar a llenar este formulario verifique el departamento y el área a la cual pertenecerá el proyecto existen. En caso de que el departamento o el área no existan, en la parte de

debajo de este formulario se encuentran dos ligas las cuales al darles clic nos proporcionan los formularios para dar de alta esta información.

Una vez que lo anterior se haya llevado a cabo la información que requiere el sistema deberá ser ingresada de acuerdo a las especificaciones a continuación presentadas:

Nombre de la casilla	Información de la casilla
Proyecto	Escoger la opción de Tesis o Seminario, de acuerdo al tipo de proyecto.
Departamento	Seleccione el Departamento al cual pertenece el proyecto.
Área	Clasificar el proyecto en una de las Áreas consideradas para el Departamento seleccionado.
Nivel	En nivel puede ser Licenciatura o Posgrado.
Posgrado	Si el proyecto es de Posgrado, deberá seleccionarse entre Especialidad, Maestría o Doctorado.
Núm. de Alumnos	Dato numérico. Escoger el número de integrantes para el proyecto
Clave del registro	La clave del registro esta compuesta por el año, el semestre y tres dígitos que pueden ser aleatorios. Por ejemplo 0401130, donde 04 corresponde al año, 01 corresponde al semestre y finalmente últimos tres dígitos que son aleatorios.
Titulo del proyecto	Digitarlo en mayúsculas y minúsculas, sin comillas.
RFC del Director	Digitarlo sin el guión entre las letras y los números. Por ejemplo REAX780213.
Objetivo	Digitar el objetivo del proyecto.
Temario	Digitar el Temario del proyecto.
Descripción	Digitar aquí cualquier información que considere relevante, para ampliar o aclarar algún concepto.
Comentarios	Digitar aquí cualquier información que considere relevante, para ampliar o aclarar algún concepto.

Tipo de investigación	Digitarlo en mayúsculas y minúsculas.
Campo de investigación	Clasificar el proyecto en uno de los cinco campos de la ciencia considerados.
Financiamiento	Se solicita se indique la o las fuentes de financiamiento del proyecto, si es que existen.
Avance del Proyecto	Dato de tipo numérico que va desde 0 hasta el 100, sin el símbolo de porcentaje.
Fechas	Tienen el formato día, mes y año. En el se debe colocar la fecha en que se registro el proyecto y el la fecha final la que el departamento le asigne.

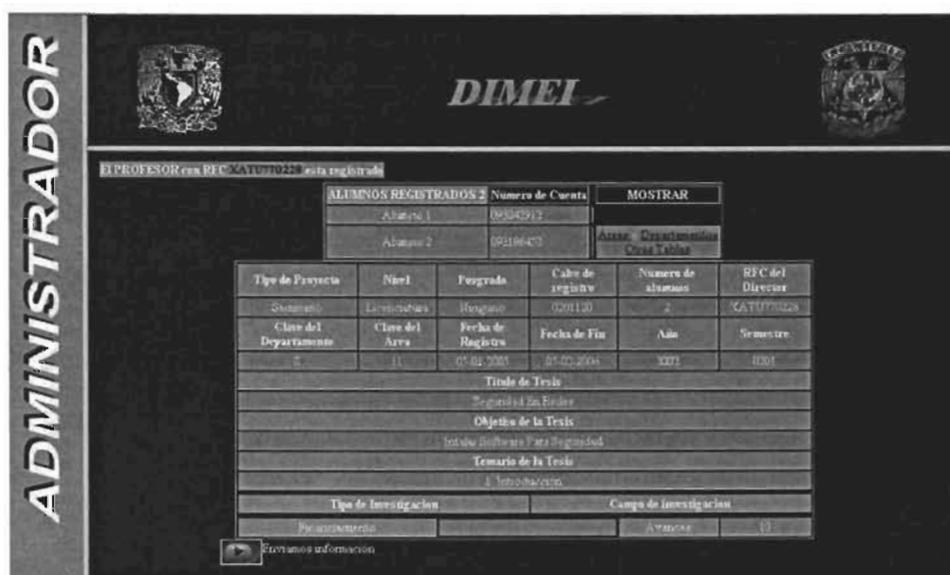
The screenshot shows the 'ADMINISTRADOR' interface for DIMEI. The main heading is 'Preparar los Datos para el registro de Proyectos'. The form contains the following fields and values:

- Proyecto: Seminario
- Departamento: Computación
- Area: Redes
- Nom. de Alumno: 2
- Nivel: Licenciatura
- Posgrado: Ninguno
- Título del Proyecto (\*): Seguridad en Redes
- Objetivo del Proyecto (\*): Iniciar software para seguridad
- Clave del Registro (\*): 0104130
- RFC de Director (\*): XATU770226
- Tema: 1. Introducción, 2. desarrollo, 3. conclusiones
- Comentarios: Tesis de prueba
- Tipo de Investigación:
- Campo de Investigación:
- Financiamiento:
- Avance: 10
- Fechas: Fecha de Registro del Proyecto (\*): 05 Enero 2005; Fecha Fin del Proyecto (\*): 05 Febrero 2006

Una vez que ya fue ingresada la información de clic en el botón de guardar. Hecho esto el sistema le proporcionará un formulario en el que tendrá que teclear el número de cuenta de los alumnos que presentan el tema de tesis. En caso de que estos no se encuentren dados de alta se le proporcionará el formulario para dar del alta al alumno.



Después de ingresados los números de cuenta, el sistema muestra una pantalla con toda la información recaudada para que sea verificada antes de ser mandada a la base de datos. Si la información es correcta de clic en el botón de Enviamos información.



**Opción Alta otros**

En opción se encuentran albergados otros formularios para dar de alta, áreas, jefes de departamento, departamentos, nacionalidades, etc. Al dar clic en esta opción se presenta la siguiente pantalla, en la que se presentan varias opciones:



Como se puede observar en la pantalla se encuentran originalmente tres opciones sin embargo al dar clic en la liga de mostrar otros formularios, se manda una lista de la cual se pueden seleccionar otro formularios.

#### Opción de Alta de un jefe de departamento

De clic en esta opción para que aparezca una pantalla como la que se muestra a continuación:

CAMPO	DATO
Tipo de jefe (*)	Seleccionar opción
BPO del jefe (*)	
Password (*)	
Correo (*)	
edad (*)	

Limpiar campos    Guardar datos    Actualizar

Muestra el formulario para dar una alta de profesor  
Muestra el formulario para buscar profesores  
Muestra el formulario para dar una alta de tipo Profesor

Es importante que la persona a la cual se vaya a registrar como jefe de departamento, también este dado de alta en la tabla de profesores. Por lo tanto siga los siguientes pasos:

1. Ver liga "Mostrar el formulario para buscar un profesor", mismo que mostrará el RFC y el nombre de los profesores.
2. Sólo en caso de no encontrar el RFC del profesor ir al formulario "Mostrar alta de profesor" y registrarlo.
3. Realizar la alta del jefe una vez registrado sus datos en el formulario de profesor.

Realizados los pasos anteriores podrá llegar el formulario, proporcionando la información como a continuación se indica:

<b>Nombre de la casilla</b>	<b>Información de la casilla</b>
Tipo de jefe	Seleccionar de la lista.
Password	Es una clave proporcionada por el Jefe para acceder al sistema.
Login	Es una palabra que identifica al usuario.
RFC	Digitarlo sin el guión entre las letras y los números. Por ejemplo REAX780213.
E-mail	Digitarlo en minúsculas.

ADMINISTRADOR

DIMEI

Se ingresaron datos en la tabla de jobs

CAMPO	DATO
tipo de job (*)	Seleccionar opción
EPZ del job (*)	
Password (*)	
Login (*)	
email (*)	

Muestra el formulario para dar una alta de profesor  
 Muestra el formulario para buscar profesores  
 Muestra el formulario para dar una alta de tipo Profesor

Ya ingresados los datos de clic en el botón guardar datos. El sistema mandará un aviso de que los datos han sido guardados correctamente en caso contrario mandará un aviso de error.

ADMINISTRADOR

DIMEI

Alta realizada de forma correcta Campos validos

tipo de job (*)	1
EPZ del job (*)	NATOSBITO1
Password (*)	1234
Login (*)	1234
email (*)	1234@1234567890.com

### Opción Alta de un departamento

Seleccionada está opción de clic y se mandará un formulario que tiene el siguiente aspecto:

La información solicitada para dar de alta un departamento deberá ser proporcionada con las siguientes especificaciones:

En la parte de abajo del formulario encontrará una liga que muestra una consulta de todos los jefes existentes en la base de datos y sus claves. Seleccione la clave correspondiente al jefe de departamento, en caso de que el jefe de departamento no este en la tabla que muestra la consulta, se proporciona otra liga que despliega el formulario para dar de alta a un jefe de departamento.

Nombre de la casilla	Información de la casilla
Clave del departamento	Proporcionada por la división
Nombre del departamento	Escribir en mayúsculas y minúsculas.
Teléfono	Escribir el número sin espacios o guiones. Ejemplo, 56332462.
Clave del jefe de departamento	Dar clic en el hipervínculo para ver la clave.

The screenshot shows the 'ADMINISTRADOR' interface for 'DIMEI'. The main content area displays a form titled 'De insertar valores en la tabla de departamentos'. The form has two columns: 'CAMPO' and 'DATO'. The fields are as follows:

CAMPO	DATO
Clave del Departamento (*)	<input type="text"/>
Nombre del Departamento (*)	<input type="text"/>
Numero telefonico del Departamento (*)	<input type="text"/>
Descripción	<input type="text"/>
Clave del jefe de departamento (*)	<input type="text"/>

Below the form are three buttons: 'Limpiar campos', 'Guardar datos', and 'Actualizar'. Below the buttons are three links: 'De clic para ver las claves de los departamentos', 'De clic para ver las claves de los jefes', and 'Muestra el formulario para dar el alta de un jefe'. At the bottom right, there is a 'Ayuda' button and navigation icons.

Ingresados los datos, el sistema manda una un mensaje de que los datos fueron guardados en la base de datos.

The screenshot shows the 'ADMINISTRADOR' interface for 'DIMEI' displaying a confirmation message: 'Alta realizada forma correcta'. Below the message is a table showing the details of the entered data:

Clave del Departamento (*)	13
Nombre del Departamento (*)	Mantenimiento De Equipos
Numero telefonico del Departamento (*)	66228101
Descripción	Mantenimiento
Clave del jefe de departamento (*)	El campo no existe

### Opción Alta de un área

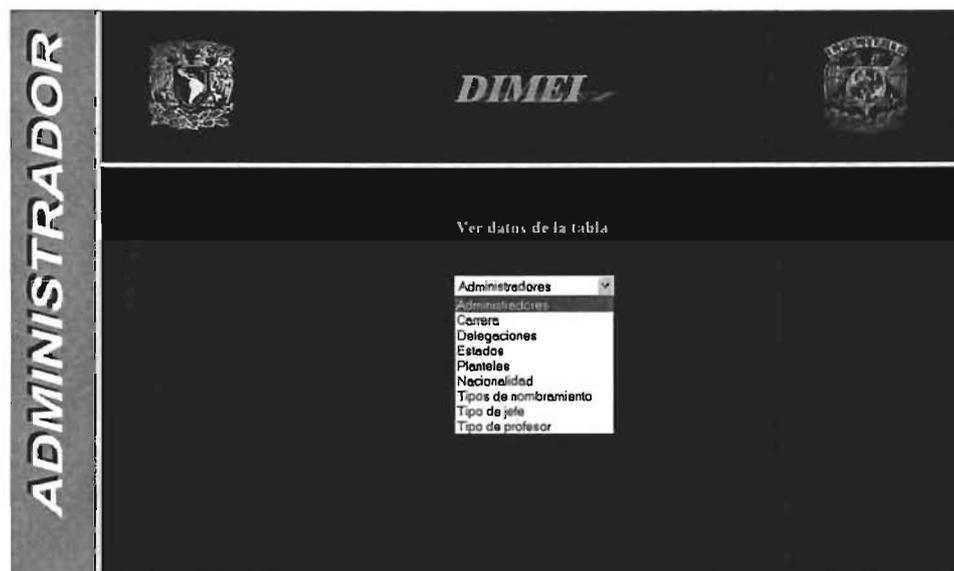
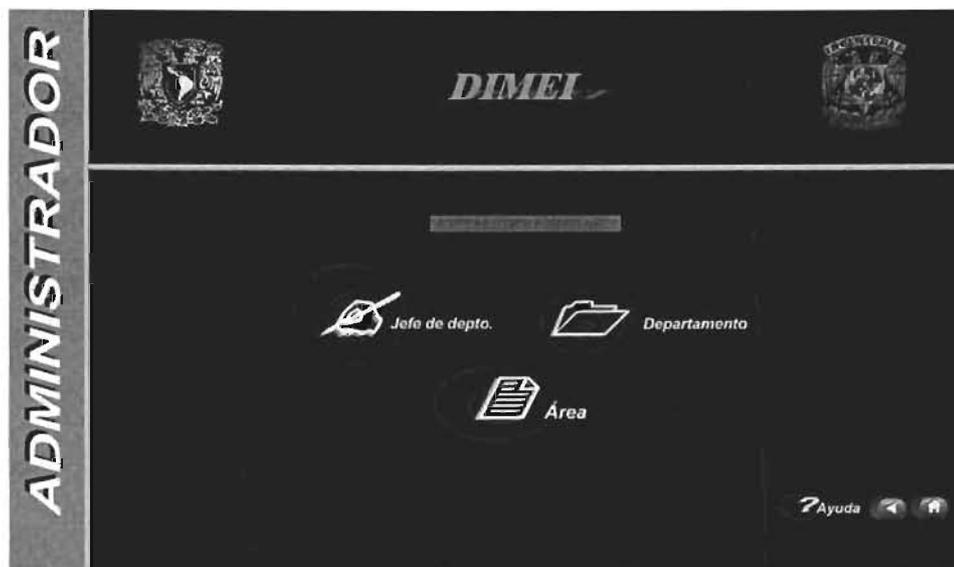
Al seleccionar el dar de alta un área se muestra un formulario que pide la siguiente información:

Nombre de la casilla	Información de la casilla
Nombre del área	Escribir en mayúsculas y minúsculas.
Descripción	Información acerca de las actividades realizadas en el área.
Clave del jefe de departamento	Seleccionar el correspondiente de la lista.

The screenshot shows the 'ADMINISTRADOR' interface for DIMEL. It features a header with the DIMEL logo and two crests. Below the header, there is a message: 'Se insertan valores en la tabla de areas'. A table with two columns, 'CAMPO' and 'DATO', is displayed. The 'CAMPO' column contains 'Nombre del Área (\*)', 'Una descripción del Área', and 'Nombre de su departamento (\*)'. The 'DATO' column contains a text input field, another text input field, and a dropdown menu with the text 'Seleccionar opción'. Below the table are three buttons: 'Limpiar campos', 'Guardar datos', and 'Actualizar'. At the bottom, there is a link 'Muestra el Formulario para dar de Alta un Departamento' and three icons: 'Ayuda', a printer icon, and a refresh icon.

Si el departamento no se encuentra en la lista de clic en la liga que está en la parte inferior de esta pantalla para dar de alta un nuevo departamento.

En caso de querer dar de alta información en cualquier otra tabla de la base de datos se proporciona una liga llamada **“Mostrar otros formularios”**, que como su nombre lo dice muestra una lista de todos los formularios que componen el sistema. La pantalla que se muestran son las siguientes:



### Opción Alta de un Administrador

Para dar de alta un administrador seleccione esta opción de la lista y a continuación aparecerá el siguiente formato:



La información debe ser proporcionada como se indica a continuación:

Nombre de la casilla	Información de la casilla
RFC	Digitarlo sin el guión entre las letras y los números. Por ejemplo REAX780213.
Login	Es una palabra que identifica al usuario.
Password	Es una clave proporcionada por el administrador para acceder al sistema.
Nombre Completo	Escribirlo en mayúsculas y minúsculas.
Descripción	Escribirla en mayúsculas y minúsculas.

### Opción Alta de una carrera

En este formulario podrá dar de alta la carrera a la que pertenezca un alumno y que por alguna razón aún no se encuentra en la base de datos.

ADMINISTRADOR

**DIMEL**

La tabla a mostrar es Carrera

Faltan datos en los campos  
Se Insertan valores en la tabla de carreras

CAMPO	DATO
Clave de la carrera (*)	<input type="text"/>
Nombre de la carrera (*)	<input type="text"/>
Plantel al que pertenece la carrera (*)	Seleccionar opción

Limpiar campos   Guardar datos   Actualizar

[Muestra el Formulario para dar de Alta un Plantel](#)

Ayuda

Para dar de alta una carrera proporcione la siguiente información:

Nombre de la casilla	Información de la casilla
Clave	Utilizar de 1 a 4 dígitos para definirla.
Nombre	Escribir en mayúsculas y minúsculas.
Plantel al que pertenece	Seleccionar uno de la lista.

En caso de que el plantel al que pertenece la carrera no se encuentra en la lista, en la parte inferior del formulario, encontrara una liga que proporciona el formulario para darlo de alta el plantel.

Una vez que ya es reconocida la información el sistema muestra una pantalla de de aviso al usuario.

Alta de plantel de forma correcta	
Clave de la casilla (*)	20
Nombre de la casilla (*)	Persepolis
Plantel al que pertenece la casilla (*)	Facultad De Ciencias

### Opción Alta de plantel

Para dar de alta un plantel se proporciona el siguiente formulario:

CAMPO	DATO
Clase (*)	
Nombre del Plantel (*)	

Limpiar campos   Guardar datos   Actualizar

Muestra el Formulario para ver los Plantel

Ayuda

La información que se debe proporcionar al sistema tiene que tener las siguientes condiciones:

Nombre de la casilla	Información de la casilla
Clave del Plantel	La proporcionado por le alumno.
Nombre del Plantel	Escribir en mayúsculas y minúsculas.

Para que pueda hacer una consulta de las claves de los demás planteles, en la parte de debajo de la pantalla se encuentra una liga que muestra una consulta de los planteles y sus respectivas claves.

### Opción Alta de una nacionalidad

Si desea llevar acabo un alta de una nacionalidad de clic en está opción y aparecerá la siguiente pantalla:



Proporcione la siguiente información:

Nombre de la casilla	Información de la casilla
Clave	Coloque el número subsiguiente al que se muestra en la tabla.
Nombre	Escribir en mayúsculas y minúsculas.

Para saber cual es la clave que le corresponde a la siguiente nación, en la parte inferior del formulario se encuentra una liga, la cual al dar clic muestra el nombre de las nacionalidades y sus correspondientes claves. Para una nacionalidad nueva escriba el nombre y como clave el número siguiente al último de la lista.

### Opción Alta de un nombramiento

Para dar de alta un nombramiento tiene que proporcionar la información de la forma en la que se define a continuación:

Nombre de la casilla	Información de la casilla
Clave	Coloque el número subsiguiente al que se muestra en la tabla.
Nombre	Escribir en mayúsculas y minúsculas.

Para ver el número subsiguiente de clave, en la parte inferior del formulario se muestra una liga que hace una consulta a la tabla de nombramientos. En esta tabla se muestran las claves y los nombramientos existentes hasta el momento.

The screenshot shows the 'ADMINISTRADOR' interface for DIMEL. It features a header with the DIMEL logo and two crests. The main content area displays a message: 'La tabla a mostrar es Tipo de nombramiento' and 'Faltan datos en los campos Se Insertan valores en la tabla de nombramientos'. Below this is a table with two columns: 'CAMPO' and 'DATO'. The table contains two rows: 'Clave (\*)' and 'Nombramiento (\*)', each with an empty input field. At the bottom of the form are three buttons: 'Limpiar campos', 'Guardar datos', and 'Actualizar'. A footer note says 'De Clic para ver la clave subsiguiente' and there are icons for 'Ayuda', a back arrow, and a home icon.

## Opción Alta de Tipos de Profesor

Para dar de alta un tipo de profesor tiene que proporcionar la información de la forma en la que se define a continuación:

Nombre	Información de la casilla
Clave	De clic en el hipervínculo para ver las claves y coloque la subsiguiente.
Tipo de profesor	Escribir en mayúsculas y minúsculas.
Descripción	Escribir en mayúsculas y minúsculas.

Para ver el número subsiguiente de clave, en la parte inferior del formulario se muestra una liga que hace una consulta a la tabla de tipos de profesor. En esta tabla se muestran las claves y los tipos de profesor existentes hasta el momento.

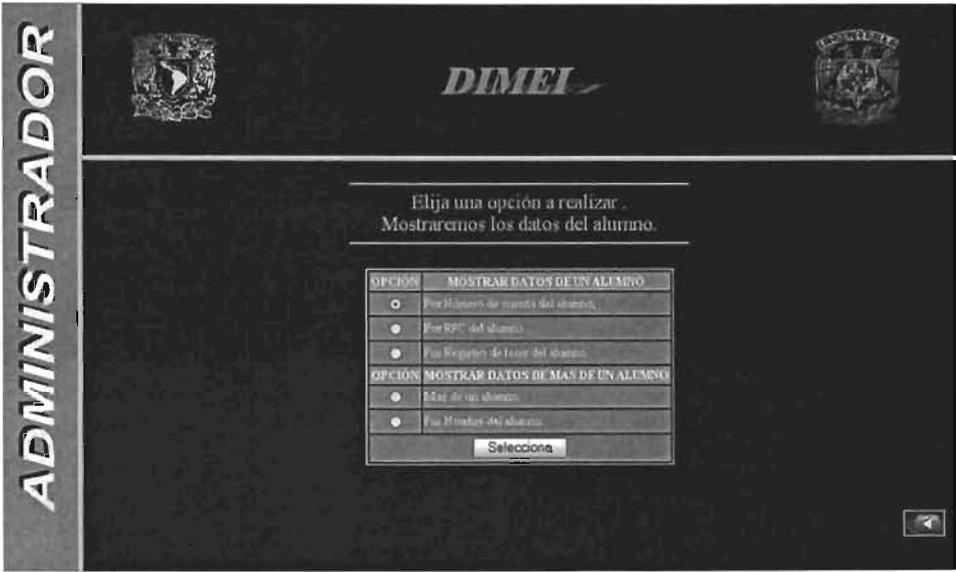
## Opción Mostrar

Esta opción nos permite seleccionar entre varias maneras de consultar información a la base de datos. Dentro de la información más importante que se puede consultar, es la información referente a los alumnos, profesores y proyectos. A continuación explicamos cada una de las opciones en las que se encuentra dividida esta opción.



## Opción Mostrar Alumnos

Esta opción nos permite ver la información del alumno accediendo de diferentes maneras entre las que podemos encontrar la opción de por número de cuenta del alumno, por RFC del alumno, por registro de tesis del alumno o podemos encontrar la opción de mostrar información de más de un alumno. Para esto contamos con la siguiente pantalla:



The screenshot shows a web interface for an administrator. On the left, there is a vertical sidebar with the word 'ADMINISTRADOR' in white capital letters. The main content area has a dark background with a white header containing two logos and the text 'DIMEI'. Below the header, there is a prompt: 'Elija una opción a realizar. Mostraremos los datos del alumno.' This is followed by a table of options:

OPCIÓN	MOstrar DATOS DE UN ALUMNO
<input type="radio"/>	Por Número de cuenta del alumno.
<input type="radio"/>	Por RFC del alumno.
<input type="radio"/>	Por Registro de Tesis del alumno.
OPCIÓN	MOstrar DATOS DE MAS DE UN ALUMNO
<input type="radio"/>	Más de un alumno.
<input type="radio"/>	Por Muestra de alumnos.

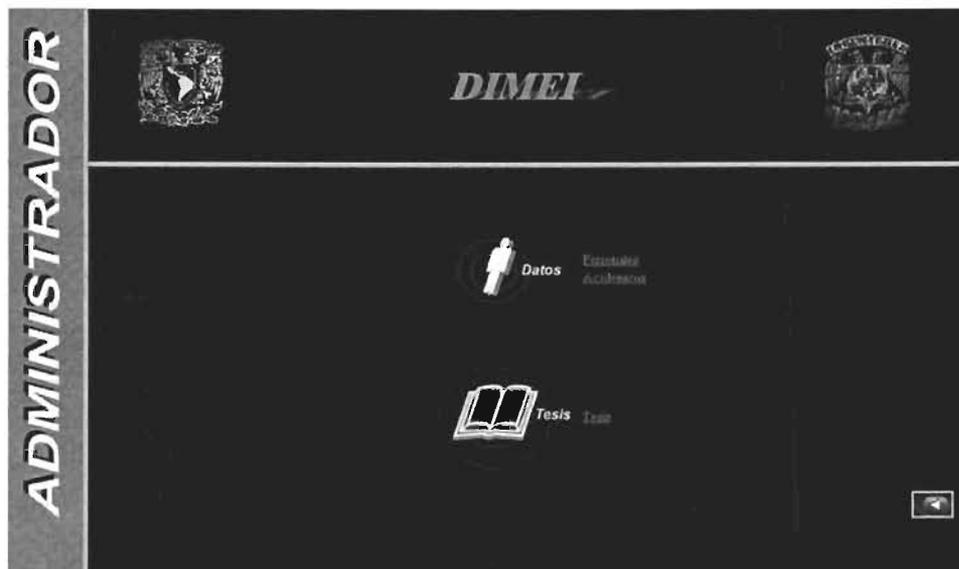
Below the table is a button labeled 'Selección'. In the bottom right corner of the interface, there is a small square button with a left-pointing arrow.

### Opción mostrar por número de cuenta del alumno

Nos muestra un formulario en el que es necesario colocar el número de cuenta del alumno, omitiendo el guión entre número y colocando un cero al principio si la generación es anterior al 2000.

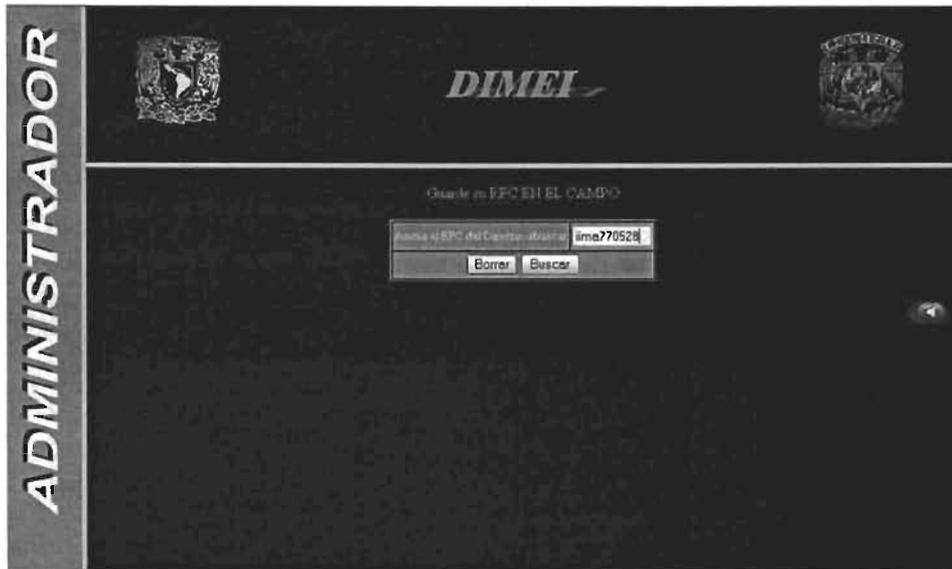


Después de dar clic en el botón enviar, el sistema enviará una pantalla en la que se podrán consultar los datos personales, académicos y de proyecto de tesis del alumno. Para esto de clic en las opciones que se presentan. Para mayor información vea de la página 152 a la 154.

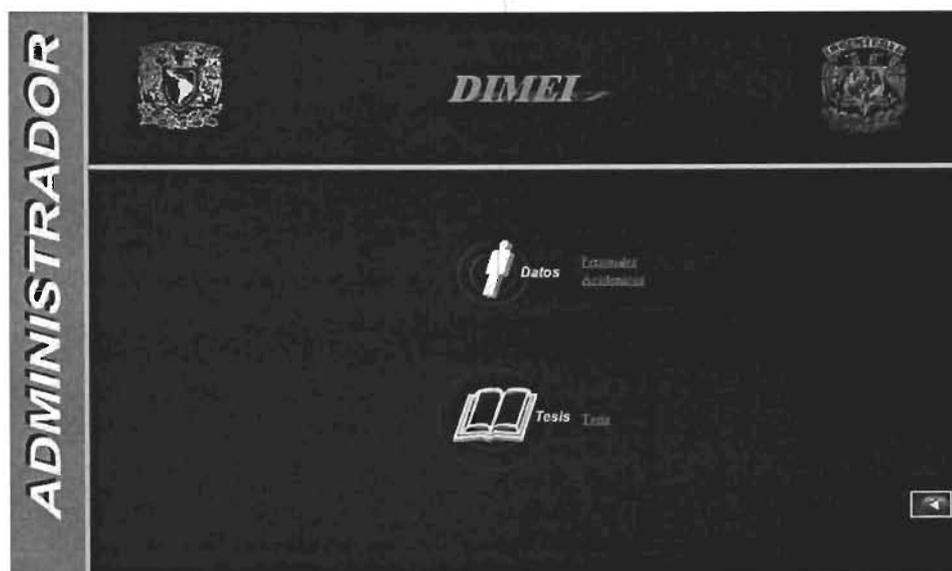


### Opción mostrar por RFC del alumno

Esta opción nos permite consultar la información del alumno proporcionando su RFC, éste debe ser proporcionado sin guiones entre las letras y los números. Para esto se presenta una pantalla como la siguiente:

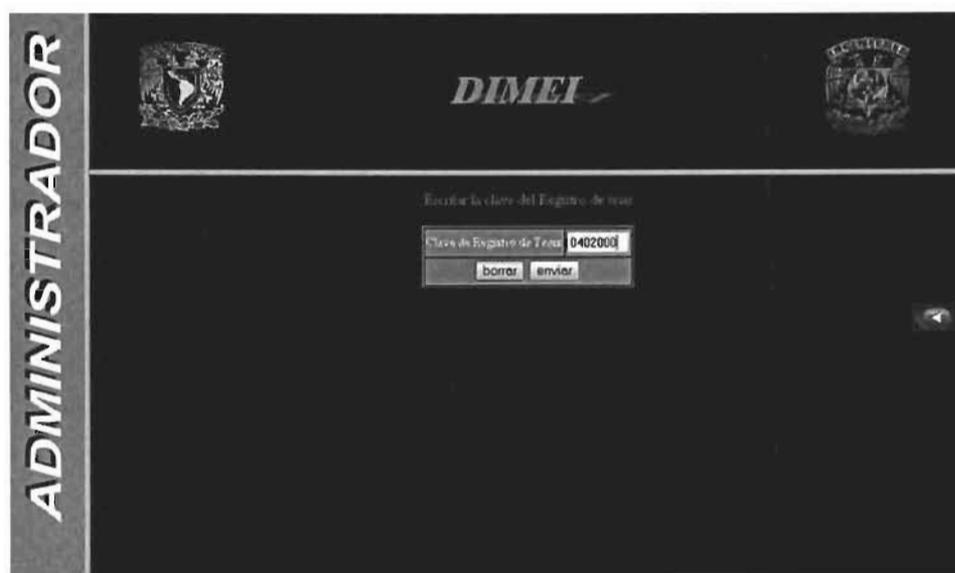


Una vez proporcionada esta información de clic en el botón de buscar, y si esta información está bien proporcionada el sistema enviará una pantalla con toda la información.



### Opción mostrar por el número de registro de la tesis

Está es una opción que nos permite consultar la información del alumno accediendo a través de el número de registro de la tesis que este desarrollando.



Una vez enviada esta información el sistema manda la información de los alumnos que estén colaborando en el desarrollo de esa tesis.

### Opción más de un alumno

Está opción nos muestra una consulta de todos los alumnos registrados en la base de datos.

**ADMINISTRADOR**

**DIMEI**

En construcción  
Se consulta la tabla alumnos

num. Cuenta	ID Alumno	nombre alumno	apellido P	apellido M	sexo	fecha	numero	Ext	numero	Int	colegio	codigo postal	email
99304212	01MA770238	Angel	Isidro	Mercado	Hombre	Cast	17	0	0	0	Las Cruces	11227	angel.isidro.mercado@univ.edu.ar
0048129	02A3760129	Marcos Osorio	Reyes	Agarce	Mujer	Avil	15	0	0	0	San Lorenzo	1125	marcososorio@univ.edu.ar
00417348	02B3771118	Ignacio Ramirez	Mendoza	Geymes	Hombre	Est	13	0	0	0	Ampliacion Santa Marta Avellan	3505	ignaciorm@univ.edu.ar

[Volver](#)

### Opción mostrar por el nombre del alumno

Para poder buscar la información del alumno por medio de su nombre, se presenta una pantalla con tres campos en la que deberá ser proporcionado el apellido paterno, materno y nombre del alumno. De clic en el botón buscar y a continuación se mandará la pantalla donde podrá consultar los datos.

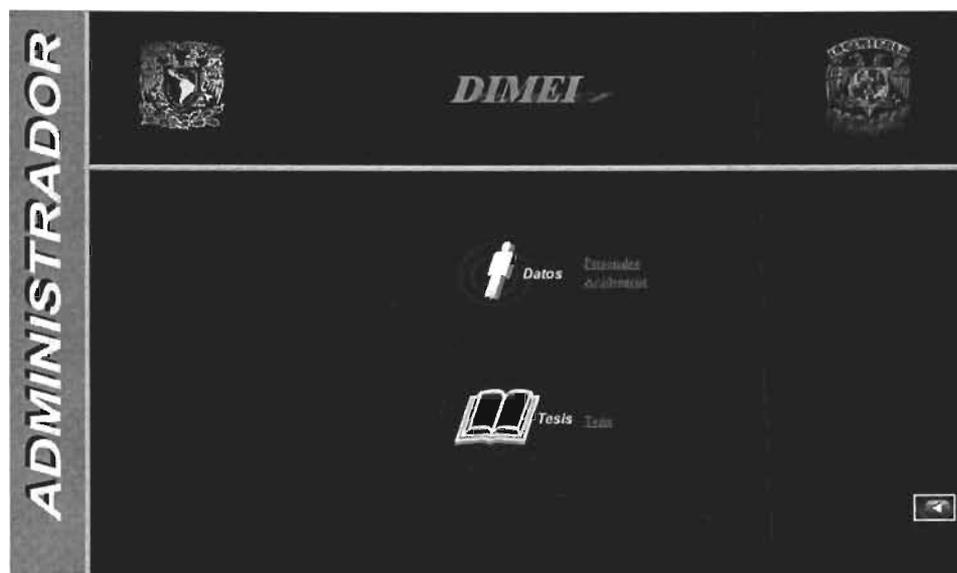
**ADMINISTRADOR**

**DIMEI**

En construcción  
Proporcionar los Alumnos solicitado

Nombre Alumno	Angel
Apellido Paterno	Isidro
Apellido Materno	Mercado
<input type="button" value="Busca Alumno"/>	<input type="button" value="Borrar texto"/>

[Ayuda](#)  



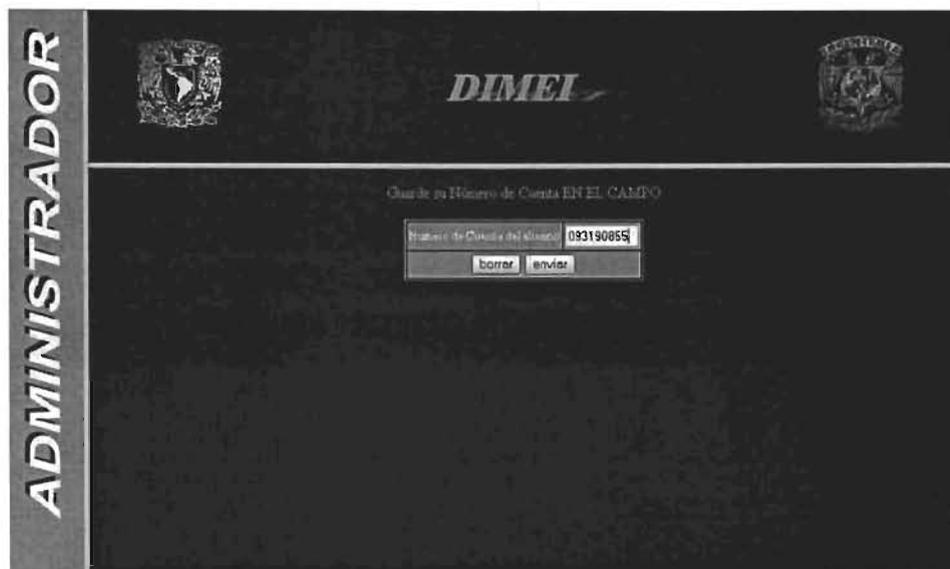
### Opción Mostrar Profesores

Esta opción nos permite ver la información de los profesores accediendo de diferentes maneras entre las que podemos encontrar la por número de cuenta del alumno al que le dirige su tesis, por RFC, por registro de tesis dirigida o podemos encontrar la opción de mostrar información más de un profesor y por el nombre del profesor. Para esto contamos con la siguiente pantalla:



### Opción mostrar por número de cuenta de un alumno al que dirigió tesis

Nos muestra un formulario en el que es necesario colocar el número de cuenta del alumno, omitiendo el guión entre número y colocando un cero al principio si la generación es anterior al 2000.



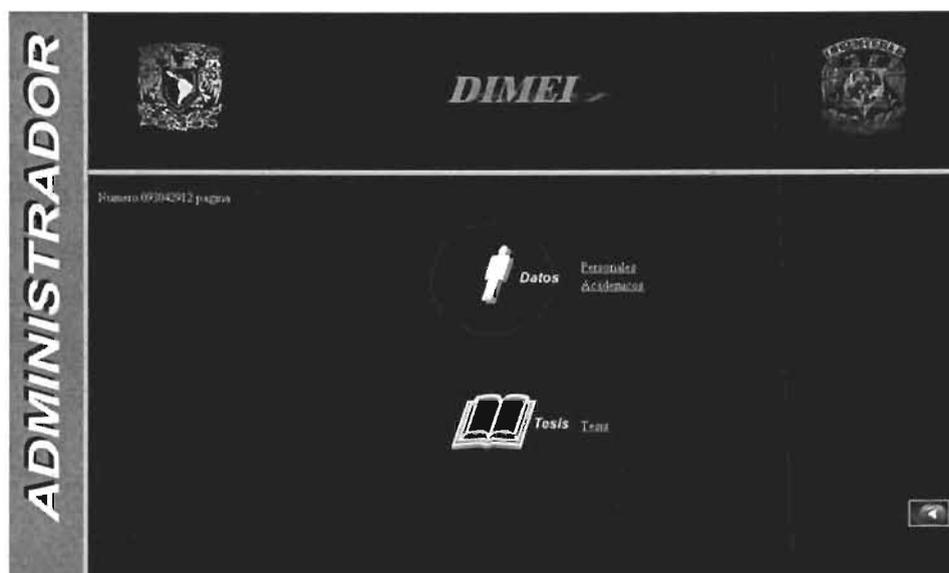
The screenshot displays the DIMEL administrator interface. On the left, a vertical sidebar contains the word "ADMINISTRADOR". The main content area features the DIMEL logo at the top center, flanked by two institutional crests. Below the logo, the text "Clave de su Número de Cuenta EN EL CAMPO" is visible. A search form is present with a text input field labeled "Número de Cuenta del Alumno" containing the value "083190885". Below the input field are two buttons: "borrar" and "enviar".

### Opción por el RFC del profesor

Esta opción nos permite consultar la información del profesor proporcionando su RFC, éste debe ser proporcionado sin guiones entre las letras y los números. Para esto se presenta una pantalla como la siguiente:



Una vez proporcionada está información de clic en el botón de buscar, y si esta información está bien proporcionada el sistema enviará una pantalla con tres opciones distintas, que son para ver los datos personales, académicos o los de sus tesis dirigidas.

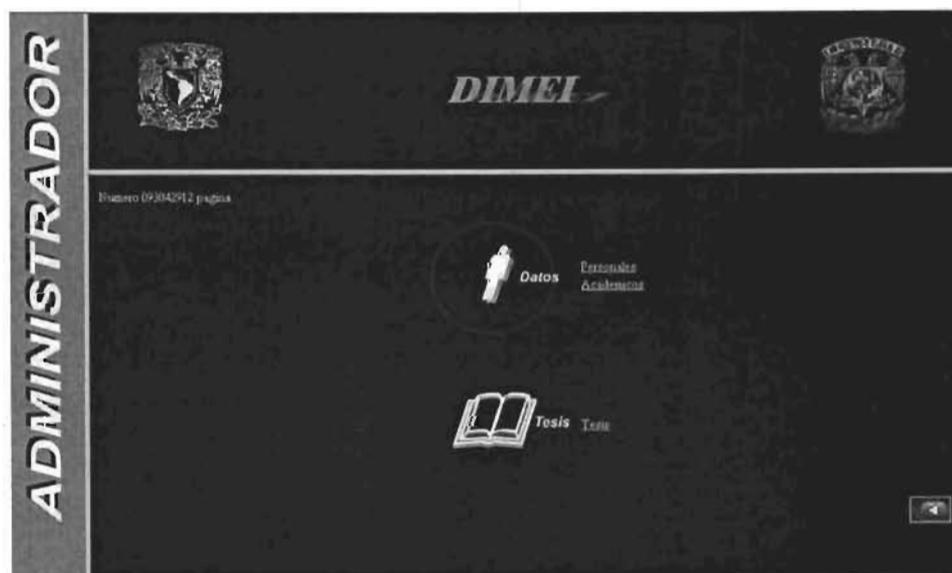


### Opción mostrar por registro de una tesis dirigida

Con esta opción se pueden conocer los datos de un profesor ingresando el número de una registro de una tesis dirigida por él.

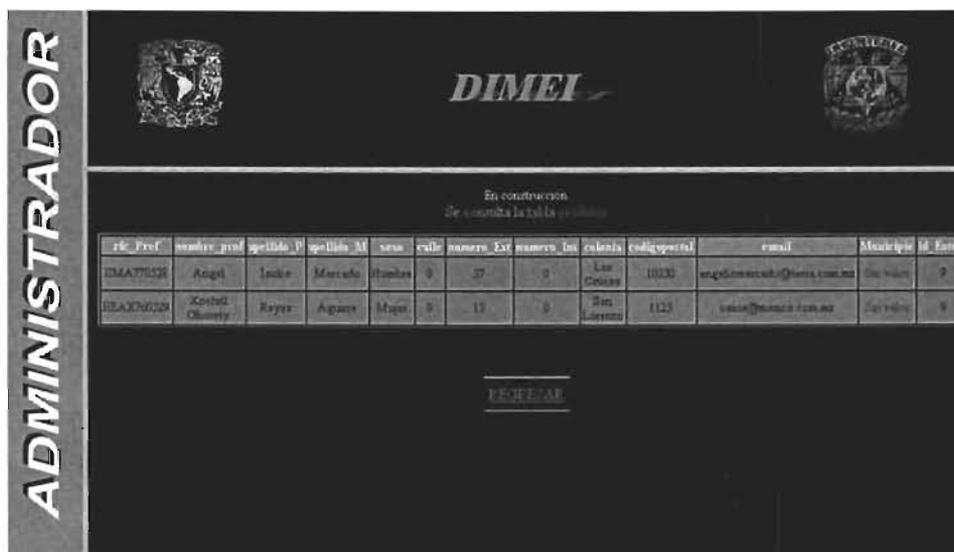


Posteriormente se mandará la pantalla para consultar los datos.



### Opción mostrar más de un profesor

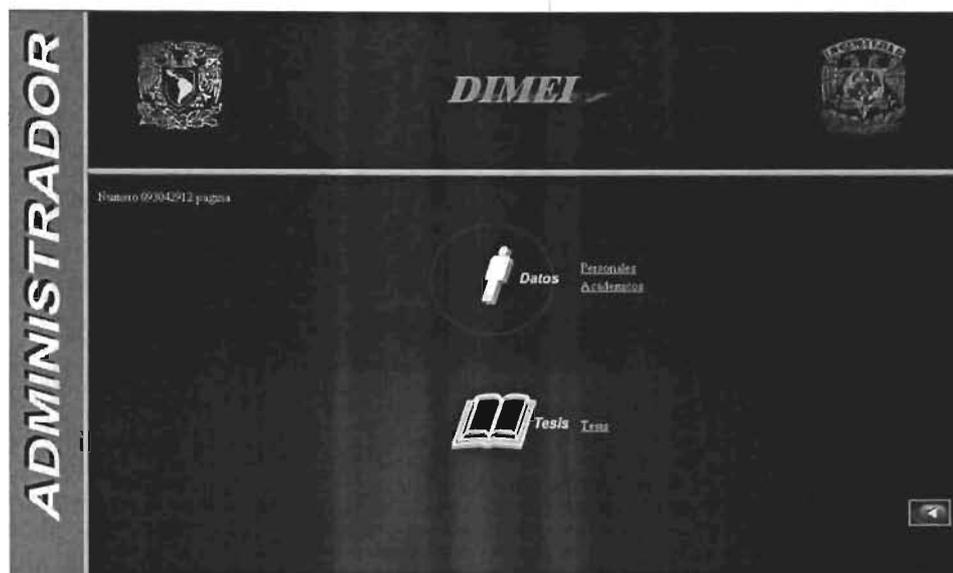
Esta opción permite hacer una consulta de todos los profesores registrados por el sistema.



### Opción por el nombre del profesor

Para poder buscar la información del profesor por medio de su nombre, se presenta una pantalla con tres campos en la que deberá ser proporcionado el apellido paterno, materno y nombre. De clic en el botón buscar y a continuación se mandará la pantalla donde podrá consultar los datos.





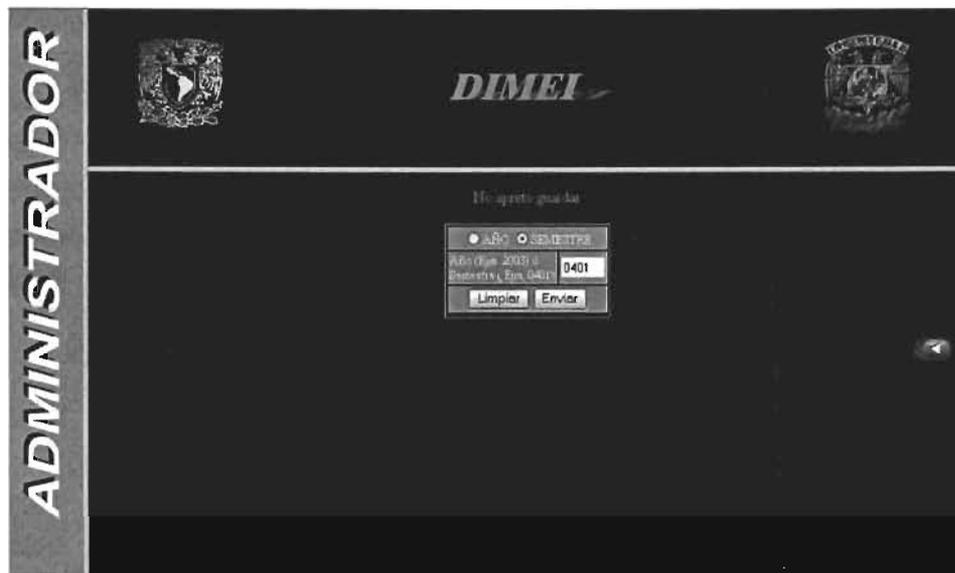
### Opción mostrar proyectos de Tesis

Opción nos permite ver la información de los proyectos accediendo de diferentes maneras entre las que podemos encontrar la opción de por año y semestre, RFC del director, número de cuenta del alumno, departamento o área de estudio, licenciatura o posgrado y por el número de registro. Para esto contamos con la siguiente pantalla:



### Opción mostrar por año o semestre

Aquí el sistema muestra una pantalla en la que podemos buscar una tesis por año o semestre. En caso de seleccionar ver por año, deberá ingresar esté con sus cuatro dígitos, si por el contrario decide ver las tesis por semestre, esté deberá ser ingresado componiéndolo del año y del semestre, por ejemplo si desea ver las tesis del 2004 pero del segundo semestre año escolar, esté que daría de la siguiente manera: 0402.



The screenshot shows the DIMEL administrator interface. On the left, there is a vertical sidebar with the word "ADMINISTRADOR" written vertically. The main content area has a dark background with the DIMEL logo at the top center. Below the logo, there is a search form with the following elements:

- A header "He escrito que la" followed by two radio buttons: "AÑO" (selected) and "SEMESTRE".
- A text input field containing "0401".
- Below the input field, there is a label "Año (del 2003) a" and "Semestre, Fin. 0401".
- At the bottom of the form are two buttons: "Limpiar" and "Enviar".

Una vez hecho esto, el sistema proporciona una lista de todas las tesis pertenecientes al año y al semestre ingresado y para mayor información seleccionar cualquiera de ellas y pedir al sistema que mande información completa.

### Opción por el RFC del profesor

Esta opción nos permite consultar la información de las tesis dirigidas por un profesor en particular proporcionando su RFC, el debe ser proporcionado sin guiones entre las letras y los números. Para esto se presenta una pantalla como la siguiente:



Posteriormente se muestra una pantalla donde se puede encontrar una lista de las tesis dirigidas por el profesor, puede seleccionar cualquiera de ellas para ver más información.



**ADMINISTRADOR**

**DIMEI**

Mostrar datos de la tesis 0702001

Número de Registro: 0702001	Proyecto de Tesis: 0702001	Nivel: Pregrado	Proyecto: 0702001	Año: 2001	Estado: 01
Título de la Tesis: Tesis de una propuesta					
Descripción: Descripción de la propuesta					
Forma de pago a crédito: 0702001					
Estatus de la Propuesta: 0702001					
Estatus de abono: 0702001					

Datos del director de tesis

Nombre: Abad Lopez Marcelo	Departamento: Departamento de Ingenierías Mecánicas	Código de identificación del usuario: 0702001	Estado: 01
----------------------------	---	---	------------

Datos de los alumnos registrados en la tesis

Número de cuenta: 093190865	Nombre: Carlos Daniel Pareda	Código de identificación del usuario: 0702001	Estatus del alumno: 01
Nombre: Carlos Daniel Pareda	Institución: Facultad de Ingeniería	Año de ingreso: 2001	Año de egreso: 2001
Número de cuenta: 093190865	Nombre: Carlos Daniel Pareda	Código de identificación del usuario: 0702001	Estatus del alumno: 01
Nombre: Carlos Daniel Pareda	Institución: Facultad de Ingeniería	Año de ingreso: 2001	Año de egreso: 2001

### Opción mostrar por el número de cuenta del alumno

En esta opción se puede consultar la información de un proyecto de tesis proporcionando el número de cuenta, éste debe ser proporcionado sin guiones entre los números. Para esto se presenta una pantalla como la siguiente:

**ADMINISTRADOR**

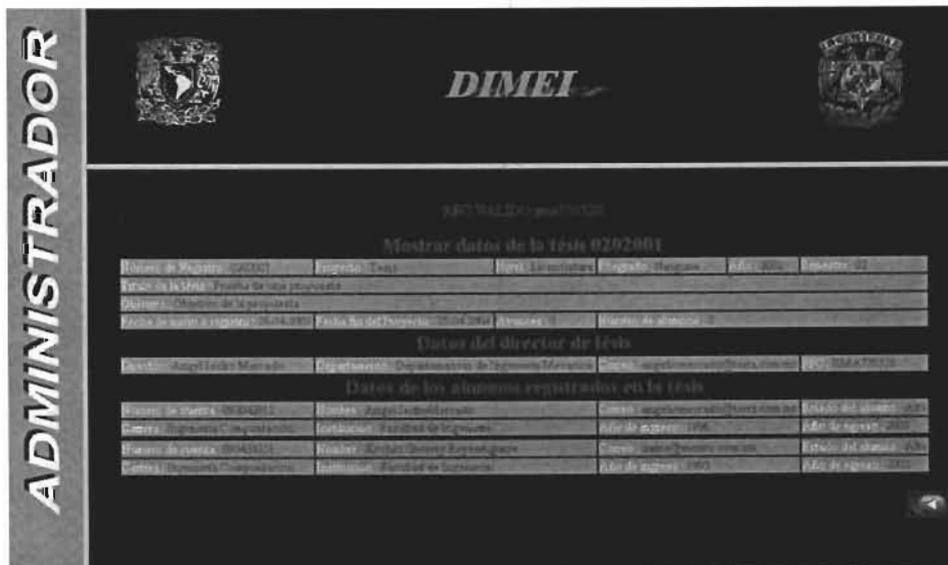
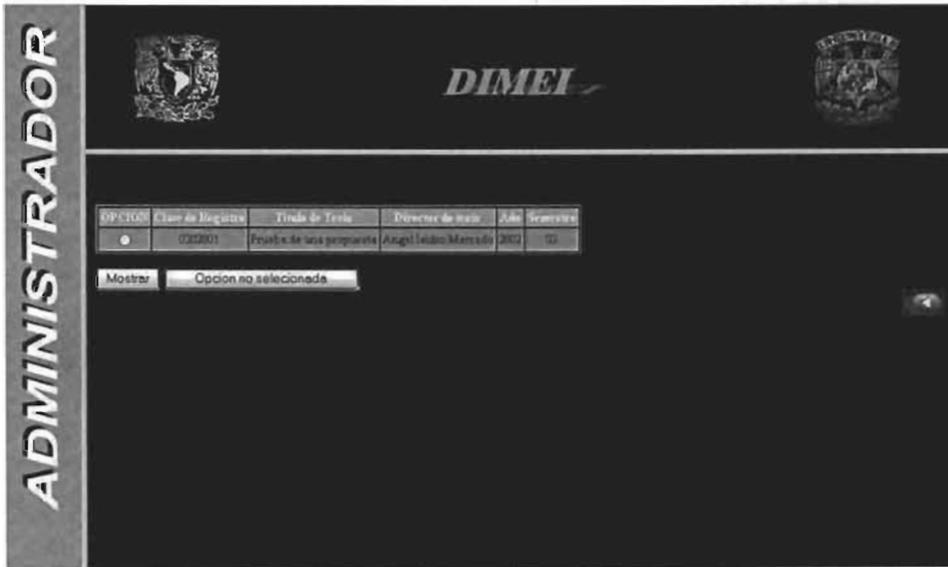
**DIMEI**

Quede el Número de Cuenta EN EL CAMPO

Número de Cuenta del alumno: 093190865

borrar enviar

Posterior mente se muestra una lista que contiene la(s) tesis que tiene el alumno y que puede ir desde licenciatura, maestría y doctorado. La lista se presenta en el siguiente formato y si el usuario desea mayor información sólo deberá seleccionar una opción y dar clic el botón mostrar para que nos presente una pantalla con la información completa.

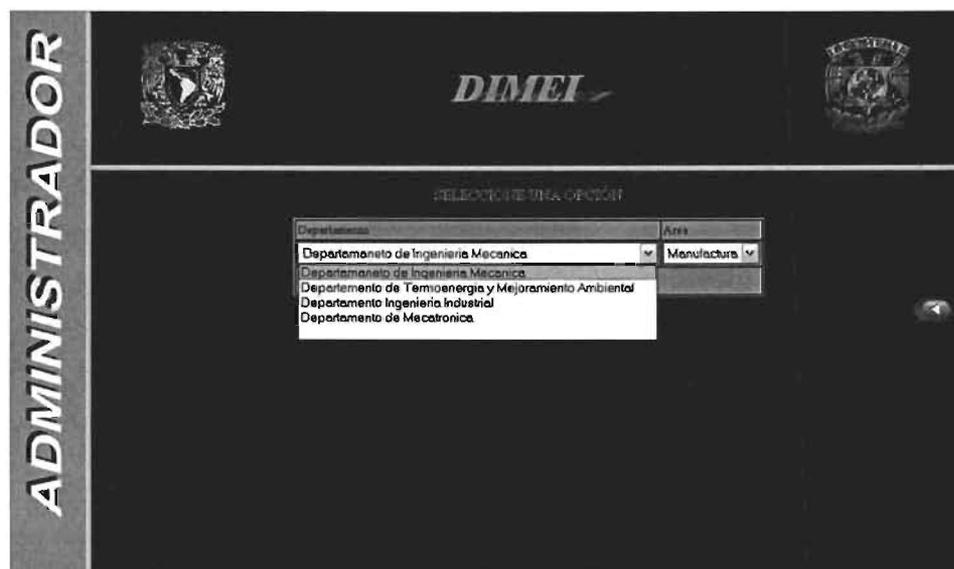


### Opción mostrar por departamento o área de estudio

El sistema muestra una pantalla con tres opciones a seleccionar, mostrar las tesis por departamento, por área de estudio o por ambos.



Después de seleccionar aquella opción deseada aparecerá una pantalla que contiene listas desplegables para que se pueda seleccionar el departamento o área. Por ejemplo la pantalla que a continuación se muestra pertenece a la opción de mostrar por departamento y área de estudio.



Si existen tesis registradas para el departamento y el área, entonces se mandará una pantalla con la lista de tesis y se podrá escoger cualquiera de ellas para revisar mejor la información.

### Opción mostrar por licenciatura o posgrado

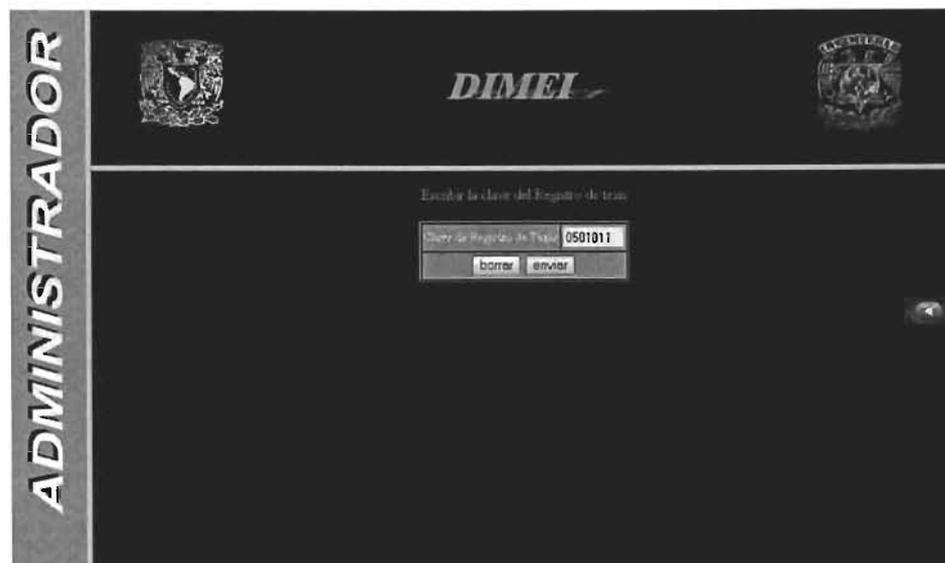
Permite consultar las tesis dadas de alta en licenciatura o posgrado, para seleccionar está opción muestra un formulario. Ya seleccionada alguna de clic en el botón mostrar para que el sistema mande una lista de las tesis registradas y si se desea ver mayor información de alguna en particular de clic en el botón mostrar para que envíe todos lo datos almacenados.



The screenshot shows the DIMEL administrator interface. On the left is a vertical sidebar with the word "ADMINISTRADOR" in white text. The main content area has a dark background with the DIMEL logo and two university crests at the top. Below the header, there is a section titled "Selecciona el Nivel" (Select the Level). This section contains two dropdown menus: "Nivel" (Level) with "Licenciatura" (Bachelor's) selected, and "Posgrado" (Postgraduate) with "Ninguna" (None) selected. Below these menus is a "Buscar" (Search) button.

### Opción mostrar por registro

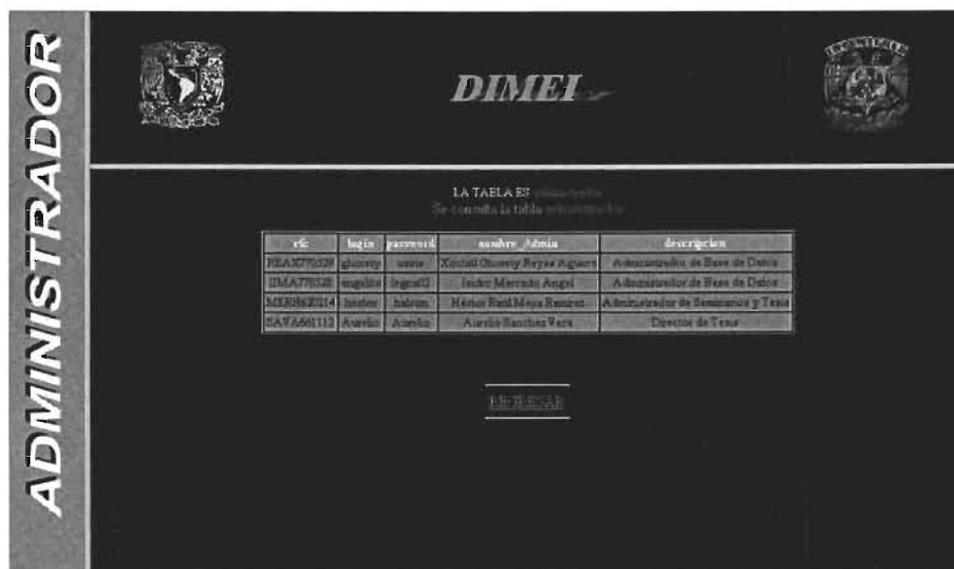
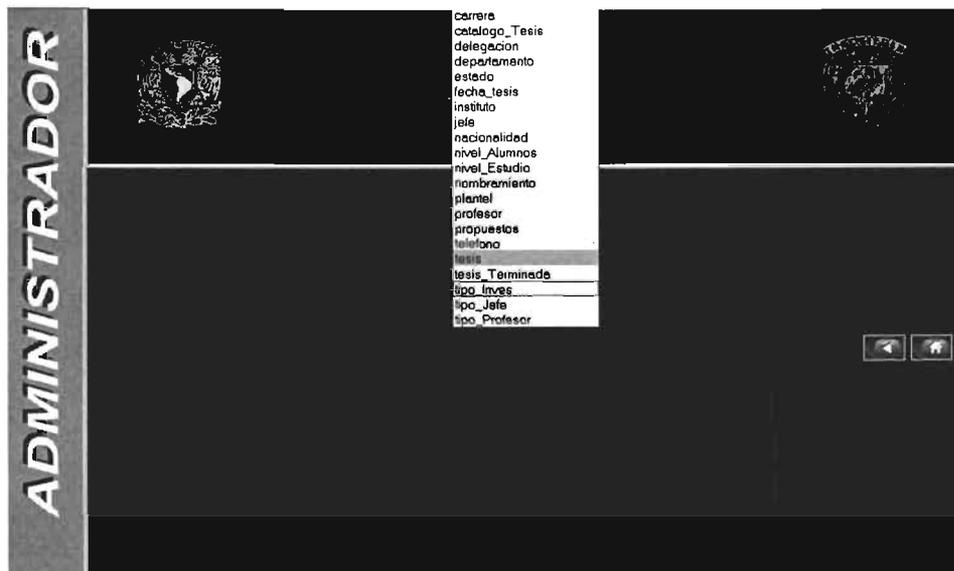
El sistema muestra una pantalla en la que podemos buscar una tesis por su registro. Coloque esté en la casilla y de clic en el botón buscar para que le muestre la información.

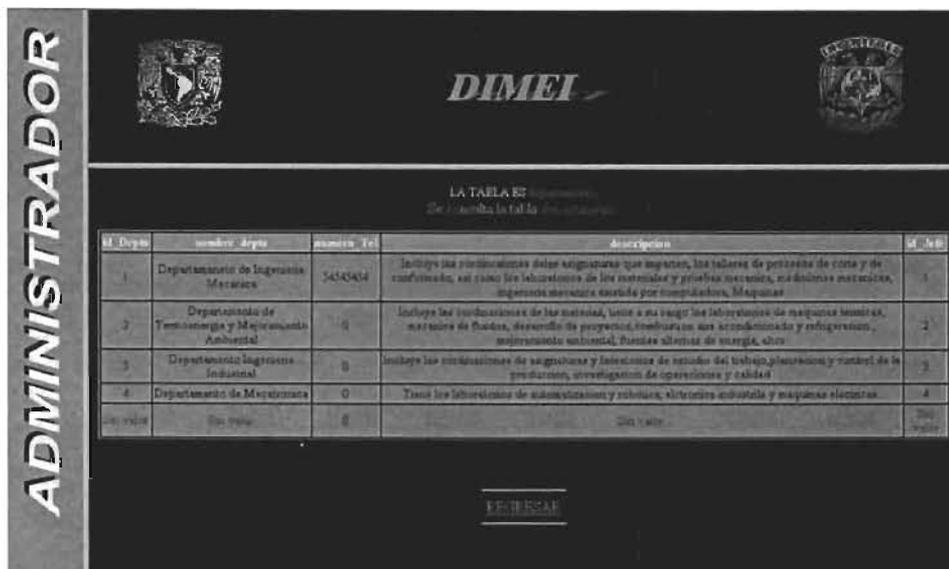


The screenshot shows the DIMEL administrator interface. On the left is a vertical sidebar with the word "ADMINISTRADOR" in white text. The main content area has a dark background with the DIMEL logo and two university crests at the top. Below the header, there is a section titled "Escriba la clave del Registro de tesis" (Enter the thesis registration key). This section contains a text input field with the value "0501011" and two buttons: "borrar" (clear) and "enviar" (send).

**Opción mostrar otro**

Está opción muestra una pantalla en la que encontramos una lista con los nombres de todas las tablas que conforman la base de datos, de dicha lista podemos escoger cualquiera de la que desee obtener mayor información. El sistema mostrara una consulta general de la información almacenada. Por ejemplo veamos las pantallas que genera el sistema después de seleccionar la opción de administradores y departamentos.



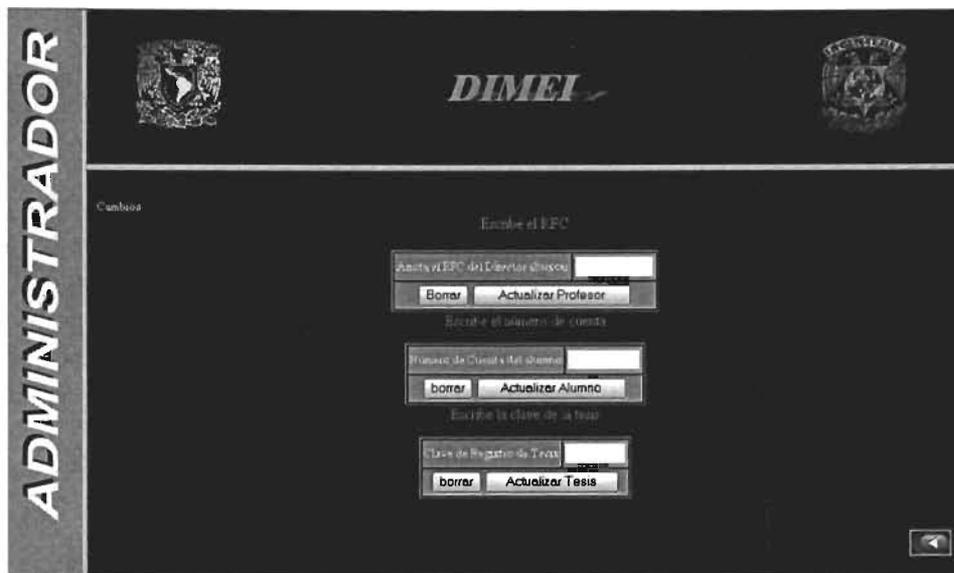


**Opción Cambio**

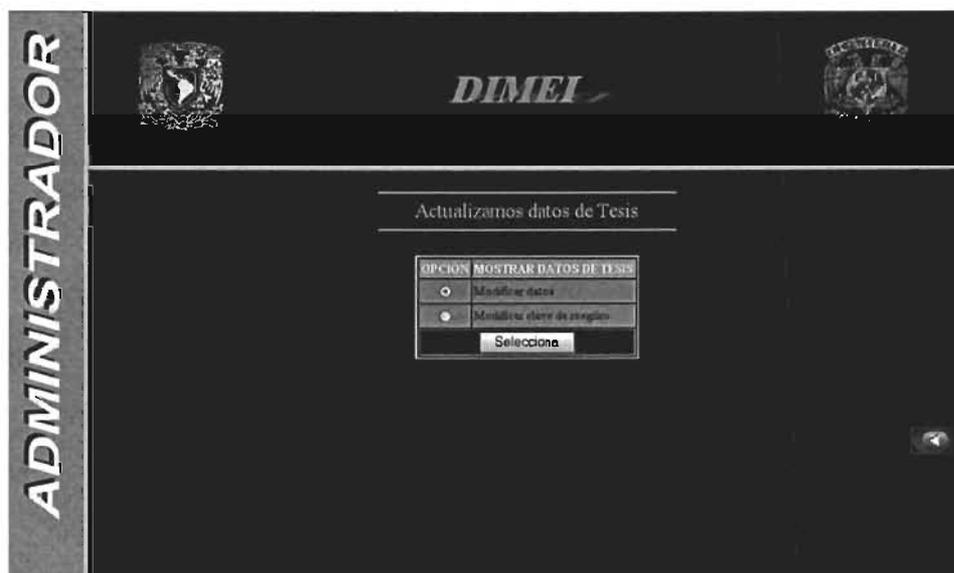
Permite poder modificar la información almacenada. La información que puede ser modificable corresponde a los alumnos, profesores, proyectos de tesis, jefes de departamento, departamentos y áreas. Para poder modificar la información se presenta una pantalla en la cual deberá proporcionarse la información que el sistema solicite.

En la siguiente pantalla se muestran tres opciones, que corresponde al cambio de información en alumnos, profesores y proyectos de tesis. Para hacer un cambio en la información del alumno proporcione el número de cuenta, para el profesor proporcione el RFC y para el proyecto de tesis proporcione el registro. Es importante recordar que solo podrá hacer cambios en uno a la vez. Por ejemplo hagámoslo para un proyecto de tesis.

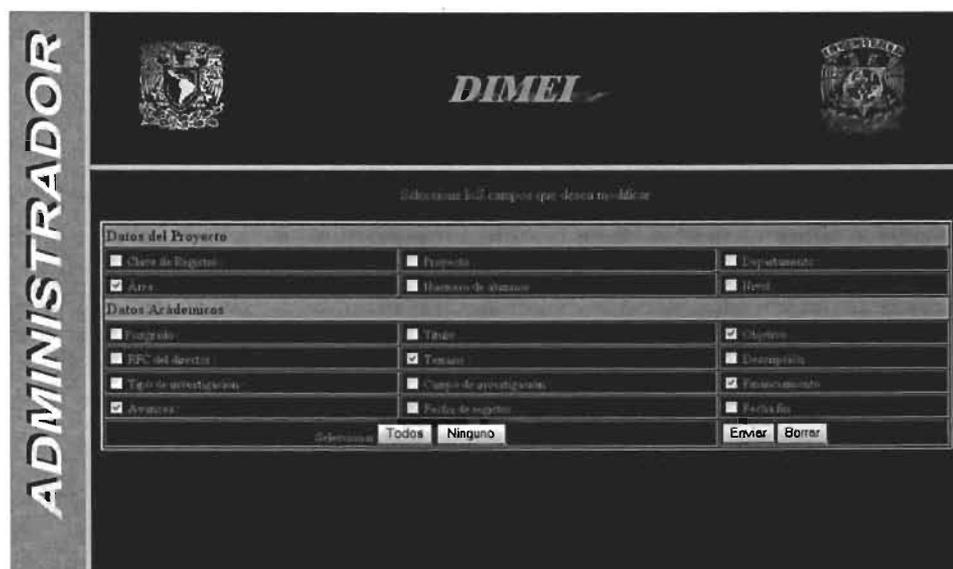
Lo primero que tenemos que hacer es anotar el número de registro de la tesis.



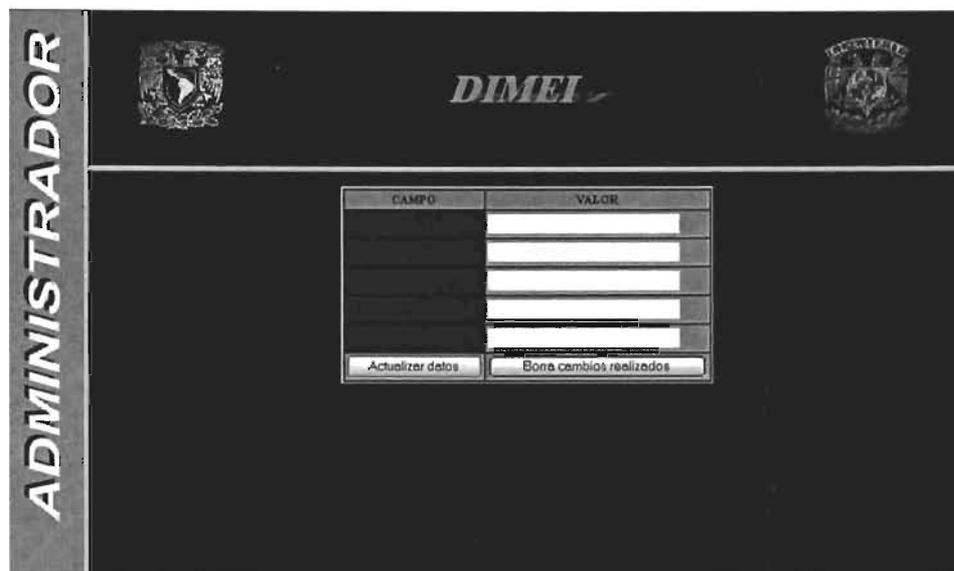
Posterior a esto aparecerá una pantalla que contiene dos opciones una de ellas para modificar los datos y otra para modificar el registro de tesis, selección de acuerdo a los cambios que desea llevar a cabo.



Después de dar clic en el botón seleccionar aparecerá una pantalla con una flecha a la cual debe dar clic, para que el sistema mande una pantalla que contiene un formulario que le permitirá seleccionar los campos que desea modificar.



Una vez seleccionados los campos de clic en el botón enviar, para que el sistema muestre el formulario con los campos a modificar.



### Opción Bajas

En la opción de bajas lo único que permite es cambiar el estado de la información en un campo donde el dato es un alta o baja. La información no puede ser retirada de la base de datos, por que en base a ella se pueden hacer análisis de tipo estadístico.

Para poder llevar acabo este proceso de baja se proporciona un formulario, en el que se debe ingresar ya sea el número de cuenta, RFC, registro de tesis, dependiendo de cual de ellos desea que pase al estado de baja. Una vez ingresado el dato se proporciona una pantalla en la que hay dos botones uno de ellos para dar de alta y otro para dar de baja. En ese momento aparece el campo que muestra el letrero baja o alta para indicar ese estado en la información.



**ADMINISTRADOR**



**DIMEI**



El alumno se puede dar de baja de su registro seleccionando en opciones.

OPCION	Clase	Número de cuenta	Nombre del alumno	Clave de la sede	Título de la sede	Nivel de estudios	Estado del Alumno
0	1	00042912	Angel Indio Mercado	020001	Facultad de las propuestas	Licenciatura	Alta



**ADMINISTRADOR**



**DIMEI**



El alumno se ha dado de Baja de esta tesis.

tipo valor

OPCION	Número de cuenta	Nombre del alumno	Clave de la sede	Título de la sede	Nivel de estudios	Estado del Alumno
1	00042912	Angel Indio Mercado	020001	Facultad de una propuesta	Licenciatura	Baja

