



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

---

---

**Facultad de Ingeniería**

**SISTEMA DE INFORMACIÓN CURRICULAR  
PARA LA FACULTAD DE INGENIERÍA (SICFI)**

**T E S I S**

PARA OBTENER EL TÍTULO DE:

**INGENIERO EN COMPUTACIÓN**

PRESENTAN:

**GUERRERO CONTRERAS LUIS FELIPE**

**MEDRANO ARELLANO HÉCTOR**

**MELCHOR OJEDA HERLINDA**



**ASESOR: M. I. ABIGAIL SERRALDE RUÍZ**

**México, D. F.**

**2007**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

**A mis padres** Con infinito agradecimiento por que a ustedes debo lo que soy, el que siempre hayan estado conmigo en las buenas y en las malas, por eso, esta meta también es de ustedes, es producto de su amor, dedicación, esfuerzos y enseñanzas, gracias por darme esta invaluable herencia, que es el mejor regalo que me pueden dar, y que me dio la oportunidad de realizarme en la vida como profesionista.

Eternamente gracias.

**A mis hermanos** Por el apoyo y la gran paciencia que tuvieron a lo largo de mi carera, que supieron comprenderme como hermano y como amigo.

**A la Dra. Carmen Álvarez-Buylla Rocés y al M.C. Gerardo Ferrando Bravo** Por la confianza depositada en mi, para la realización de este proyecto.

**A mi familia** Por el todo el amor, cariño, confianza y apoyo que me han brindado siempre, ya que sin su ayuda este logro no habría sido posible, gracias a sus incontables palabras de aliento, a sus bromas y risas, gracias por compartir mis logros y tropiezos.

**A mis amigos** A todos aquellos que han sido son y serán parte de mi vida, que han reído y llorado conmigo, nombrarlos seria injusto, olvidarlos imperdonable, pero cada uno sabe lo que para mi significa llamarlos amigo.

**A Herlinda y Luis Felipe** Gracias por la amistad invaluable que me han brindado y por los momentos inolvidables que hemos pasado juntos en la realización de este trabajo.

**A Departamento de Información y Estadística de la Facultad de Ingeniería** y a cada uno de los compañeros que ahí tuve durante mi paso, ya que ahí fue donde se brindo el apoyo para que este proyecto se llevara acabo.

A todos ellos, ya a los que se me escapan de mi mente infinitamente Gracias.

**Héctor Medrano Arellano**

**A mis padres Esther y Agustín**, quienes me enseñaron a terminar siempre todo lo que comienzo y ello me sirve para seguir siempre adelante, formándome un carácter muy particular.

**A Alma**, la mejor persona que he conocido en mi vida y que considero mi mejor amiga, de quien aprendí a intentar las cosas sin pensar en el fracaso, viendo solo la oportunidad de mejorar.

**A mis hermanos José, Marí, Toño, Jesús, Luz, Gabriel, Alma y Ana**, quienes me apoyaron en todas mi decisiones y que siempre han estado a mi lado para brindarme su cariño.

**A esos chamacos**, Linda, Héctor, Jorge, Carlos, José Luis y todos aquellos que puedo considerar como verdaderos amigos y que siempre tuvieron tiempo para provocarme una sonrisa en esos momentos especiales.

A todos ellos les agradezco y comparto esto que es uno más de mis logros.

**Luis Felipe Guerrero Contreras**

**A mis padres y hermanos** por todo el amor y el apoyo brindado y porque hoy veo llegar al fin una de las metas de mi vida.

**A ti mama** quiero expresarte un profundo agradecimiento por tu forma de salir adelante lo cual me alentó a lograr esta realidad.

**A ti papa** por enseñarme que cuando se quiere algo se puede conseguir.

**A mis hermanos Mari, Silvia, Lupe y Ángel** por el apoyo que siempre me han dado y ese gran cariño y sobre todo por la confianza que tienen en mí.

**A ti Gerardo** por que eres el apoyo que me faltaba y porque te has ganado el cariño, admiración y respeto por tu atención y paciencia que tienes hacia mi.

**A mis amigos Héctor, Luis, Álvaro, Jorge, José Luis** que me han apoyado en estos últimos años y sin ustedes tampoco hubiese llegado a este momento tan importante.

**Y a la alegría de mi vida Angy, miguelito, Fer y Richy** que con su inocencia llenan de alegría.

A todos ellos les agradezco y comparto esto que es uno más de mis logros.

**Melchor Ojeda Herlinda**

# Contenido

<b>UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.....</b>	<b>1</b>
Facultad de Ingeniería .....	1
SISTEMA DE INFORMACIÓN CURRICULAR PARA LA FACULTAD DE INGENIERÍA (SICFI) .....	1
TESIS.....	1
<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Antecedentes.....	1
1.1.1 Consejo Técnico (CT).....	3
1.1.2 Dirección.....	3
1.1.2 Secretaría General.....	3
1.1.3 Secretaría de Servicios Académicos.....	5
1.1.4 Secretaría Administrativa.....	6
1.1.5 Divisiones.....	7
1.2 Problemática.....	9
1.2.1 Análisis de la Situación Actual.....	9
1.2.1 Problemática de la información curricular del personal académico.....	10
1.3 Descripción del problema.....	12
1.4 Objetivos.....	13
1.5 Propuesta de Solución.....	13
<b>2. FUNDAMENTOS PARA EL DESARROLLO DE SISTEMAS.....</b>	<b>14</b>
2.1 Sistemas de Información.....	14
2.1.1 Sistemas de procesamiento de datos.....	15
2.1.2 Sistemas Informáticos para la administración.....	15
2.1.3 Sistemas de apoyo para la toma de decisiones.....	16
2.2 Descripción del sistema.....	16
2.3 Ciclo de desarrollo del sistema.....	17

<b>2.4</b>	<b>Tecnología de la información.....</b>	<b>17</b>
2.4.1	Hardware. ....	17
2.4.2	Software. ....	19
2.4.3	Bases de Datos. ....	26
2.4.4	Manejadores de Bases de Datos.....	30
2.4.5	Análisis para la implementación del sistema.....	36

### **3. ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN**

#### **CURRICULAR PARA LA FACULTAD DE INGENIERÍA..... 41**

<b>3.1</b>	<b>Los elementos del sistema.....</b>	<b>41</b>
<b>3.2</b>	<b>Dominio y manejo de la información.....</b>	<b>44</b>
<b>3.3</b>	<b>Flujo y estructura de la información.....</b>	<b>48</b>
3.3.1	Caso del Personal Académico de la Facultad. ....	48
3.3.2	Caso de la Dependencia. ....	49
3.3.3	Caso del Administrador.....	50
<b>3.4</b>	<b>Requerimientos de Herramientas (Software y Hardware).....</b>	<b>50</b>
3.4.1	Hardware: ....	50
3.4.2	Software ....	50
<b>3.5</b>	<b>Otros Requerimientos .....</b>	<b>52</b>

### **4. DISEÑO DEL SISTEMA DE INFORMACIÓN CURRICULAR PARA LA**

#### **FACULTAD DE INGENIERÍA..... 54**

<b>4.1</b>	<b>Estructura del sistema .....</b>	<b>54</b>
<b>4.2</b>	<b>Diseño de la interfaz gráfica .....</b>	<b>57</b>
<b>4.3</b>	<b>Manejo de la información.....</b>	<b>62</b>
<b>4.4</b>	<b>Diseño de la base de datos.....</b>	<b>65</b>
4.4.1	Especificaciones para la creación de las tablas .....	67
<b>4.5</b>	<b>Diseño del rep orte de datos.....</b>	<b>70</b>



<b>5.</b>	<b>CONFIGURACIÓN Y DESARROLLO DEL SISTEMA DE LA FI.....</b>	<b>72</b>
5.1	Configuración del servidor.....	72
5.1.1	Sistema Operativo.....	72
5.1.2	Software.....	73
5.2	Elaboración de la base de datos.....	77
5.3	Programación de Clases.....	78
5.4	Programación de JSP's.....	79
5.4.1	Formularios.....	79
5.4.2	Despliegue de Información.....	80
5.4.3	Diseño.....	81
<b>6.</b>	<b>IMPLEMENTACIÓN DEL SISTEMA. ....</b>	<b>82</b>
6.1	Pruebas.....	82
6.1.1	Prueba de función.....	83
6.1.2	Prueba de desempeño (rendimiento). ....	83
6.1.3	Prueba de Aceptación.....	86
6.1.4	Prueba de Seguridad. ....	87
6.1.5	Pruebas de Portabilidad y Usabilidad. ....	88
6.2	Mantenimiento.....	88
<b>7.</b>	<b>MANUALES.....</b>	<b>89</b>
7.1	Manual Técnico.....	89
7.1.1	Introducción.....	89
7.1.2	Requerimientos del sistema.....	89
7.1.3	Respaldo.....	91
7.1.4	Documentos de referencia.....	92
7.2	Manual del Académico.....	93
7.2.1	Secciones del Sistema.....	96
7.2	Manual del Administrador.....	104

**8. CONCLUSIONES..... 107**

**9. BIBLIOGRAFIA ..... 110**

# **1. Introducción.**

## **1.1 Antecedentes.**

La globalización económica de nuestro tiempo con sus características de complejidad e incertidumbre, aumenta la intensidad de la competencia y sitúa al conocimiento como un elemento estratégico que obliga a las instituciones de enseñanza superior a desarrollarse en un espacio de productividad global.

Estos cambios han generado nuevos y grandes desafíos a nuestra realidad educativa, científica y tecnológica. Nuestro país enfrenta importantes retos asociados con la internacionalización de valores y la intensificación en el intercambio de conocimientos.

La Universidad Nacional Autónoma de México, una corporación pública -organismo descentralizado del Estado- dotado de plena capacidad jurídica, tiene por fines impartir educación superior para formar profesionistas, investigadores, profesores universitarios y técnicos útiles a la sociedad; organizar y realizar investigaciones, principalmente acerca de las condiciones y problemas nacionales, y extender con la mayor amplitud posible, los beneficios de la cultura.

En ella, se cuenta con el derecho de organizarse como lo estime mejor, impartir sus enseñanzas y desarrollar sus investigaciones de acuerdo con el principio de libertad de cátedra y de investigación.

Para el presente trabajo, centraremos nuestra atención en la Facultad de Ingeniería, donde después de un análisis de sus sistemas administrativos, se ha descubierto la necesidad de automatizar algunos procesos administrativos, como un paso necesario en su transición a la globalización, ya que no sólo en la educación debe ser pionera, si no también sus procesos administrativos deben ser ejemplo para otras instituciones.

Es así que, a partir del año 2003, la Facultad de Ingeniería está empeñada en la "modernización educativa" acorde con las exigencias de nuestro tiempo; para ello se estableció el Plan de Desarrollo 2003-2007, como una acción temporal y permanente.

El principio organizativo de la Facultad de Ingeniería que norma su desarrollo institucional, se basa en políticas y directrices que garanticen el cumplimiento de las tareas fundamentales de La Universidad, las cuales se hayan centradas en una serie de instancias que las conforman y son mostradas en el siguiente organigrama:

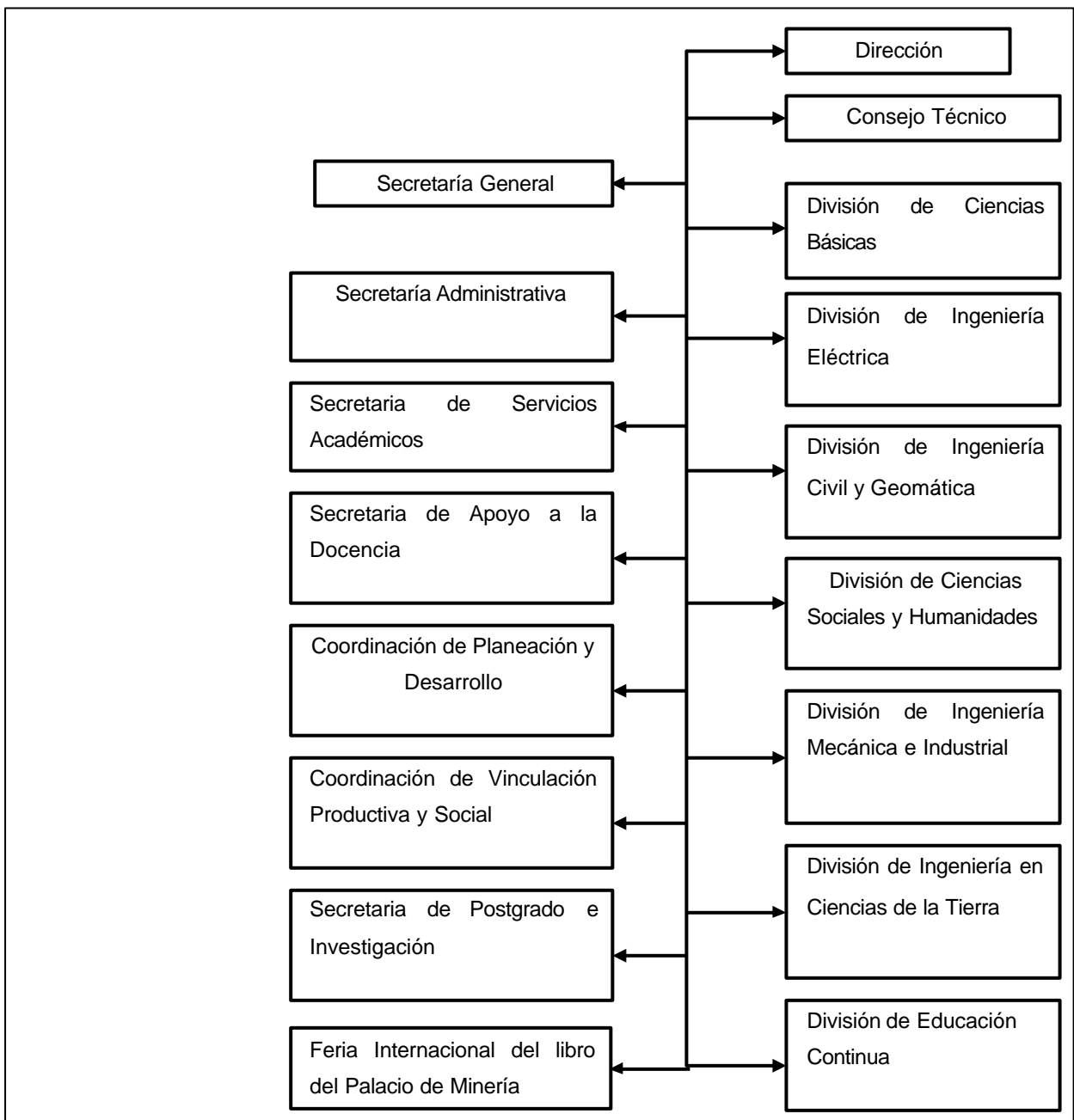


Figura 1.1. Organigrama de la Facultad de Ingeniería.

### **1.1.1 Consejo Técnico (CT).**

Es la autoridad colegiada de la Facultad para tomar decisiones sobre todos y cada uno de los asuntos académicos de la Facultad. Su ámbito de acción cubre aspectos como la aprobación de planes y programas de estudio ó la sanción de todos los asuntos del personal académico. En sus etapas iniciales, corresponde al Consejo Técnico aprobar o no, todos los trámites relativos al personal académico. El Consejo Técnico está constituido por el Director y por profesores representantes de las divisiones de ciencias básicas, posgrado, ciencias sociales y humanidades, y de cada una de las carreras que se imparten en la Facultad, así como por dos representantes de los alumnos.

Para realizar sus tareas el Consejo Técnico se ve auxiliado por las Comisiones Dictaminadoras, que tienen como entradas solicitudes sobre determinados casos, como salidas los respectivos dictámenes.

### **1.1.2 Dirección.**

De acuerdo con la Legislación Universitaria, al Director le corresponden las funciones de representar a la Facultad; convocar al Consejo Técnico y presidir con voz y voto las sesiones; cuidar el cumplimiento de las disposiciones que dicte el Consejo Técnico; vigilar dentro de la Facultad el cumplimiento de la Legislación Universitaria, de los planes y programas de trabajo y, en general, de las disposiciones y acuerdos que normen la estructura y el funcionamiento de la Universidad, dictando las medidas conducentes; promover convenios de cooperación e intercambio académicos con otros órganos e instituciones; informar periódicamente a la comunidad de las actividades desarrolladas y autorizar las funciones de los órganos de la Facultad; y las demás funciones que le señala la Legislación Universitaria.

### **1.1.2 Secretaría General.**

Es la entidad encargada de planear, implantar, evaluar y coordinar la ejecución de los asuntos con carácter académico de la Facultad. Entre sus funciones se encuentran:

- Ejercer el secretariado del Consejo Técnico.
- Supervisar las actividades de los órganos de apoyo académico a su cargo

- Colaborar con el Director en las actividades de planeación, evaluación y apoyo académico de la Facultad.
- Ser el enlace entre las Divisiones, la Dirección y el Consejo Técnico de la Facultad.
- Realizar los trámites relativos al personal académico de la Facultad ante la Administración Central (La Dirección General de Personal y la Dirección General de Asuntos del Personal Académico).

La estructura de la Secretaría General se compone de cuatro departamentos, dos unidades y una coordinación.

Los departamentos que intervienen en la atención de los asuntos del personal académico son:

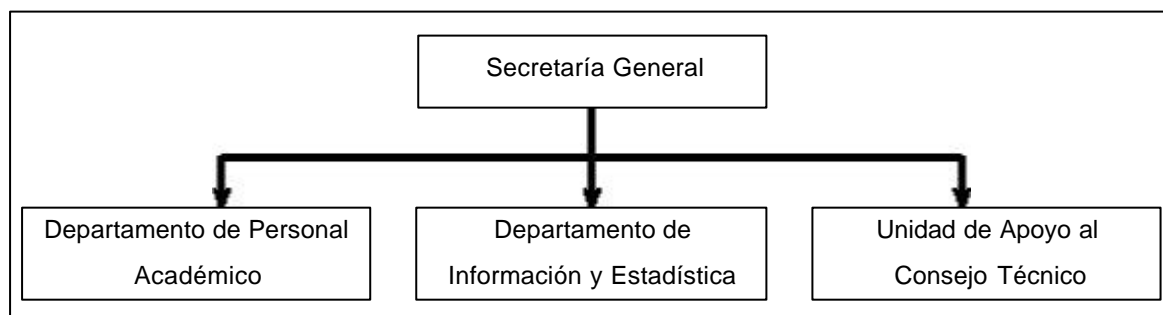


Figura 1.2. Departamentos involucrados en los asuntos del personal académico

El Departamento de Personal Académico (DPA) realiza todas las gestiones necesarias en la tramitación de los asuntos del personal académico: concentra formatos y documentos, verifica información, prepara y envía la documentación necesaria a la DGP.

El Departamento de Información y Estadística es responsable de la custodia de los expedientes del personal académico, así como del acopio y procesamiento de información relativa a éstos.

La Unidad de Apoyo al Consejo Técnico se encarga de presentar la información al Consejo Técnico y a sus comisiones auxiliares.

Es necesario aclarar que los tres anteriores departamentos realizan otras actividades adicionales a las citadas y se mencionan algunas.

La Secretaría General recibe todos los trámites propuestos por los académicos y las divisiones, para ser concentrados y someterlos a la consideración del Consejo Técnico de la Facultad. Dentro de esta actividad, envía los casos que así lo requieran a las comisiones dictaminadoras.

Una vez que las solicitudes han sido sancionadas por el Consejo Técnico, la Secretaría General debe recabar de las divisiones toda la documentación pertinente y dar cauce a los trámites, principalmente ante la DGP y la DGAPA.

La DGP únicamente reconoce a la Secretaría General de la Facultad como la entidad con la cual puede realizar los trámites del personal académico de la Facultad.

### **1.1.3 Secretaría de Servicios Académicos.**

Coordina la ejecución de las actividades de administración escolar, apoya a profesores y alumnos, y proporciona los servicios que se requieren en la Facultad para el mejor desempeño de las actividades académicas.

Algunas de sus funciones son: proporcionar a profesores y alumnos los apoyos de administración escolar, difusión, biblioteca y estudios académicos; preparar los informes que le sean solicitados por la Dirección y que coadyuven al desarrollo académico de la Facultad; coordinar y promover aquellas actividades socioculturales, deportivas y recreativas que propicien la formación integral de los estudiantes de la Facultad.

Para llevar a cabo sus funciones, el Secretario de Servicios Académicos se apoya en los siguientes subórganos a su cargo:

- Unidad de Servicios de Cómputo Administrativo (USECAD)
- Coordinación de Bibliotecas
- Departamento de Administración Escolar
- Departamento de Audiovisual y Fotocopiado
- Publicaciones
- Apoyo a la Comunidad

#### **1.1.4 Secretaría Administrativa.**

Su función principal es regular el mejor aprovechamiento de los recursos financieros de la Facultad de acuerdo con las políticas fijadas por el Director.

Funciones principales: formular el presupuesto asignado a la Facultad y, una vez aprobado, supervisar su correcto ejercicio; custodiar el registro adecuado de las operaciones contables que se realicen; vigilar los depósitos y utilización de los ingresos extraordinarios que se generen, conforme al reglamento vigente; cuidar que las adquisiciones de bienes y servicios se efectúen eficaz y oportunamente de acuerdo con los procedimientos de la institución; supervisar que los trámites relativos a los movimientos del personal académico administrativo se lleven a cabo correcta y oportunamente; supervisar la limpieza de las instalaciones de la Facultad; mantener la seguridad de su comunidad y salvaguardar sus bienes patrimoniales; mantener relaciones cordiales con los representantes sindicales de las AAPAUNAM y el STUNAM; a fin de agilizar los trámites requeridos.

Para llevar a cabo estas funciones, el Secretario Administrativo se apoya en los siguientes órganos:

- Coordinación de Finanzas
- Coordinación de Servicios Generales
- Departamento de Presupuesto
- Departamento de Contabilidad
- Departamento de Personal Administrativo
- Departamento de Adquisiciones y Servicios
- Departamento de Intendencia y Vigilancia
- Departamento de Almacén e Inventarios
- Departamento de Transporte
- Departamento de Mantenimiento y dos delegaciones administrativas que se ubican en postgrado y Educación Continua, respectivamente.



### **1.1.5 Divisiones.**

Son las unidades organizacionales en las que el Director de la Facultad delega sus atribuciones. Entre sus principales competencias se encuentran las de planear y organizar las actividades académico-administrativas acordes con los objetivos institucionales de la Facultad; promover y regular el desarrollo de las actividades de docencia, investigación y difusión de la cultura; velar el cumplimiento de los planes y programas de estudio; fomentar entre el personal docente la actualización y superación académica; y coordinar académica y administrativamente las carreras que en su caso le competan.

La Facultad consta de siete divisiones:

- División de Ciencias Básicas (DCB)
- División de Ciencias Sociales y Humanidades (DCSH)
- División de Ingeniería Eléctrica (DIE)
- División de Ingeniería Mecánica e Industrial (DIMEI)
- División de Ingeniería en Ciencias de la Tierra (DICT)
- División de Ingeniería Civil, Geomática (DICyG)
- División de Educación Continua (DEC)

En la generalidad, cada división organizacionalmente se compone de:

- Jefe de división
- Secretario académico
- Jefes de departamento
- Coordinadores de carrera
- Coordinadores de asignatura
- Subcoordinadores de asignatura
- Encargado del personal académico

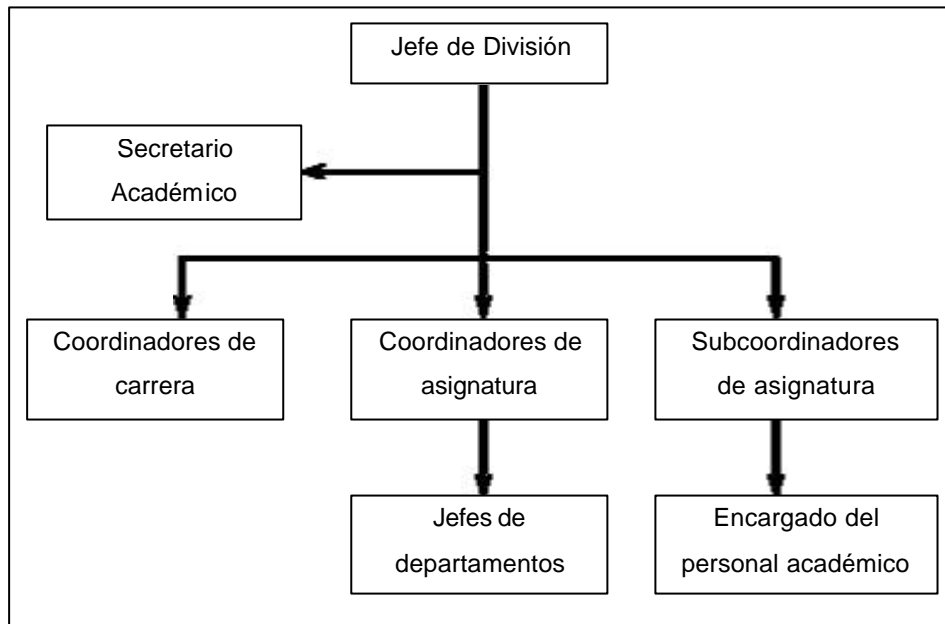


Figura 1.3. Composición de división

Por otra parte, la DEC no posee personal académico, ya que sus labores las realiza con personal contratado por honorarios por servicios profesionales.

Como podemos observar en la estructura organizacional de las divisiones, el encargado del personal académico no existe formalmente dentro de la estructura de la División, obedece a un nombramiento por honorarios que por lo general recae en personal académico contratado para realizar funciones administrativas, o bien, en personal de confianza.

Cada División tiene la facultad de iniciar todos los trámites que requiere el personal académico adscrito a ella, siendo también responsable de la administración de dicho personal.

El Personal Académico es uno de los actores fundamentales en la institución. El encargado de transmitir el conocimiento a los alumnos para lograr su formación profesional.

- Figuras del Personal Académico:
- Técnicos académicos
- Ayudantes de profesor o de investigador
- Profesores e investigadores

La Facultad cuenta con una planta académica integrada por 1,252 profesores, de los cuales 237 son profesores de carrera y 1,015 de asignatura; adicionalmente tiene 87 técnicos académicos y 377 ayudantes de profesor.

## **1.2 Problemática.**

### **1.2.1 Análisis de la Situación Actual.**

Este trabajo pretende proporcionar un marco administrativo adecuado, utilizando un sistema de información y un conjunto de reglas normativas, que agilice la información curricular del personal académico de la Facultad de Ingeniería.

Podría parecer contradictorio ofrecer información sobre el personal académico cuando precisamente este estudio busca determinar la mejor forma de obtenerla. La realidad es que, para obtener esta sencilla información, fue necesario hacer una gran inversión de tiempo y recursos.

Esto se debe, en parte, a la estructura organizacional de la Facultad, lo que genera un flujo de información lento a través de todos los niveles jerárquicos, provocándose también deformaciones, llevando un impacto a la toma de decisiones.

Es un hecho indiscutible, para poder realizar la planeación en la Facultad es necesario conocer a detalle los recursos que dispone, entre ellos su personal académico, hasta ahora semi-conocido para los encargados de la planeación, puesto se conoce de manera parcial su perfil, su conformación, su origen, sus grados académicos, su currículum, etc., por tanto, es indispensable sentar las bases para la conformación de un sistema contenedor de información curricular para la toma de decisiones.

Es necesario destacar que la información curricular es vital para el crecimiento del personal académico. De ahí la importancia de contar con un conjunto de procedimientos que permitan la disposición rápida y confiable de la información académica, en todos los niveles y situaciones.

El objetivo de este estudio es proporcionar un conjunto de procesos que permitan obtener información confiable y oportuna sobre el personal académico, sentando las bases necesarias para el desarrollo de un sistema de información basado en cómputo.

El enunciado anterior en realidad se refiere a un subsistema que junto con otros ya existentes formaría parte de un gran sistema de información del personal académico de la Facultad de Ingeniería.

### **1.2.1 Problemática de la información curricular del personal académico.**

El disponer de información para la toma de decisiones, sin que ésta sea producto de un proceso de validación, puede repercutir negativamente, por lo cual es necesario contar con procedimientos perfectamente establecidos para brindar confiabilidad en la información de ellos generada.

Para efectos de este estudio se considerará exclusivamente la información del personal académico, con la finalidad de estandarizarlos y alcanzar una mayor calidad y oportunidad en la información.

Dentro de la Facultad existen diversos niveles en el flujo de la información. El nivel más bajo corresponde a las transacciones que se dan entre los profesores y sus coordinadores o jefes de departamento. Son los coordinadores o los jefes de departamento los encargados de conformar a la planta académica de la Facultad, ya que proponen, de acuerdo con la oferta de grupos, a los profesores que impartirán las asignaturas, asimismo reciben las peticiones de los profesores de carrera también para impartir clase, asignan a los ayudantes y coordinan las tareas de apoyo, tales como los laboratorios y prácticas.

Todas estas propuestas, en lo general, concentradas por cada jefe de división, quien le indica al responsable del personal académico de su propia división que realice los trámites correspondientes. El papel del secretario académico es, en lo general, el de coordinar a los distintos jefes de departamento y supervisar al encargado del personal académico.

El siguiente nivel se establece una vez que la división tiene una propuesta sobre su planta académica, dado que le corresponde a la Secretaría General de la Facultad concentrar a su vez las peticiones de todas las divisiones para someterlas a la consideración del Consejo Técnico.

El nivel final es aquel en donde una vez aprobadas las propuestas por parte del Consejo Técnico, nuevamente la Secretaría General concentra los documentos necesarios y realiza el trámite ante la Dirección General de Personal (DGP), en ocasiones, ante la Dirección General de Asuntos del Personal Académico (DGAPA). Esta gestión es realizada por el Departamento de Personal Académico de la Secretaria General de la Facultad.

Ha sido necesario realizar una investigación de campo en todos y cada uno de los niveles que conforman a la Facultad, con el fin de conocer la problemática y detectar las posibles causas que intervienen en la gestión administrativa y generación de información propia del personal académico, con esto se obtuvo la siguiente información:

- Dada la lejanía entre el primer nivel y el nivel ejecutivo (que realiza la toma de decisiones), no es posible tener información actualizada.
- Las diferentes divisiones manejan la información de acuerdo con sus criterios, conveniencias y experiencias. De tal forma que para los concentradores de la información resulta muy complicado uniformizarla, teniendo incluso que modificarla, corriendo el riesgo, involuntario de corromperla.
- El trabajo de cada división se hace sin tomar en cuenta al resto de ellas, es decir, no existe comunicación. Esta situación, producto quizá de la falta de políticas o procedimientos provoca en el mejor de los casos, pérdida de tiempo.
- En virtud de que cada división ha organizado su información de acuerdo con sus necesidades, resulta muy común encontrar que se han creado sentimientos de pertenencia sobre la información institucional, así como en la forma de procesada. Esta deformación del sentido de responsabilidad ha derivado en prácticas en contra de la organización, como ofrecer información falsa o incompleta, lo cual afecta
- El tiempo que se invierte en la realización de un curriculum puede ser hasta de un año. Esta dispersión en el tiempo ocasionada por múltiples factores, conjuntamente con la carga normal de trabajo de los involucrados en el proceso, suele provocar el olvido administrativo

del asunto, pudiendo derivarse en problemas. Asimismo, estos procesos en trámite afectan la toma de decisiones ya que se desconoce su resultado.

- No se han definido las responsabilidades, de tal forma que éstas no son plenamente asumidas por alguien. En los casos en que los académicos sufren algún problema, son ellos mismos quienes deben dar seguimiento hasta lograr la solución del caso.
- Es imposible cuantificar las omisiones o las equivocaciones, ya que no existe una plena supervisión de la gestión administrativa.
- La falta de información provoca el desconocimiento de las consecuencias de una decisión.
- Algunos Académicos se rehúsan a la tecnología.

En lo relativo a información curricular del personal académico, y con el objeto de conocer a fondo dichos procedimientos al interior de la Facultad, se realizaron una serie de entrevistas con los responsables de gestionar los movimientos de personal académico de cada una de las divisiones que conforman la dependencia, encontrándose diversos problemas, tales como:

- No existe un manual de procedimientos para realizar un curriculum.
- No cuentan oportunamente con las nuevas políticas que se generan en la información a través de la red de cómputo de la Facultad.
- No todos los responsables se auxilian en el cómputo para realizar sus tareas e incluso, en ocasiones, ni siquiera las personas a su cargo (cuando existen) tienen la más elemental capacitación en cómputo.
- La mayoría no cuenta con el equipo necesario que le permita el manejo de información.
- Varios responsables de la gestión administrativa son miembros del personal sindicalizado, sujetos a un horario inamovible con sus correspondientes consecuencias.
- La información que tienen del personal académico es parcial y con errores.
- No tienen un adecuado control de gestión sobre los movimientos que se generan en la división.

### **1.3 Descripción del problema.**

No se cuenta con un sistema que permita la recaudación de la Información curricular de los académicos de la Facultad de Ingeniería

## **1.4 Objetivos.**

Que la Secretaria General pueda recabar información curricular de los Académicos de la Facultad de una manera más rápida y segura, facilitando la evaluación del trabajo académico del profesorado, así como disminuir el tiempo que dedican al llenado de diferentes formatos.

Proporcionar una herramienta tecnológica y a la vanguardia para agilizar los procesos administrativos relacionados con los Académicos de la Facultad

Aprovechar los recursos tecnológicos para tener la información curricular al día

Obtener indicadores y estadísticas fiables y actualizadas de manera expedita acerca del quehacer académico de la Facultad de Ingeniería.

Aplicar los conocimientos adquiridos durante nuestra permanencia en esta facultad en el desarrollo de una herramienta que sea útil para la Facultad y la Sociedad.

## **1.5 Propuesta de Solución.**

Una vez realizado el análisis de la problemática de la información curricular del personal académico de la Facultad de Ingeniería y las causas que originan el problema tratado en este estudio, estamos en posibilidad de hacer algunas recomendaciones para su solución a manera de propuesta, y que constituyen la base para el diseño del sistema de información curricular del personal académico de la Facultad de Ingeniería, que ayude a recabar toda la información necesaria de los académicos. De tal forma que facilite el trabajo tanto a la Secretaria General como al Académico, así como Para satisfacer y cumplir los objetivos trazados. Ya que Debido a la relevancia de la Facultad de Ingeniería y a la complejidad de la información curricular de sus académicos, a la actividad académica evaluada de manera periódica, tanto por la UNAM, como por el sistema educativo nacional y organismos internacionales, es necesario contar con dicho sistema

## **2. Fundamentos para el Desarrollo de Sistemas.**

En este capítulo se realizara el diseño y desarrollo del sistema de Información Curricular de la Facultad de Ingeniería. Iniciaremos con la evaluación de los diferentes lenguajes utilizados para la web, manejadores de bases de datos y la arquitectura Cliente/Servidor; con la finalidad de poder tomar la mejor decisión que nos brinde las herramientas necesarias para el desarrollo de dicho sistema.

En el capitulo anterior se definieron las especificaciones y las condiciones en las que operará el sistema, uno de estos puntos a considerar en la confiabilidad y la disponibilidad del sistema por lo que requerimos un servidor que satisfaga estas características.

### **2.1 Sistemas de Información.**

Un sistema de información es la combinación de trabajo, información, personas y de las tecnologías de información, con las cuales podemos llegar a las metas y alcanzar los objetivos para una mejor administración, regulación, planificación y el control dentro de una organización. Las actividades básicas que producen la organización de la información requerida para la toma de decisiones, análisis de problemas y creación de nuevos productos y servicios; son la entrada, procesamiento y salida de la información, las cuales explicaremos a continuación.

- La entrada: es la captura ó colecta de datos dentro de la organización o de su ambiente externo para su procesamiento en el sistema.
- El procesamiento: consiste en la transformación de los datos de entrada en un formato más explícito para el personal laborable de la organización.
- Salida de la información: es la distribución de la información procesada al personal o hacia las actividades donde se requiera.

Los Sistemas de Información, requieren de una retroalimentación, donde la salida de los datos se vuelve a tomar en cuenta con los elementos de la organización permitiendo ayudar y evaluar la correcta entrada de la información.



El propósito de los sistemas de información siempre dependerá de las necesidades de la empresa o la institución, los sistemas de procesamiento de datos, los sistemas de información para la administración (MIS, Management Information systems), y los sistemas de apoyo para la toma de decisiones (DSS, Decisión Support Systems), son diferentes tipos de sistemas de información computarizados que se analizan y diseñan mediante la aplicación de los conceptos y las técnicas del diseño y del análisis de sistemas.

### **2.1.1 Sistemas de procesamiento de datos.**

Los sistemas de procesamiento de datos son aquellos sistemas de información computarizados que se desarrollan para procesar grandes volúmenes de información generada en las funciones administrativas, tales como la nómina o el control de inventarios. Los sistemas de procesamiento: ayudan al elemento humano para su mejor manejo de la información, automatizando las tareas y funciones que se realizan manualmente.

### **2.1.2 Sistemas Informáticos para la administración.**

Los sistemas de información para la administración (MIS) no sustituyen a los sistemas de procesamiento de datos, más bien todos toman en cuenta a las funciones de procesamiento de datos. Los MIS son sistemas que se sustentan en la relación que surge entre las personas y las computadoras. Los MIS requieren para su operación de las personas, del software (programas de cómputo) y del hardware (computadoras, impresoras, etc.). Estos sistemas de información para la administración soportan un amplio espectro de tareas de las organizaciones, más aún que los sistemas de procesamiento de datos, incluyendo el análisis, decisiones y la toma de decisiones.

Los usuarios de los sistemas de información para la administración, utilizan una base de datos compartida para tener acceso a la información. Dicha base de datos, almacena, tanto datos como modelos que ayudan, al usuario en la interpretación y el uso de la información. Los sistemas de información para la administración generan la información que eventualmente se utiliza en la toma de decisiones. Un sistema de información para la administración puede llegar a unificar ciertas funciones informáticas de la empresa, aunque el sistema como tal no exista dentro del negocio como una estructura individual.

### **2.1.3 Sistemas de apoyo para la toma de decisiones.**

El sistema de apoyo para la toma de decisiones es un tercer tipo de sistema de información computarizada (DSS; Decision Support Systems). El sistema de apoyo para la toma de decisiones es similar a los sistemas de información tradicionales para la administración, en el sentido de que ambos dependen de una base de datos como fuente de información; pero se distingue del sistema de información para la administración, al hacer énfasis en el soporte en cada una de las etapas de la toma de decisiones. Sin embargo, la decisión en si, depende de la persona responsable de la misma. Los sistemas de apoyo para la toma de decisiones se diseñan con una orientación hacia la persona o el grupo que los utilizará, y no como los sistemas de información tradicionales para la administración.

## **2.2 Descripción del sistema.**

Un sistema se integra de componentes básicos que realizan una tarea definida sobre la base de los objetivos funcionales definidos inicialmente. A su vez, cada uno de estos componentes está constituido por una serie de actividades que se realizan con el objeto de hacer cumplir las especificaciones establecidas para cada una de las componentes principales y que en su conjunto permiten lograr los objetivos del sistema en su totalidad.

La descripción de un sistema, es llevada en diferentes pasos, uno de estos pasos lo constituye la identificación de cada una de los componentes del sistema; sin embargo, antes de realizar este paso o proceso, es importante tener claros los objetivos propios al sistema. La combinación de los elementos anteriores deberá definir con precisión las características y límites del sistema, esto traería los siguientes beneficios:

- Conocer las características propias de cada componente.
- Definir los recursos del sistema.
- Considerar la evolución en el tiempo de cada componente y del sistema.

De lo anterior se establece que un sistema no sólo se describe sobre la base de sus objetivos y componentes, sino también se ven involucran otros factores como lo son sus recursos, medio ambiente y evolución en el tiempo.

## **2.3 Ciclo de desarrollo del sistema.**

El ciclo de desarrollo de los sistemas ó ciclo de vida de los sistemas (SDLC; Systems Development Life Cycle). Es un enfoque por etapas de análisis y de diseño, que postula que el desarrollo de los sistemas mejora cuando existe un ciclo específico de actividades del analista y de los usuarios.

Aquí dividimos el ciclo en siete etapas. Aunque cada etapa se presenta de manera discreta, nunca se lleva a cabo como un elemento independiente. En lugar de ello, se realizan al mismo tiempo diversas actividades, y éstas llegan a repetirse.

Por ello es de mayor utilidad suponer que el ciclo de desarrollo de los sistemas transcurre en etapas (con actividades en plena acción que luego cesan poco a poco) y no como elementos separados.

## **2.4 Tecnología de la información.**

### **2.4.1 Hardware.**

#### **2.4.1.1 Arquitectura Cliente/Servidor.**

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información, sus principales componentes son: los clientes, los servidores y la infraestructura de comunicaciones.

Los clientes interactúan con el usuario, en forma gráfica, aunque también se comunican a través de procesos auxiliares cuya tarea es establecer conexión con el servidor, solicitar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

Por lo general las funciones de los clientes son:

- Manejo de la interfaz del usuario
- Captura y validación de datos de entrada
- Generación de consultas e informes sobre las bases de datos

Los servidores proporcionan un servicio al cliente y devuelven los resultados, en algunas situaciones hay procesos auxiliares, que se encargan de recibir las solicitudes del cliente, verificar la protección, iniciar un proceso en el servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente.

Los servidores realizan las siguientes funciones:

- Gestión de periféricos compartidos
- Control de accesos concurrentes y bases de datos compartidas
- Enlaces de comunicaciones con otras redes de área local o externa.

Para establecer la comunicación entre los clientes y los servidores es necesaria una infraestructura de comunicaciones, que nos proporciona los mecanismos básicos de direccionamiento y transporte. La mayor parte de los sistemas cliente/servidor actuales, se basan en redes locales, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Las principales características de la arquitectura cliente/servidor son las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos ó ningún cambio en el cliente.

Como ejemplos de clientes pueden citarse interfaces de usuario para enviar comandos a un servidor, API (Application Programming Interface) para el desarrollo de aplicaciones distribuidas, herramientas

en el cliente para hacer acceso a servidores remotos (por ejemplo, servidores de SQL) o aplicaciones que solicitan acceso a servidores para algunos servicios.

Una infraestructura cliente/servidor consta de tres componentes esenciales, todos ellos de igual importancia y estrechamente ligados:

- **Plataforma operativa:** esta deberá soportar los modelos de distribución cliente/servidor y los servicios de comunicación deberán utilizar preferentemente componentes estándar de la industria para los servicios de distribución. Los desarrollos propios deben coexistir con las aplicaciones estándar y su integración deberá ser imperceptible para el usuario. Igualmente, podrán alojarse programas escritos utilizando diferentes tecnologías y herramientas.
- **Entorno de desarrollo de aplicaciones:** Aunque es recomendable evitar la proliferación de herramientas de desarrollo, se garantizará que el enlace entre estas y el middleware no sea excesivamente rígido. Por lo que se podrán utilizar diferentes herramientas para desarrollar partes de una aplicación. Un entorno de aplicación incremental, debe posibilitar la coexistencia de procesos cliente y servidor desarrollados con distintos lenguajes de programación y/o herramientas, así como utilizar distintas tecnologías (por ejemplo, lenguaje estructurado, lenguaje orientado a objetos, multimedia).
- **Administración de sistemas:** Estas funciones aumentan considerablemente el costo de una solución, sin embargo no se pueden evitar, por lo que deben adaptarse a las necesidades de la organización.

En las primeras fases de la solución se decidirá la plataforma operativa y el entorno de desarrollo.

## **2.4.2 Software.**

### **2.4.2.1 Lenguajes de programación.**

Los lenguajes de programación son unas de las herramientas más importantes para la implementación de nuestro sistema. Dado que hoy en día se cuenta con gran variedad de lenguajes de programación que son utilizados como herramientas de desarrollo, procederemos a realizar una

evaluación de estos, para determinar cual será el que nos garantice una mayor funcionalidad, seguridad y costos, por lo que a continuación daremos las características de algunos de ellos.

#### **2.4.2.2 ASP.**

ASP es un lenguaje de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP nos ofrece las siguientes ventajas importantes acerca de los otros modelos de programación:

- Rendimiento. ASP es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time.
- Eficacia y flexibilidad. Debido a que ASP se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web.
- Facilidad de uso. ASP emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web.
- Escalabilidad y disponibilidad. ASP se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.
- Seguridad. La seguridad de los programas ASP depende de la tecnología de Windows.
- Plataforma: con respecto a la plataforma ASP solo es soportada por bajo sistemas operativos Windows, y servidor de IIS propio de tecnología Windows

#### **2.4.2.3 PHP.**

PHP es un lenguaje de programación soportado por HTML. La sintaxis está heredada de C, Java y Perl. Este lenguaje está orientado para los constructores de páginas Webs, permitiéndoles crear páginas dinámicamente generadas de forma rápida.

Desaparece por tanto el problema de si el cliente puede o no ejecutar sentencias de comandos, el servidor Web solo envía el resultado en código **HTML** standard interpretable por cualquier explorador.

#### **2.4.2.4 JAVA.**

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

Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas de punta.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma (Windows, Linux, Apple, etc.). Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier estación de trabajo. Lo cual es una ventaja para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo. Esto es posible porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet. Java fue pensado originalmente para utilizarse en cualquier tipo de electrodoméstico pero la idea fracasó. Uno de los fundadores de SUN rescató la idea para utilizarla en el ámbito de Internet y convirtieron a Java en un lenguaje potente, seguro y universal gracias a que lo puede utilizar todo el mundo y es gratuito. Una de los primeros triunfos de Java fue que se integró en el navegador Netscape y permitía ejecutar programas dentro de una página web, hasta entonces impensable con el HTML.

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas veces con grandes ventajas. 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 de la web se puede hacer utilizando Java.

Sus características principales son las siguientes:

- Java es un lenguaje de programación orientado a objetos muy completo, simple, distribuido, interpretado, robusto, seguro, de arquitectura neutral, portable, multitarea y dinámico"
- Puede utilizarse para presentar animaciones, figuras, juegos, para responder a acciones del usuario, o cualquier otra tarea. El proceso para incluir un Applet en una página Web consiste en escribir el código Java, compilarlo, e insertar el Applet utilizando el lenguaje HTML
- Aplicaciones en general: Este lenguaje nos permite crear toda clase de programas, es pues un lenguaje de propósito general. Los programas java no necesitan ser recompilados cuando se desea cambiar de plataforma, los programas resultantes son un conjunto de instrucciones en el código máquina del procesador sobre el que serán ejecutados dichos programas. Java genera programas que contienen lo que se llama bytecodes, estos bytecodes son un conjunto de instrucciones, independientes del procesador, y que serán interpretados por lo que llamamos máquina virtual (o incluso, pre-procesador Java) sobre la que se ejecutará cualquier programa que se desarrolle en este lenguaje, sin importar el Sistema Operativo que la máquina esté utilizando.
- Java no es un lenguaje compilado a código fuente nativo, como pueda serlo C++ o Pascal, pero tampoco se trata de un lenguaje de macros. El entorno de desarrollo Java funciona en buena medida como si utilizase un lenguaje compilable, salvo porque el compilador traduce el programa fuente Java a bytecodes en vez de a código máquina, y que estos bytecodes no se ejecutan directamente en la CPU, sino mediante la máquina virtual, que hace las funciones de intérprete
- Java es un lenguaje realmente atractivo por distintos motivos, por ejemplo su enfoque ala web y su independencia de funcionamiento sin importarnos el sistema operativo que se use (Unix, Windows, Linux, etc.)



### 2.4.2.5 JSP.

Una página Java en servidor (JSP, Java Server Pages) es una plantilla para una página Web que emplea código Java para generar un documento HTML dinámicamente. Las páginas JSP se ejecutan en un componente servidor conocido como contenedor de JSP, que las traduce a servlets Java equivalentes. Es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basado en scripts que utilizan una variante del lenguaje java.

Por esta razón los servlets y las páginas JSP están íntimamente relacionados. Lo que se puede hacer con una tecnología es, en gran medida, también posible con otra; aunque cada una tiene capacidades propias, las páginas JSP tienen todas las ventajas de los servlets por ejemplo:

- Tienen un mayor rendimiento y capacidad de adaptación (escalabilidad) que las secuencias de comandos CGI, porque se conservan en la memoria y admiten múltiples subprocesos.
- No es necesaria una configuración especial por parte del cliente.
- Incorpora soporte para sesiones http, lo que hace posible la programación de aplicaciones.
- Tienen pleno acceso a la tecnología Java, capacidad de reconocimiento del trabajo en red, subprocesos y conectividad a bases de datos sin las limitaciones de los applets del cliente.
- las páginas JSP tienen ventajas propias:
- Se vuelven a compilar automáticamente cuando es necesario.
- Como están en el espacio común de documentos del servidor Web, dirigirse a ellas es más fácil que dirigirse a los servlets.
- Como las páginas JSP son similares al HTML, tienen mayor compatibilidad con las herramientas de desarrollo Web.

La página JSP pasa por tres etapas en la evolución de su código:

- Código fuente JSP. Este código es el que realmente escribe el desarrollador. Se encuentra en un archivo de texto con extensión.jsp y consiste en una mezcla de código de plantilla HTML, instrucciones en el lenguaje Java, directivas JSP y acciones que describen como generar una página Web para dar servicio a una petición concreta.

- Código fuente Java. El contenedor de JSP traduce el código fuente JSP al código fuente de un servlet java equivalente. Este código fuente guarda en un área de trabajo y suele ser útil en el proceso de depuración de errores.
- Código Java compilado. Como cualquier otra clase Java, el código del servlet generado se compila en código de bytes en un archivo.class, preparado para ser cargado y ejecutado.

El contenedor de JSP administra cada una de estas etapas de la página JSP automáticamente, basándose en la situación temporal de cada archivo. Como respuesta a la petición http, el contenedor comprueba si el archivo fuente.jsp ha sufrido modificaciones desde que el código fuente Java se compiló por última vez.

Otro elemento que podríamos utilizar para nuestros desarrollos, es el proyecto llamado Jakarta cuyo producto más destacado Tomcat seguía la línea marcada por Java, es decir, la de ser gratuito.

#### **2.4.2.5.1 Alcance de JSP.**

Una página JSP no es más que un fichero de texto (y cuya extensión es.jsp) en cuyo interior hacemos uso de determinados elementos para generar una respuesta en el terminal del cliente, estos elementos en su mayoría son vistos por el motor de JSP como objetos y como tales objetos tienen un alcance o duración durante la cual van a ser accesibles. Estos alcances (scope) son de cuatro tipos diferentes.

- De página (page): Los objetos que se declaran a este nivel son solo accesibles dentro de la página en la cual fueron creados. Las referencias de los objetos creados con alcance de página se almacenan en otro objeto denominado pagecontext.
- De petición (request): En este caso los objetos declarados a este nivel son accesibles por todas las páginas que van a procesar la misma petición (request). Las referencias creadas sobre estos objetos se almacenan en el objeto request.
- De sesión (session): Los objetos con alcance de sesión son accesibles por todas aquellas páginas que están procesando peticiones dentro de la misma sesión en la cual el objeto fue creado. Las referencias de los objetos creados se almacenan dentro del objeto session.

- De aplicación (application): Es el nivel más alto en el sentido en que los objetos creados a este nivel son accesibles por todas las páginas que procesan peticiones dentro de la misma aplicación. Las referencias se almacenan en el objeto application

Cada uno de los niveles anteriores implica un mayor grado de visibilidad para los diferentes objetos que creemos, esto es muy importante a tener en cuenta sobre todo cuando estemos desarrollando páginas JSP que estén compuesta por múltiples páginas, muchos errores se deben a estar tratando de acceder a objetos que están fueran del alcance de la ejecución de la página actual.

Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a jsp le da mucha versatilidad. Los JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet.

Los JSP's y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de maquina. Siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente, por ejemplo conexión a bases de datos y manejo de sesiones.

#### **2.4.2.6 Servlets.**

Los Servlets son módulos que extienden los servidores orientados a petición-respuesta, como los servidores web compatibles con Java. Por ejemplo, un servlet podría ser responsable de tomar los datos de un formulario de entrada de pedidos en HTML y aplicarle la lógica de negocios utilizada para actualizar la base de datos de pedidos de la compañía.

Los Servlets son para los servidores lo que los applets son para los navegadores. Sin embargo, al contrario que los applets, los servlets no tienen interface gráfico de usuario.

Los servlets pueden ser incluidos en muchos servidores diferentes porque el API Servlet, el que se utiliza para escribir Servlets, no asume nada sobre el entorno o protocolo del servidor. Los servlets se están utilizando ampliamente dentro de servidores HTTP.

Los Servlets son un reemplazo efectivo para los scripts CGI. Proporcionan una forma de generar documentos dinámicos que son fáciles de escribir y rápidos en ejecutarse. Los Servlets también solucionan el problema de hacer la programación del lado del servidor.

Los servlets se utilizan para manejar peticiones de cliente HTTP. Por ejemplo, tener un servlet procesando datos posteados sobre HTTP utilizando un formulario HTML, incluyendo datos del pedido o de la tarjeta de crédito. Un servlet como este podría ser parte de un sistema de procesamiento de pedidos, trabajando con bases de datos de productos e inventarios, y quizás un sistema de pago on-line.

Características de los Servlets:

- Un servlet puede manejar múltiples peticiones concurrentes, y puede sincronizarlas. Esto permite a los servlets soportar sistemas como conferencias on-line
- Pueden reenviar peticiones a otros servidores y servlets. Con esto los servlets pueden ser utilizados para cargar balances desde varios servidores que reflejan el mismo contenido, y para particionar un único servicio lógico en varios servidores, de acuerdo con los tipos de tareas o la organización compartida.

### **2.4.3 Bases de Datos.**

La mayoría de los sistemas de información ya sean implantado en sistemas de cómputo tanto grandes como pequeños, utilizan una base de datos que pueden abarcar varias aplicaciones, por esta razón estos sistemas utilizan un administrador de base de datos.

Desde el surgimiento de las computadoras, el paso de la información a medios digitales ha requerido la búsqueda de formas de almacenamiento que permitan organizarla de una forma adecuada. Inicialmente se usaron los llamados Sistemas de Procesamiento de Archivos que eran un conjunto

de programas de prestaban servicios a los usuarios finales; basados en registros guardados en archivos, creándose respectivamente programas para su manipulación (obtener y guardar información). Éstos tenían varias desventajas:

- Redundancia en la información (repetición).
- Inconsistencia en la información (problemas en la actualización lo que provocaba que copias de ésta no concordaran).
- Falta en el manejo de la concurrencia (múltiples usuarios con actualizaciones al mismo tiempo).
- Dependencia entre programa y archivo.

Para trabajar de una manera más efectiva, surgieron las Bases de Datos y los Sistemas Manejadores de Bases de Datos.

#### **2.4.3.1 Definición.**

Una Base de Datos es un conjunto de información organizada y almacenada en un archivo lógico compuesto de uno o más archivos físicos cuya finalidad es la de servir a una aplicación o más de la mejor manera posible.

#### **2.4.3.2 Sistemas Administrador de Base de Datos.**

Existen tres características importantes referentes a los sistemas de bases de datos: la separación entre los programas de aplicación y los datos, el manejo de múltiples vistas por parte de los usuarios y el uso de un catálogo para almacenar el esquema de la base de datos consta de tres niveles que resulta muy útil para conseguir las características mencionadas.

- Nivel Interno: Describe todos los detalles para el almacenamiento de la base de datos, así como los métodos de acceso.
- Nivel Conceptual: Se describe la estructura de toda la base de datos para una comunidad de usuarios mediante un esquema conceptual. Este esquema oculta los detalles de las estructuras

de almacenamiento y se concentra en describir entidades, atributos, relaciones, operaciones de los usuarios y restricciones.

- Nivel Externo: Se describen varios esquemas externos o vistas de usuario, cada nivel externo describe la parte de la base de datos que interesa a un grupo de usuarios determinado y a este grupo le oculta el resto de la base de datos.

#### **2.4.3.3 Modelado de datos.**

El modelo conceptual de datos define y modela los aspectos importantes de la información que el cliente necesita saber a tener y las relaciones entre dicha información.

#### **2.4.3.4 El Modelo Relacional.**

Se basa en la percepción del mundo real, que consiste en un conjunto de objetos llamados entidades y relaciones entre ellas. Representa la estructura lógica general de la base de datos gráficamente.

- Entidad: Es un objeto concreto como una persona ó abstracto como un día festivo. Puede ser una persona, cosa o evento de interés para los usuarios. Siempre son sustantivos tangibles o intangibles. Una entidad está formada por un conjunto de atributos que son características de ésta.
  - Entidades fuertes o dominantes. Entidad que tiene suficientes atributos para formar una llave primaria.
  - Entidades débiles o subordinadas. Entidad que no tiene suficientes atributos para formar una llave primaria.
  - Dependencia por existencia. Cuando la existencia de la entidad X depende de la existencia de la entidad Y. X es subordinada y, Y es dominante.
- Llave Candidato: Conjunto de uno o más atributos que juntos identifican a una entidad como única.
- Llave primaria: Es la llave candidato elegida para identificar a una entidad como única, que debe ser la llave candidato más pequeña posible.
- Llave foránea: Es la llave primaria de una entidad y es la que nos sirve para relacionar a otra entidad.

- Relación: Una relación es una asociación entre entidades. Las relaciones entre entidades pueden ser:
  - Una a una
  - Una a muchas
  - Muchas a una
  - Muchas a muchas

#### **2.4.3.5 Minería de datos.**

La Minería de Datos es un conjunto de técnicas para la inducción de conocimiento útil a partir de una gran cantidad de datos. Por lo que tiene una gran importancia dentro de las bases de datos a diferencia de otras disciplinas como la estadística tradicional, el reconocimiento de patrones, la inteligencia artificial, etc. suele distinguirse de ellas por el campo de aplicación, la minería de datos estudia información acumulada en empresas y otras organizaciones acerca de sus clientes, usuarios, etc. La interacción entre la minería de datos y la información de las organizaciones conlleva a un conocimiento del cual pueden extraerse ventajas como pueden ser mejorar la rentabilidad, el nivel de servicio.

Tradicionalmente, las técnicas de minería de datos se aplicaban sobre información contenida en almacenes de datos. De hecho, muchas grandes empresas e instituciones han creado y alimentado bases de datos especialmente diseñadas para proyectos de minería de datos en las que centralizan información potencialmente útil de todas sus áreas de negocio, etc. No obstante, actualmente está cobrando una importancia cada vez mayor la minería de datos desestructurados como por ejemplo la información contenida en ficheros de texto, en Internet, etc.

##### **2.4.3.5.1 Minería de datos y otras disciplinas análogas.**

Suscita cierta polémica el definir las fronteras existentes entre la minería de datos y disciplinas análogas, como pueden serlo la estadística, la inteligencia artificial, etc. algunos encuentran en la minería de datos una serie de problemas y métodos específicos que la hacen distinta de otras disciplinas.

#### **2.4.3.5.2 Protocolo de un proyecto de minería de datos.**

Un proyecto de minería de datos tiene varias fases necesarias que son, esencialmente:

- Comprensión del objetivo y del problema que se quiere resolver
- Determinación, obtención y limpieza de los datos necesarios
- Creación de modelos matemáticos
- Validación, comunicación, etc. de los resultados obtenidos
- Integración, si procede, de los resultados en un sistema transaccional o similar

#### **2.4.4 Manejadores de Bases de Datos**

##### **2.4.4.1 Oracle.**

Oracle es un sistema de administración de base de datos (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, entre los que se destacan:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Su mayor defecto es su enorme precio, que es de varios miles de euros (según versiones y licencias). Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad.

Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL ó Firebird. Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo Linux.



Oracle surge a finales de los 70 bajo el nombre de Relational Software a partir de un estudio sobre SGBD (Sistemas Gestores de Base De datos) de George Koch. Computer World definió este estudio como uno de los más completos jamás escritos sobre bases de datos. Este artículo incluía una comparativa de productos que erigía a Relational Software como el más completo desde el punto de vista técnico. Esto se debía a que usaba la filosofía de las bases de datos relacionales, algo que por aquella época era todavía desconocido.

La tecnología Oracle se encuentra prácticamente en todas las industrias alrededor del mundo. Oracle es la primera compañía de software que desarrolla e implementa software para empresas 100 por ciento activado por Internet a través de toda su línea de productos: base de datos, aplicaciones comerciales y herramientas de desarrollo de aplicaciones y soporte de decisiones. Oracle es el proveedor mundial líder de software para administración de información, y la segunda empresa de software independiente más grande del mundo.

#### **2.4.4.2 Microsoft SQL Server.**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basada en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo accesan a la información.
- Además permite administrar información de otros servidores de datos

Este sistema incluye una versión reducida y su como es mejor que el de Oracle, pero se rige bajo los mismos esquemas de Microsoft, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase ASE.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa una potente base de datos (Microsoft SQL Server) con un entorno de desarrollo cómodo y de alto rendimiento (VBA Access) a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para la mayoría de las plataformas de desarrollo, incluyendo.NET.

#### **2.4.4.3 MySQL.**

MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad. Aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.

- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Para agregar, acceder a, y procesar datos guardados, usted necesita un administrador como MySQL Server. Dado que los servidores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y que no puede hacer con el software en diferentes situaciones.

#### **2.4.4.5 Microsoft Access.**

Microsoft Access es un sistema de gestión de bases de datos creado y modificado por Microsoft (DBMS) para uso personal o de pequeñas organizaciones. Es un componente de la suite Microsoft Office aunque no se incluye en el paquete *básico*. Para bases de datos de gran calibre (en cuanto a volumen de datos o de usuarios) es recomendable usar otros sistemas como Microsoft SQL Server, MySQL, Postgres u Oracle. Su principal función es ser una potente base de datos, capaz de trabajar en sí misma o bien con conexión hacia otros lenguajes de programación. Pueden realizarse consultas directas a las tablas contenidas mediante instrucciones SQL. Internamente trae consigo el lenguaje Visual Basic for Application (VBA) el cual es similar en forma a VB6.

Permite el ingreso de datos de tipos: Numéricos, Texto, Fecha, Sí/No, OLE, Moneda, Memo y Boolean. Pueden desarrollarse aplicaciones completas basadas en Microsoft Access, pues trae consigo las herramientas necesarias para el diseño y desarrollo de formularios para el ingreso y trabajo con datos e informes para visualizar e imprimir la información requerida.

Permite el desarrollo de pequeñas aplicaciones autónomas formadas por formularios Windows y código VBA (Visual Basic para Aplicaciones). Una posibilidad adicional es la de crear ficheros con bases de datos que pueden ser consultados por otros programas. Entre las principales funcionalidades de Access se encuentran:

- Crear tablas de datos indexadas.
- Relaciones entre tablas (creación de bases de datos relacionales).
- Creación de consultas y vistas.
- Formularios.
- Informes.
- Interacción con otras aplicaciones que usen VBA (resto de aplicaciones de Microsoft Office, Autocad, etc.).

Es un software de gran difusión entre pequeñas empresas (PYMES) cuyas bases de datos no requieren de excesiva potencia, ya que se integra perfectamente con el resto de aplicaciones de Microsoft y permite crear pequeñas aplicaciones con unos pocos conocimientos de programación.

Entre sus mayores inconvenientes figuran que no es multiplataforma, pues sólo está disponible para sistemas operativos de Microsoft, y que no permite transacciones. Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta críticos o muchos accesos simultáneos a la base de datos.

#### **2.4.4.6 PostgreSQL.**

PostgreSQL es un sistema administrador de bases de datos relacionales (RDBMS) y está considerado como la base de datos de código abierto más avanzada del mundo.

Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. La siguiente es una breve lista de algunas de esas características, a partir de PostgreSQL 7.1.x,

- **Objeto-Relacional:** aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.
- **Altamente Extensible:** soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Soporte SQL Comprensivo:** soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- **Integridad Referencial:** soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **API Flexible:** la flexibilidad de API de PostgreSQL ha permitido a los vendedores proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

MVCC, ó control de concurrencia Multi-Versión (Multi-Versión Concurrency Control), es la tecnología de PostgreSQL usa para evitar bloqueos innecesarios. Si alguna vez ha usado algún DBMS con capacidades SQL, tal como MySQL ó Access, probablemente habrá notado que hay ocasiones en las que una lectura tiene que esperar para acceder a la información de la base de datos. La espera es provocada por usuarios que están escribiendo en la base de datos. Resumiendo, el lector está bloqueado por los escritores que están actualizando registros.

Mediante el uso de MVCC, PostgreSQL evita este problema por completo. MVCC está considerado mejor que el bloqueo a nivel de fila porque un lector nunca es bloqueado por un escritor. En su lugar, PostgreSQL mantiene una ruta a todas las transacciones realizadas por los usuarios de la base de datos. PostgreSQL es capaz entonces de manejar los registros sin necesidad de que los usuarios tengan que esperar a que los registros estén disponibles.

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

La característica de PostgreSQL conocida como Write Ahead Logging incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en hipotético caso de que la base de datos se caiga, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso en el caso de caída, ya que en cualesquiera cambios que no fueron escritos en la base de datos pueden ser recuperados usando el dato que fue previamente registrado. Una vez que el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando cayó la base de datos.

#### 2.4.5 Análisis para la implementación del sistema.

Después de dar un bosquejo y ver las características de varios lenguajes de programación podemos observar las diferencias de estos para poder tomar la mejor decisión para el proyecto:

	<b>JSP</b>	<b>ASP</b>	<b>PHP</b>	<b>JAVA</b>	<b>SERVLET</b>
<b>Servidores web</b>	Server Apache Server Tomcat	IIS Personal Web Server	IIS Server Apache		
<b>Plataformas HW</b>	Disponible en todas las plataformas (multiplataforma)	Plataformas compatibles con Win32	Disponible en todas las plataformas (multiplataforma)	Disponible en todas las plataformas (multiplataforma)	Disponible en todas las plataformas (multiplataforma)
<b>Portabilidad</b>					
<b>Uso de scripts</b>	Uso de JavaScript	Uso de VBScript y Jscript	Uso de JavaScript	Uso de JavaScript	Uso de JavaScript
<b>Seguridad</b>	Modelo de seguridad de Java	Modelo de seguridad basado en la arquitectura		Modelo de seguridad de Java	Modelo de seguridad de Java

		de NT			
<b>Acceso a bases de datos</b>	A través de JDBC	A través de ADO (Active Data Object)	A través de dll	A través de JDBC	A través de JDBC
<b>Personalización de tags</b>	Se puede ampliar a través del uso de librerías	No puede ampliarse	Se puede ampliar a través del uso de librerías	Se puede ampliar a través del uso de librerías	Se puede ampliar a través del uso de librerías
<b>Tipo de Licencia</b>	GNU(Licencia Publica General)	A través de JDBC	GNU(Licencia Publica General)	GNU(Licencia Publica General)	GNU(Licencia Publica General)

Tabla 2.1. Comparativo de lenguajes de programación

Ambos necesitan un programa que los contenga, y sea el que envíe efectivamente páginas web al servidor, y reciba las peticiones, las distribuya entre los servlets, y lleve a cabo todas las tareas de gestión propias de un servidor web. Mientras que servidores como el Apache están especialmente pensados para páginas web estáticas CGI, y programas ejecutados por el servidor, tales como el PHP, hay otros servidores específicos para servlets y JSP's llamados *contenedores de servlets (servlet containers)* o *servlet engines*. Los principales son los siguientes

La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web

#### 2.4.5.1 Herramientas de desarrollo.

Para el desarrollo del sistema además del software y el hardware, también haremos uso de otras herramientas, para un mejor desarrollo y para poder dar presentación a los reportes que generará el sistema, así como para poder manejar la información de los requerimientos.

### **2.4.5.1.1 UML.**

UML (Unified Modeling Language) es un conjunto de herramientas, que nos permite modelar (analizar y diseñar) sistemas orientados a objetos, un UML esta dividido en lo siguiente:

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

Se pueden seleccionar cualquiera de las herramientas anteriores para el desarrollo de sistemas seleccionando las más adecuadas a nuestras necesidades. Un UML no es un método de desarrollo, pero es una herramienta que nos ayuda a pasar del análisis al diseño y de este al código. Lo que nos lleva a producir código a partir de las especificaciones que se planteen para el sistema. UML al no ser un método de desarrollo es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

Algún software para trabajar con UML pueden ser los siguientes:

- Rational Rose.
- StarUML.

### **2.4.5.1.2 IReport.**

La herramienta iReport es un constructor ó diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Esta herramienta nos permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport es una herramienta gráfica, y es OpenSource, lo que la hace una de las más utilizadas para Java. Los datos para



imprimir pueden ser extraídos por varios métodos como son por JDBC, TableModels, JavaBeans, XML, etc.

La lista siguiente describe algunas de las características importantes de iReport:

- 100% escrito en JAVA y además OPENSOURCE y gratuito.
- Maneja el 98% de las etiquetas de JasperReports
- Browser de la estructura del documento.
- Soporta JDBC.
- Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
- Incluye Wizard's (asistentes) para crear automáticamente informes.
- Tiene asistentes para generar los subreportes
- Tiene asistentes para las plantillas.
- Facilidad de instalación

#### **2.4.5.1.3 Eclipse.**

El Proyecto Eclipse es un proyecto de desarrollo de software de código abierto dedicado a proporcionar una plataforma industrial robusta, con amplias características y para el desarrollo. Está compuesto de tres subproyectos: la Plataforma Eclipse, la Java Development Tool y el Plug-in Development Environment. El éxito de la Plataforma Eclipse depende de cómo sea capaz de admitir una amplia gama de herramientas de desarrollo para reproducir lo mejor posible las herramientas existentes en la actualidad. Eclipse lo forman el núcleo, el entorno de trabajo (Workspace), el área de desarrollo (Workbench), la ayuda al equipo (Team support) y la ayuda o documentación (Help) los cuales se describen a continuación:

- Núcleo: su tarea es determinar cuales son los plug-ins disponibles en el directorio de plug-ins de Eclipse. Cada plug-in tiene un fichero XML manifest que lista los elementos que necesita de otros plug-ins así como los puntos de extensión que ofrece. Como la cantidad de plug-ins puede ser muy grande, solo se cargan los necesarios en el momento de ser utilizados con el objeto de minimizar el tiempo de arranque de Eclipse y recursos.

- Entorno de trabajo: maneja los recursos del usuario, organizados en uno o más proyectos. Cada proyecto corresponde a un directorio en el directorio de trabajo de Eclipse, y contienen archivos y carpetas.
- Interfaz de usuario: muestra los menús y herramientas, y se organiza en perspectivas que configuran los editores de código y las vistas. A diferencia de muchas aplicaciones escritas en Java, Eclipse tiene el aspecto y se comporta como una aplicación nativa. No está programada en Swing, sino en SWT (Standard Widget Toolkit) y Jface (juego de herramientas construida sobre SWT), que emula los gráficos nativos de cada sistema operativo. Este ha sido un aspecto discutido sobre Eclipse, porque SWT debe ser portada a cada sistema operativo para interactuar con el sistema gráfico. En los proyectos de Java puede usarse AWT y Swing salvo cuando se desarrolle un plug-in para Eclipse.
- Ayuda al grupo: este plug-in facilita el uso de un sistema de control de versiones para manejar los recursos en un proyecto del usuario y define el proceso necesario para guardar y recuperar de un repositorio. Eclipse incluye un cliente para CVS.

Al igual que el propio Eclipse, el componente de ayuda es un sistema de documentación extensible. Los proveedores de herramientas pueden añadir documentación en formato HTML y, usando XML, definir una estructura de navegación.

### **3. Análisis de requerimientos del sistema de Información Curricular para la Facultad de Ingeniería.**

#### **3.1 Los elementos del sistema.**

La meta primaria en la fase de requerimientos del sistema se basa en identificar y documentar las necesidades de los usuarios finales del sistema, lo cual implica recabar grandes volúmenes de información.

En el caso específico de la presente investigación, los usuarios finales es la planta académica de la Facultad de Ingeniería y la necesidad se resume en una sola, que la población mencionada pueda proporcionar sus datos curriculares en una forma estandarizada.

Se creara un sistema para satisfacer dichas necesidades nos lleva a definir tres módulos donde se engloban las personas que requerirán interactuar con el sistema, con el fin de suministrar la información requerida, administrarla y/o validarla. A continuación dichos módulos aparecen ilustrados en la siguiente figura:



Figura 3.1. Módulos del sistema

El primero de ellos, es el módulo de Administrador, donde se concentra la tarea de dar de alta a los nuevos usuarios del sistema, conceder permisos ó simplemente realizar revisiones sobre el funcionamiento del sistema; dada su ubicación, la parte más alta de la pirámide, nos induce a ubicar una cantidad mínima de usuarios interactuado con el sistema.

En el siguiente nivel se ubican las dependencias, encargadas de un sector específico. Para esta sección, la generación de reportes, la revisión y validación de la información generada por las personas adjuntas a cada dependencia son las tareas preestablecidas, lo cual quiere decir que esta sección servirá como un filtro externo al sistema, ya que aun cuando se puede realizar una serie de validaciones con relación a la información capturada por los académicos de la Facultad de Ingeniería, la última palabra de la autenticidad de dicha información será dictaminada por la dependencia a la que están adscritos.

Por último, ubicamos a la planta académica dentro de la Facultad de Ingeniería, quienes motivaron en principio el desarrollo de este sistema, y a partir de los cuales se definió, previo estudio, más del 90% de los requerimientos del mismo, mientras que de las dependencias surgió la necesidad de tener la información estandarizada de cada uno de sus académicos, pero son éstos quienes se ven con la obligación ó tarea de generar dicha información.

A continuación, basándonos en la necesidad de cada una de las dependencias de contar con toda la información pertinente de su planta académica necesaria para el evaluó de la misma y al mismo tiempo cubrir los aspectos generales de la información curricular necesaria para identificar al académico, así como los aspectos detallados que lo distinguen, identificamos tres de los requerimientos básicos que se deberán cubrir como mínimo para la generación de un currículum vitae; dichos requerimientos son descritos a continuación:

- Módulo para Datos Personales
- Módulo para la Información de su formación académica
- Módulo para recabar Información sobre su trayectoria laboral
- Módulo asociado a los premios, estímulos y otros reconocimientos recibidos<sup>1</sup>

Establecidos los cuatro módulos anteriores comenzamos por destacar a cada uno de los elementos que los componen, con el fin de dar un marco general de los aspectos que creemos deben de cubrirse dentro de un currículum

---

<sup>1</sup> El cuarto módulo permitirá a la facultad así como a sus dependencias identificar los programa de apoyo o estímulo por parte de la UNAM otorgados a los académicos, información con que se pretende dar seguimiento a ciertos procesos internos como el otorgamiento de nuevos estímulos.

#### Datos Personales:

- Información personal.
- Direcciones tanto particular como laborales.

#### Formación Académica:

- Información general de sus títulos y grados obtenidos.
- Estancias, comisiones y años sabáticos.
- Cursos y/o diplomados.
- Participación en proyectos de investigación.
- Cualquier conocimiento extra que le permita realizar la labor docente.

#### Trayectoria Laboral:

- Trayectoria laboral dentro de la UNAM
  - Especificando que nombramientos administrativos y/o académicos ha tenido dentro de su estancia en la UNAM.
- Docencia y formación de recursos humanos.
  - Donde podrá distinguirse su capacidad para coordinar grupos de investigación o trabajo, fomentar la formación de personal académico o simplemente fungiendo como asesor de algunos estudiantes para obtener un grado o título a través de las modalidades de titulación.
- Actividades Profesionales
  - Aquí buscamos plasmar el desarrollo profesional que ha tenido el académico dentro del sector privado
- Productividad Académica
  - Buscando plasmar cualquier enlace con la labor editorial y la producción de material bibliográfico producto de alguna investigación o trabajo que conlleve desde la publicación de algún artículo en alguna revista hasta la edición de un libro.
- Otras actividades
  - Dado que la labor docente no se fija sólo en el ámbito laboral dentro o fuera de la UNAM, vimos la necesidad de agregar un espacio donde el académico pudiera plasmar su participación en asociaciones, actividades de investigación u otras

actividades como conferencias, ponencias, etc., con el fin de vislumbrar un marco más amplio de sus actividades.

Información de Evaluación para las dependencias:

- Premios y distinciones. Con el fin de conocer la cantidad de reconocimientos obtenidos por el académico a lo largo de su vida laboral y docente.
- Programas y estímulos. Aquí se incluirán todos los programas y estímulos de los que ha sido participe el académico en la UNAM.

Una vez definidos los elementos requeridos para captar la información curricular necesaria de la planta académica de la facultad de Ingeniería, procederemos a establecer el manejo de dicha información.

### **3.2 Dominio y manejo de la información.**

En un plano principal, se cuenta con la interacción entre los distintos tipos de usuarios con el sistema, cómo acceden al sistema e interactúan con él para realizar el registro de su información o para realizar su consulta, todo ello podemos describirlo perfectamente en la figura 3.2.

En primera instancia definiremos una entrada común a cualquier nivel, llámese administrador, dependencia o académico, con el fin de que la presentación inicial sea única, permitiendo de esta forma mantener la seguridad y el control que nos brinda el poseer una sola entrada al sistema, mientras que el sistema decidirá de forma transparente a donde deberá ser redireccionado el usuario que está accediendo, con base a un nivel jerárquico.

Para un segundo plano se definen tareas específicas a cada nivel, que van desde el ingreso de la información curricular, permitiendo la generación de un reporte, hasta la administración de los académicos, pasando por búsquedas especializadas y reportes de las mismas.

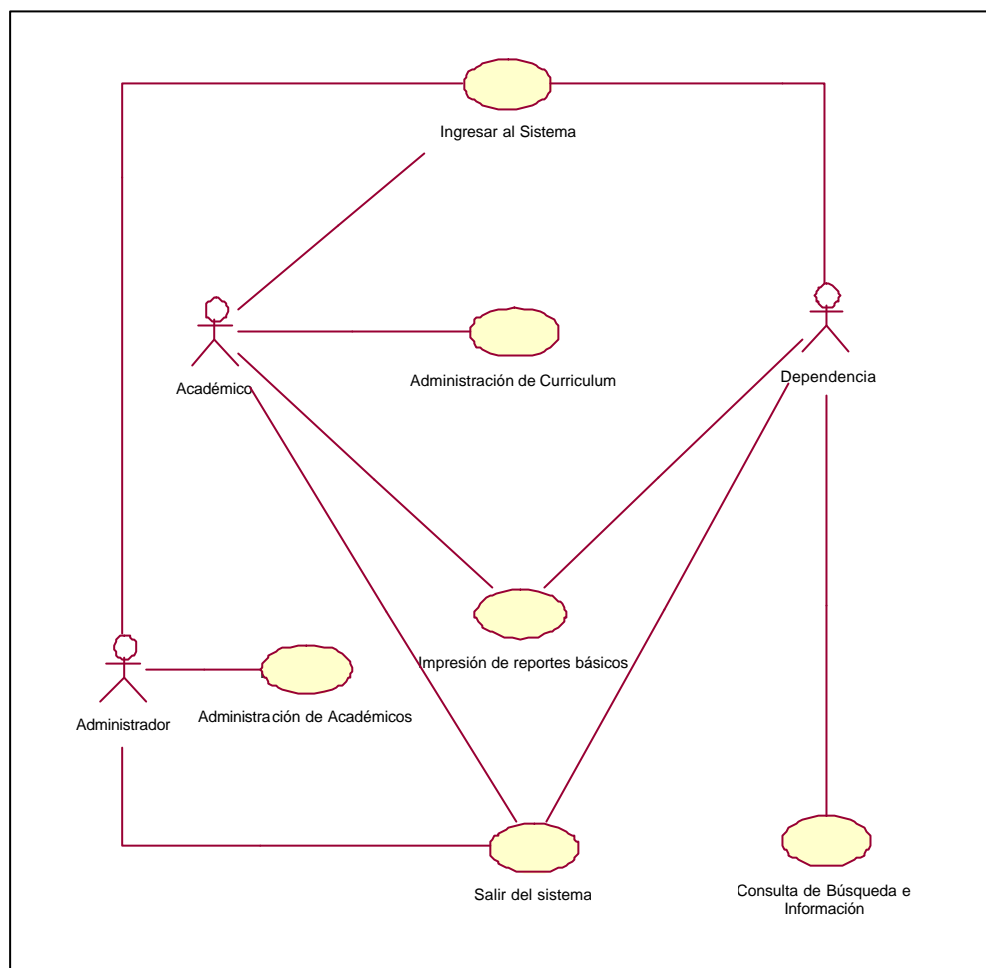


Figura 3.2. Interacción de Usuario con el sistema.

Una vez establecidas como interactuarán los distintos usuarios con el sistema, podemos establecer las correspondencias entre cada tipo de usuario y el sistema, comenzando por académico, el cual se encargará de proporcionar la información correspondiente a los módulos definidos con anterioridad, datos personales, formación académica, trayectoria laboral en la UNAM así como en el sector privado, además de la información relacionada a premios, estímulos y distinciones otorgados por la UNAM u otras dependencias e Instituciones, figura 3.3.

Además, cada módulo representado en el diagrama anterior, deberá de contar con tres bloques principales que serán: alta, consulta y modificación de los datos proporcionados por el académico. Aunado a esto, tendremos la posibilidad de generar un reporte ya sea general o de bloques específicos en un formato de fácil manejo.



Figura 3.3. Correspondencia tipo de usuario - sistema.

En el caso de las dependencias, se les brindará la opción de poder realizar búsquedas específicas, relacionadas a su sector correspondiente, pudiendo acceder a toda la información proporcionada por la planta académica que se encuentra bajo su cargo, además de poder generar archivos en un formato específico de dicha información con el fin de poder facilitar su manejo y uso.



En la figura 3.4 especificamos los bloques de información sobre el académico de los cuales la dependencia podrá generar sus reportes de forma directa o a través de una búsqueda personalizada.

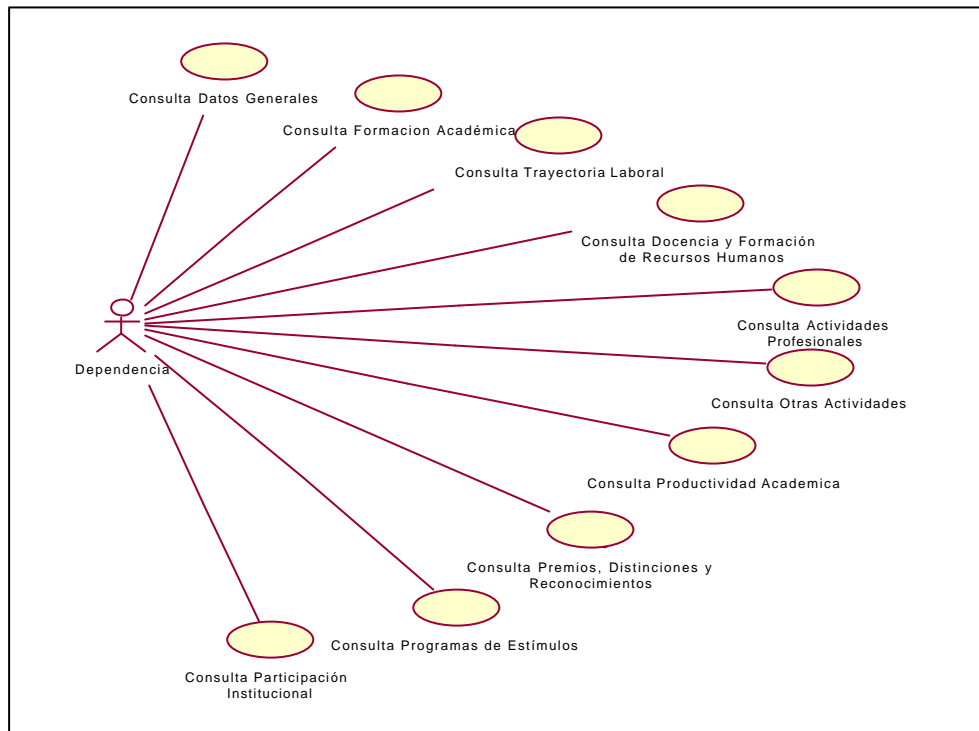


Figura 3.4. Información sobre académico.

Como puede observarse, los módulos definidos en un principio como requisitos básicos para la generación del currículum, donde se alberga la información que la planta académica deberá capturar, es la base a partir de la cual la dependencia podrá generar sus reportes así como realizar la explotación de la información para su máximo rendimiento.

Por último, el administrador del sistema; quien se encargará de dar de alta y baja a los usuarios del sistema, en el caso de que el académico requiera de la modificación o corrección de sus datos de entrada, éste deberá solicitarlo directamente al administrador de la base de datos, ya que la asignación de un login y password a través del el sistema representa una falla de seguridad.

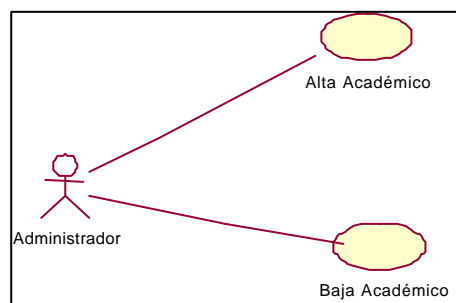


Figura 3.5. Interacción entre usuarios.

### **3.3 Flujo y estructura de la información.**

La información manipulada por este sistema y el flujo que ha de seguirse para acceder a ella con el fin de agregar nuevos datos o modificar los ya existentes; será dispuesta considerando los módulos previamente definidos:

#### **3.3.1 Caso del Personal Académico de la Facultad.**

- Ingresar al Sistema
- Módulo Académico
  - Datos Generales
  - Formación Académica
  - Trayectoria Laboral
  - Docencia y formación de Recursos Humanos
  - Actividades Profesionales
  - Otras Actividades
  - Productividad Académica
  - Premios, Distinciones y Reconocimientos
  - Programas de Estímulos
  - Participación Institucional
  - Presentación del Currículo

Para el caso de los primeros diez bloques, cada bloque contará con los siguientes módulos:

- Alta
- Actualización
- Consulta

Y en el caso del bloque para la generación del currículum vitae, sólo existirá la opción de generar un reporte parcial o total de la información en algún formato de fácil manejo.

### 3.3.2 Caso de la Dependencia.

- Ingresar al Sistema
- Módulo de Dependencia
  - Consulta por índices y tabuladores de:
    - Datos Generales
    - Formación Académica
    - Trayectoria Laboral
    - Docencia y formación de recursos Humanos
    - Actividades Profesionales
    - Otras Actividades
    - Productividad Académica
    - Premios, Distinciones y Reconocimientos
    - Programas de Estímulos
    - Participación Institucional
  - Consulta Personalizada:
    - Datos Generales
    - Formación Académica
    - Trayectoria Laboral
    - Docencia y formación de recursos Humanos
    - Actividades Profesionales
    - Otras Actividades
    - Productividad Académica
    - Premios, Distinciones y Reconocimientos
    - Programas de Estímulos
    - Participación Institucional
  - Generar un reporte parcial o total de la información curricular en un formato de fácil manipulación para ambos casos.

### **3.3.3 Caso del Administrador.**

- Ingresar al Sistema
- Módulo de Administrador
  - Alta de Usuarios
  - Baja de Usuarios

## **3.4 Requerimientos de Herramientas (Software y Hardware)**

Bajo las consideraciones producidas por el estudio previo, podemos establecer las condiciones que permitirán al sistema trabajar de manera adecuada es necesario, para ello se requiere se garanticen al menos las siguientes condiciones físicas (Hardware).

### **3.4.1 Hardware:**

- Servidor
  - Procesador de arquitectura reciente
  - Conexión al Internet
  - Memoria al menos de 512 MB
  - Espacio libre en disco al menos de 40 GB
- Estación cliente
  - Procesador con velocidad mayor a 140 MHz
  - Conexión a Internet
  - Memoria al menos de 64 MB
  - Espacio libre en disco al menos de 50 MB

### **3.4.2 Software**

- Servidor
  - PostgreSQL 7.4.1 o superior
  - JDK 1.4.2 o superior.

- Apache 1.3.29 ó superior
- Apache TomCat Versión 5.30 o superior
- Linux ó Windows
- JDBC para PostgreSQL 7.4.1 ó superior
- iReport 0.5.0 o superior
- Cliente
  - Navegadores (Netscape Navigator 4.x o superior, Internet Explorer 4.x o superior, Mozilla, Opera).
- Administrador
  - PostgreSQL 7.4.1 o superior
  - JDK 1.4.2 o superior
  - Apache 1.3.29 ó superior
  - Linux o Windows. O superior
  - JDBC para PostgreSQL 7.4.1 ó superior
  - iReport 0.5.0 o superior
- Comunicaciones
  - Soporte para redes basadas en protocolo TCP/IP y SSL.

Mientras que para el diseño e implementación del sistema los requerimientos de software se reducen a las siguientes herramientas dentro del área de desarrollo:

- Rational Rose: Empleado para el análisis y diseño del flujo de información.
- Dia: Utilizado para realizar la representación gráfica del modelado Entidad – Relación, correspondiente al manejo de la información e interacción entre el sistema y los diferentes procesos de manejo de información (alta, modificación y consulta).
- DreamWeaver: Herramienta que facilita la generación de GUI's de forma rápida y amigable, lo cual permitirá un rápido diseño y construcción de la Interfaz, en comparación del uso del bloc de notas que haría de esta tarea algo exagerada.
- JDK (Java Developed Kid): Las bases del sistema se fundan principalmente en su capacidad de portabilidad, la cual es heredada por el lenguaje de programación seleccionado para la implementación del sistema y ambiente de ejecución.

- Photoshop, Corel Draw, GIMP. Como Herramientas adicionales empleadas para generar la parte gráfica de nuestro sistema.
- Eclipse como IDE de programación para java.

Por otro lado, para el área de implementación requeriremos:

- JDK(Java Developed Kid): En esencia la Máquina virtual de Java, herramienta en la que se apoya el servidor de JSP's Tomcat.
- Tomcat: Servidor dedicado para aplicaciones basadas en Java, es el encargado de recibir las peticiones para servir los JSP's y Servlets, con el fin de mantener una interacción entre el usuario y el sistema.
- Apache: En nuestro caso, bastaría Tomcat para dar servicio a las aplicaciones basadas en JSP's, sin embargo, por la magnitud del sistema, y el alcance que se pretende obtener, es requerido usar Apache como un desfogue para reducir la carga de trabajo de Tomcat, ya que si bien éste es un buen servidor, no cuenta con las capacidades y alcance que Apache ofrece en lo que respecta a aplicaciones robustas.
- iReport: para la elaboración de los Reportes requeridos ya sea por las dependencias o los mismos académicos.

### **3.5 Otros Requerimientos**

Otras consideraciones requeridas para el sistema son:

- Portabilidad. Se debe de considerar que el sistema pueda en un momento dado, ser transferido a una diferente plataforma, a ello debemos que nuestro lenguaje de programación elegido para el sistema sea Java, con ello se garantiza un alto porcentaje de compatibilidad en la ejecución del sistema en diferentes plataformas.
- Desempeño. Que la respuesta al usuario por se encuentre dentro de los límites tolerables, dependerá en la mayor parte del servidor, principalmente.
- Usabilidad. Fácil de manejar, por cuanto se usará en su construcción basada en elementos estándares de las aplicaciones, bajo interfaz gráfica. Además de realizar pruebas de usabilidad con usuarios tipo.

- Facilidad de mantenimiento. Considerando que la construcción del sistema estará basada completamente en módulos independientes, se estima que cualquier modificación se podrá ejecutar sin complicaciones.

## **4. Diseño del Sistema de Información Curricular para la Facultad de Ingeniería**

### **4.1 Estructura del sistema**

Para el diseño del sistema se comienzan por identificar los reportes y demás salidas que deben de producirse. Hecho lo anterior se determina con toda precisión los datos específicos para cada reporte y salida. Es común realizar un bosquejo del formato o pantalla que se espera aparezca cuando el sistema este terminado.

Lo anterior se efectúa en papel o en la pantalla de una terminal utilizando para ello algunas de las herramientas automatizadas para el desarrollo de sistemas. En primera instancia identificaremos los datos de entrada, aquellos que serán calculados y los que deben de ser almacenados. Así mismo, describiremos con detalle los procedimientos de cálculo y los datos individuales, seleccionando las estructuras de archivo y los dispositivos de almacenamiento, tales como discos y cintas magnéticas o incluso archivos de papel. Los procedimientos que se escriben indican como procesar los datos y producir las salidas. Los documentos que contienen las especificaciones de diseño representan a éste de muchas maneras (diagramas, tablas y símbolos especiales).

Para el SICFI se ha seleccionado una arquitectura común de los sistemas de información que incluye una interfaz para el usuario y el almacenamiento persistente de datos, esto, coincidente con la arquitectura de tres capas, haciendo que el sistema se constituya de la forma siguiente:

- Capa de Presentación donde ubicaremos los elementos con los que interactuará el usuario, dicho de otra forma, la interfaz gráfica que permite al usuario acceder o ingresar la información al sistema de una forma transparente.
- Capa Lógica de la Aplicación aquí se definen las tareas y reglas que rigen todo el flujo de información a lo largo de su captura, validación y almacenamiento o en su caso la devolución de la información hacia el usuario con el fin de que éste sea quien efectúe los cambios y/o correcciones pertinentes. Esta capa está constituida por los siguientes elementos:



- Objetos del Dominio del Problema, son las clases donde se representan los conceptos del sistema.
  - Servicios aquí nos referimos a aquellos objetos de dominio no relacionados con el problema empleados para prestar el servicio de soporte, como la interacción con la base de datos, los reportes, las comunicaciones, etc.
- Capa de Almacenamiento, donde se ubicarán los mecanismos de almacenamiento persistente para nuestro caso se trata de una base de datos relacional.

En la siguiente figura 4.1 podemos observar la distribución de los elementos que conforman a cada una de nuestras capas, pudiéndose observar en una forma más detallada la capa de servicios, donde se ubican los módulos destinados para interactuar con los tres diferentes tipos de usuarios, Académico, Dependencia y Administrador, así como el módulo destinado a la interacción con la base de datos.

En la misma figura, para la capa de almacenamiento se puede observar una serie de bloques que contendrán en si mismos una serie de tablas interrelacionadas con una serie de catálogos, destinados a agilizar el manejo de la información, esto con base en la definición de minería de datos.

Ya definido el contenido de cada una de nuestras capas, podemos observar como el proceso de captura y presentación de la información es realizada en la capa de presentación, en ella se solicitará al usuario toda la información establecida en la sección del flujo y estructura de la información<sup>2</sup>, la forma en que realizaremos el diseño de la interfaz al usuario, definir cómo y cuándo se realizará el cálculo de la información ingresada al sistema y seleccionar las estructuras de archivo y los dispositivos de almacenamiento lo definiremos en los capítulos siguiente.

---

<sup>2</sup> Información o datos específicos solicitados al académico que el sistema maneja.

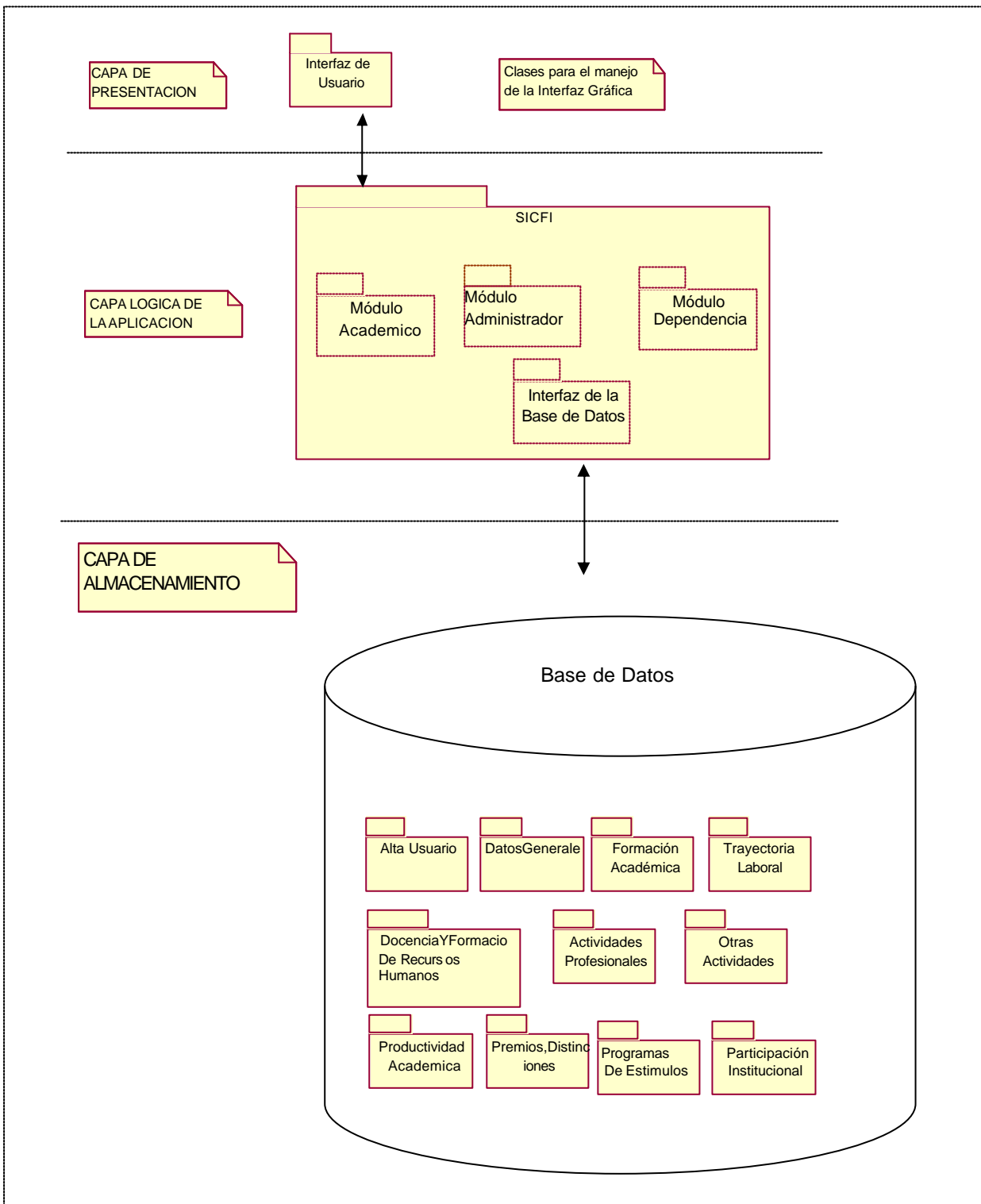


Figura 4.1. Estructura del sistema.

## 4.2 Diseño de la interfaz gráfica

En primera instancia diseñamos una ficha estándar de los currículos, considerando todas las variables identificadas en el análisis de requerimientos, definiendo qué apartados y/o variables deberán ser llenados por los académicos y cuáles deberán estar bajo el control, dándoles opciones perfectamente definidas ya sea mediante listas desplegables o casillas de opción múltiple. De esta forma, definimos los siguientes bloques identificados en los diagramas siguientes, dichos bloques describen a grandes rasgos la interacción con el usuario así como la captura o presentación de la información.

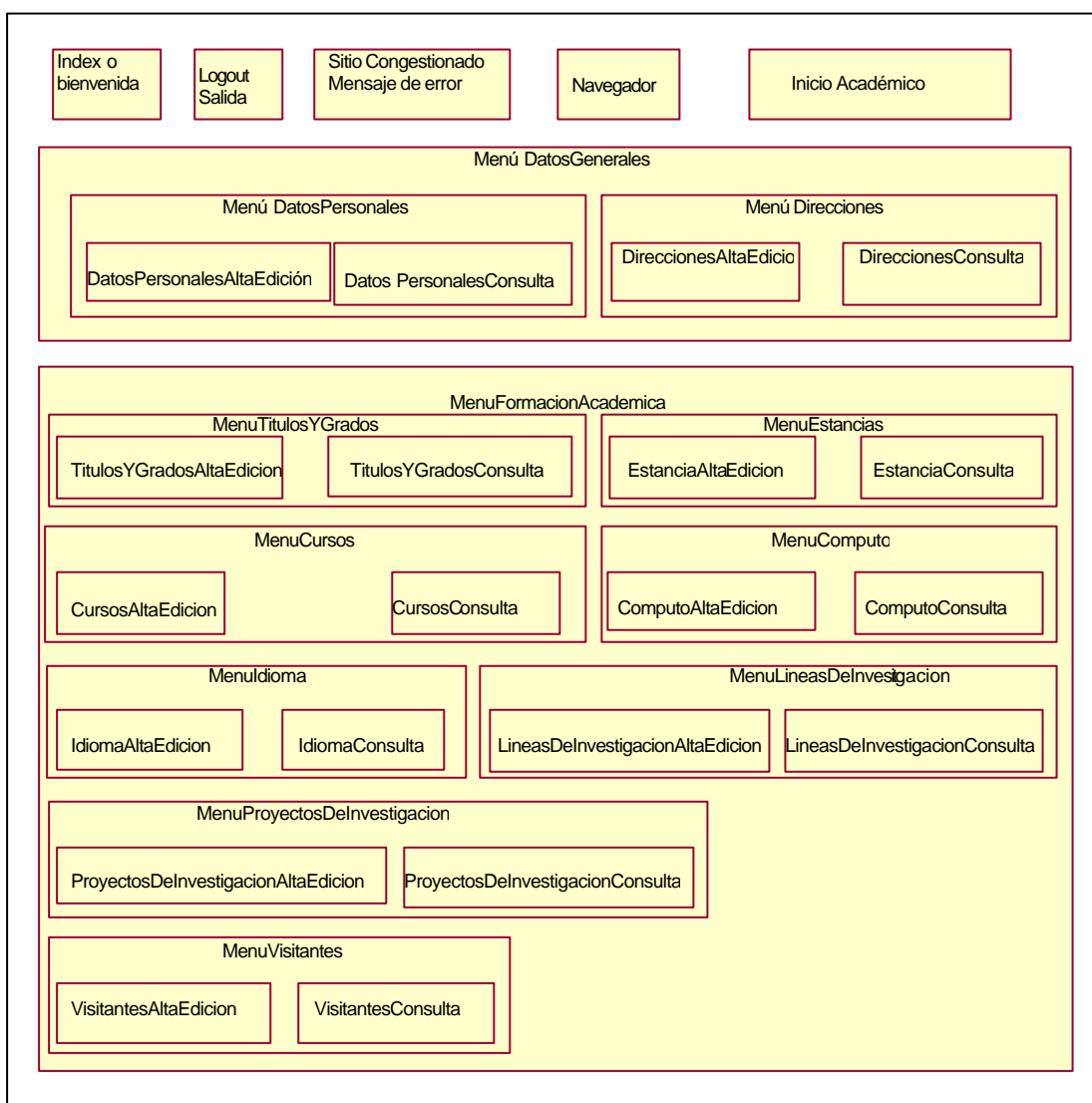


Figura 4.2. Interfaz gráfica académico

Para los diagramas de la parte superior tenemos en primera instancia la sección dedicada al index, no es solo la entrada al sistema; al mismo nivel tenemos la salida del sistema, el manejo de errores para el servidor y una ventana inicial de bienvenida para el académico, también contaremos con un header, con la funcionalidad de navegador, éste se mantendrá presente en los diferentes módulos con el fin de facilitar la movilidad del usuario por el sistema, contando con una serie de etiquetas para identificar en forma clara a cada uno de los bloques descritos a continuación.

Para el módulo siguiente, ubicamos los datos generales del académico, el cual contendrá tanto los datos personales, así como las direcciones o domicilios asociados al mismo. A partir de este módulo y los módulos consecutivos consideraremos que cada elemento estará constituido por una sección de Alta y/o modificación, así como un componente destinado para la tarea de consulta.

El módulo que le precede es el destinado a la trayectoria Académica, donde podemos destacar entre otras cosas los títulos y grados que el académico ha ido obteniendo.

En el siguiente diagrama registramos los datos relacionados con la trayectoria Laboral, en donde consideramos tanto los nombramientos académicos, así como los nombramientos administrativos dentro de la UNAM-

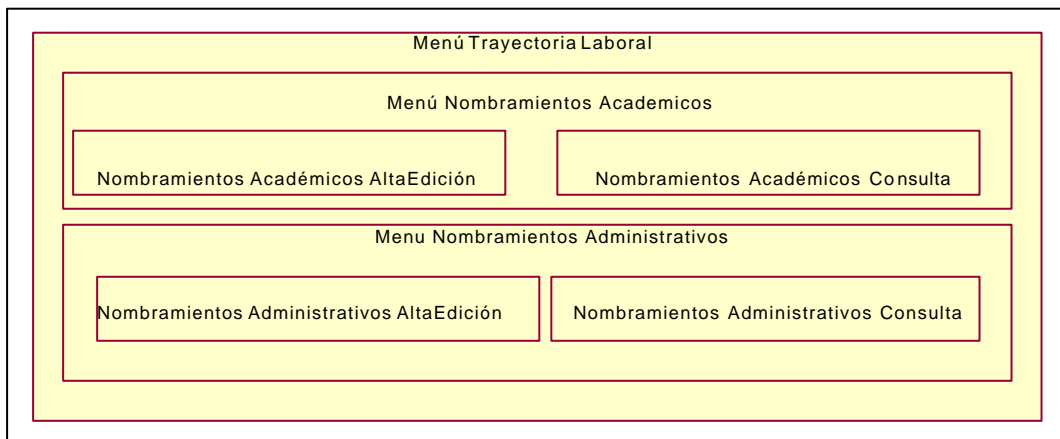


Figura 4.3. Trayectoria laboral

En el siguiente bloque se consideran lo que es la formación de recursos humanos y docencia, donde se capturarán los datos relacionados con la formación de grupos de trabajo para la realización de proyectos de investigación o desarrollo e implementación de los mismos, además de capturar la participación como docente así como las actividades como director de Tesis.

En un segundo plano tenemos las actividades profesionales, en donde recabaremos la información relacionada con el ámbito laboral dentro de la iniciativa privada.

Un poco más abajo, contamos con una sección dedicada a la captura de información relacionada con otras actividades contenidas en el ámbito docente como lo son los actos académicos, la participación en algunas asociaciones o cuerpos colegiados, así como la prestación de servicios relacionados o inferidos de manera directa en el ámbito docente.

Más adelante contaremos con el bloque dedicado a la productividad Académica, de la cual se desprende la elaboración de todo tipo de material bibliográfico, que puede desprenderse de un artículo o cita publicado en alguna revista de carácter cultural o científico, a partir de la labor editorial, la cual puede ir desde la participación en la elaboración de un capítulo para algún libro, traducción de textos, edición, etc.

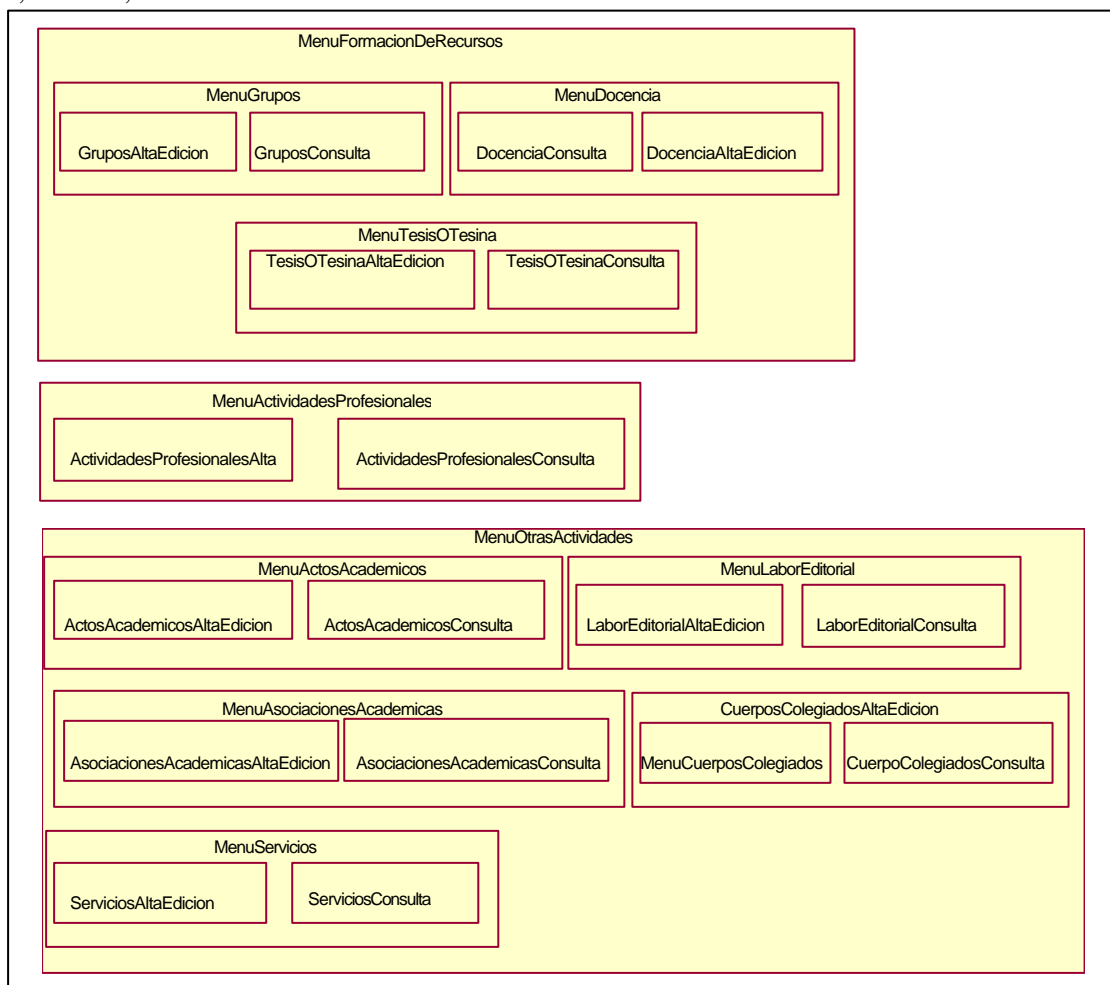


Figura 4.4. Formación de recursos

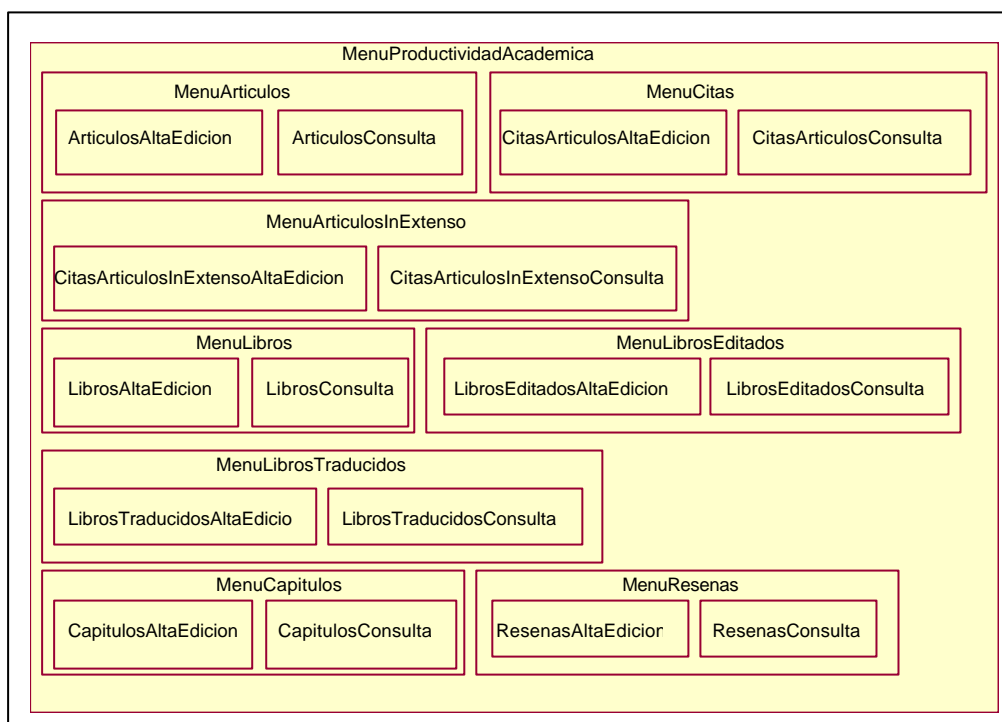


Figura 4.5. Productividad académica

Por último, tendremos los bloques destinados a los premios otorgados por la institución a los académicos, los programas de estímulos de los cuales ha sido participe, además de su colaboración en el área institucional.

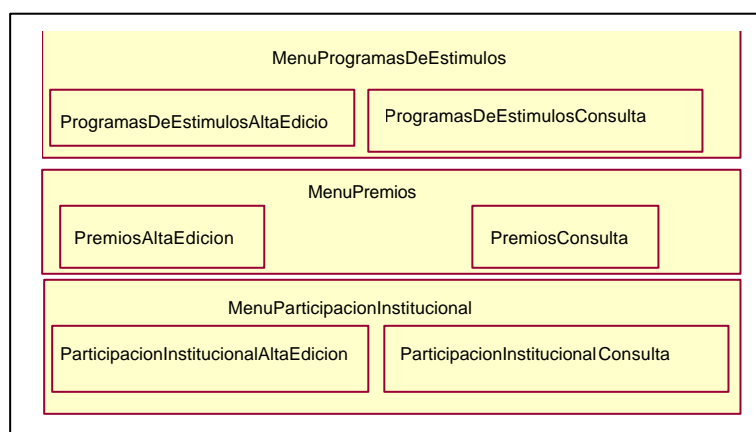


Figura 4.6. Estímulos

En segundo plano, tenemos a las dependencias encargadas de obtener reportes de la información generada por su planta académica, en este caso las opciones permitidas constituyen una ficha con la información específica que requieran, así como la posibilidad de generarles un reporte partir de una búsqueda personalizada.

Dependencia trabajará con una interfaz de inicio similar a la del académico en lo que respecta a la interacción con el sistema, pero en la parte funcional, el contenido presentado dentro de los dos mostrados en la figura siguiente tan solo contendrá elementos que permitan la generación de los respectivos reportes solicitados.

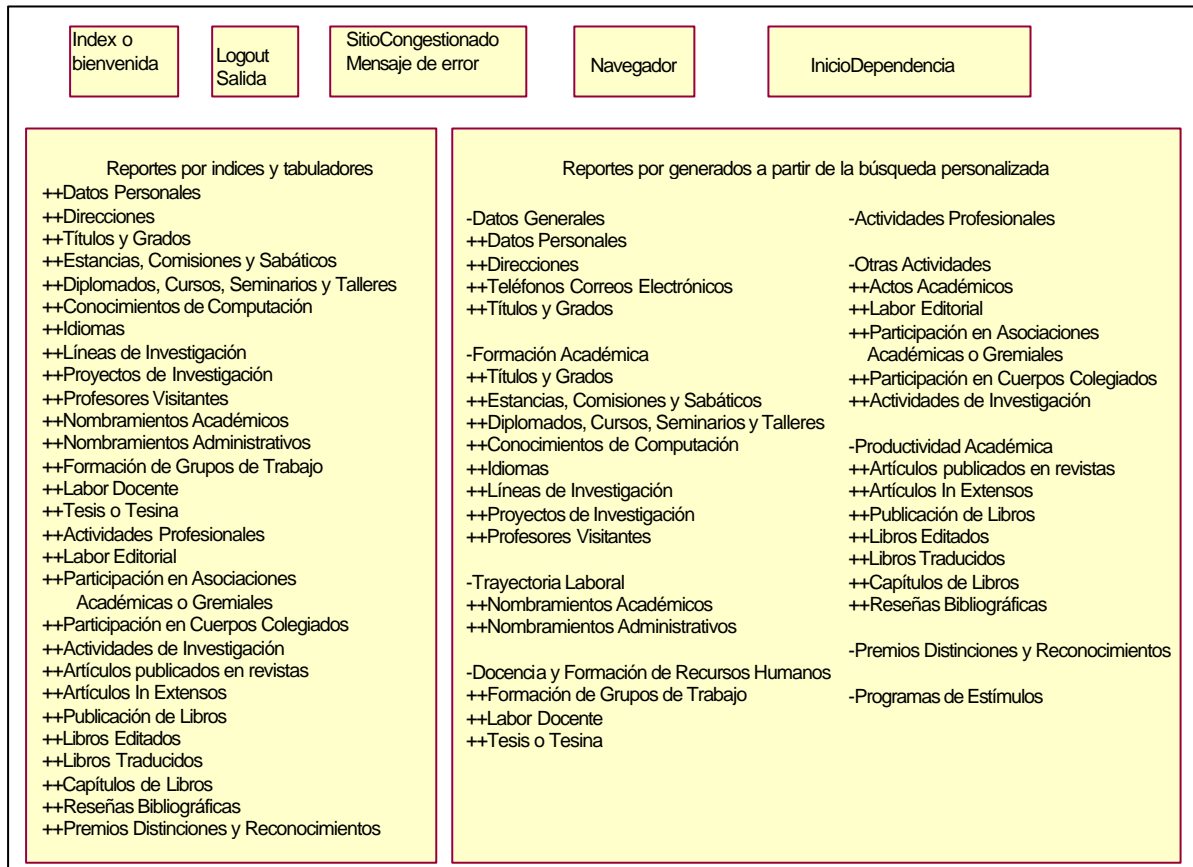


Figura 4.7. Interfaz gráfica dependencia

Al final, definimos las funciones del administrador del sistema mediante el siguiente diagrama, procurando establecer las tareas mínimas para el procesamiento de los datos. Nuevamente el sistema de navegación del sistema, la entrada, salida, presentación y manejo de errores se presentan de forma similar que en los casos del académico y de la dependencia, además de lo anterior, ubicamos dos módulos destinados para el alta y la baja de usuarios, funciones únicas del administrador en lo que respecta a su trabajo con el sistema vía Web.

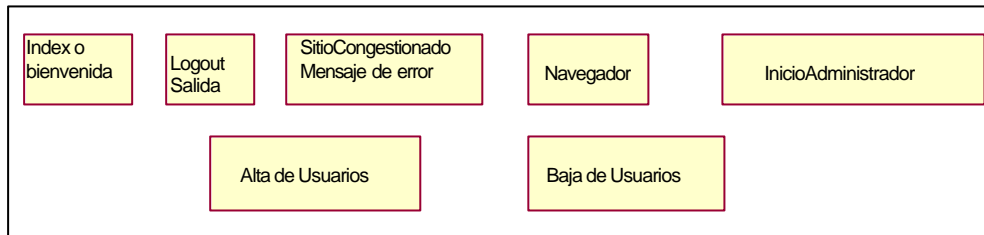


Figura 4.8. Interfaz grafica administrador

### 4.3 Manejo de la información

La interacción entre los elementos que conforman a cada uno de los bloques estará definida siempre por los procesos de Alta y/o modificación de la información y el proceso de Consulta. En ambos casos se mantiene una estrecha conexión con la base de datos.

En el caso de la consulta, el proceso implica recibir una petición por parte del usuario, validar la solicitud, ir por la información solicitada y realizar la presentación de la misma, una vez presentada, se brinda la opción de poder modificar dicha información<sup>3</sup>.

Para el caso de la alta, lo primero es validar la información proporcionada, la cual no deberá estar previamente registrada, en caso de estarlo se le presentará al usuario la información prerregistrada, dándose la opción de poder modificarla e ingresar los nuevos cambios, en el caso de que los datos proporcionados sean completamente nuevos y conformen la información suficiente para generar un nuevo registro, después de su respectiva validación serán insertados en la base de datos dando paso a la creación de un nuevo registro, el cual contará con las opciones de ser consultado o modificado, según las necesidades del Académico<sup>4</sup>.

<sup>3</sup> Las opciones de Modificación de datos solo se encuentra disponible para el módulo del Académico.

<sup>4</sup> La opción de Alta de nuevos registros en su totalidad solo se haya disponible para el módulo del académico, considerándose un caso especial para el módulo del administrador, en lo que respecta al alta de nuevos usuarios.



En el caso del módulo de la Dependencia, la interacción sólo sigue un flujo único, en el cual se realiza una petición por parte del usuario, se buscan una serie de patrones que coincidan con los datos solicitados y se da una respuesta o reporte en algún tipo de formato.

Por último, en el caso del Administrador, la interacción se establece en dos hilos de trabajo, el primero orientado en recibir a un nuevo usuario, validar su inexistencia en la base, así como la veracidad e integridad de la información proporcionada, ya sea el resultado de este proceso positivo o negativo, se procederá a agregar el nuevo usuario o a solicitar su revisión en caso de que la información presentada sea incorrecta.

Para el segundo hilo se tiene la baja de usuarios, proceso en el cual se realiza algo muy similar al alta pero esta vez se corrobora la existencia del usuario para poder realizar su eliminación de la base de datos.

Una vez establecido la interacción de los elementos, de cada módulo, procederemos a establecer cómo se realizará el manejo de los datos.

El Manejo de los datos se realizará a partir del *diagrama de clases* el cual implica a una serie de clases, para su realización se consideraron los casos de iteración establecidos previamente donde se tomaron los objetos definidos en los diagramas de la interfaz a nivel de clases.

Para poder definir las clases, UML sugiere seis características:

1. Información retenida. La clase será útil durante el análisis sólo si la información sobre el mismo ha de ser almacenada, transformada, analizada o manejada en algún otro modo. La información puede referirse a conceptos que deberán estar siempre registrados en el sistema, eventos o transacciones que ocurren en un momento específico.
2. Sistema externo. Si se tiene un sistema externo a este sistema, entonces es de interés en la etapa de modelado. Los sistemas externos deberán ser vistos como clases que el sistema contendrá o con los cuales interactuará.
3. Patrones, librerías de clases o componentes. Si se tienen patrones, librerías de clases o componentes, generalmente éstos son clases candidatos.

4. Dispositivos que el sistema maneja. Dispositivos técnicos que maneja el sistema se convertirán en clases que manejarán esos dispositivos.
5. Partes organizacionales. Especialmente en modelos de negocio, todas las partes que representan a la organización serán clases candidatas.
6. Roles de actores. Los roles de actores serán vistos como clases, por ejemplo, usuario, operador del sistema, administrador, cliente, etc.

Gráficamente, las clases se representarán en una caja rectangular dividida en 3 compartimentos: en la parte superior se encuentra el nombre de la clase, el cual está asociado con el objeto, en la parte media se encuentran los atributos y en la parte inferior se encuentran las operaciones o métodos de la clase. Las clases se relacionan entre sí a través de las relaciones descritas por líneas rectas entre dos clases y constan de roles y cardinalidad. Los roles son las frases que se encuentran en la relación y sirven para definir la cardinalidad de la relación, siempre considerando la unidad en la clase origen.

En nuestro caso y dada la magnitud de cada uno de los objetos empleados, su descripción más detallada tanto de sus variables así como de sus métodos para poder operar con los objetos, se ha decidido, presentar solo en el siguiente diagrama los paquetes que serán considerados para albergar las funciones de operabilidad de cada uno de los módulos que conforman al sistema, además del paquete destinado a realizar todas las transacciones entre los módulos definidos y la base de datos.

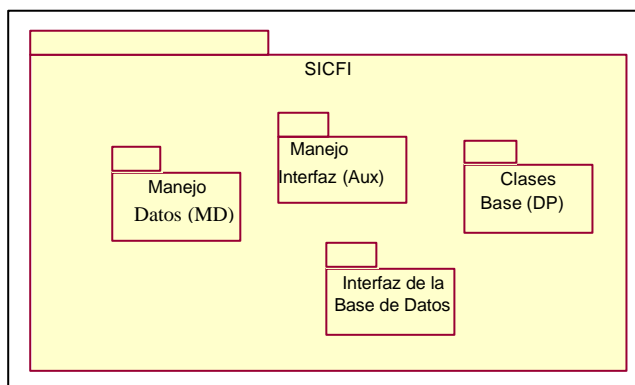


Figura 4.9. Clases

## 4.4 Diseño de la base de datos

Los datos son tan importantes como las instalaciones, el equipo o el personal de una institución. Se invierte tiempo y recursos (humanos y materiales) para adquirir los datos, la información derivada de los datos se puede utilizar para calcular la eficiencia del personal. Sin embargo a medida que los datos se guardan en bases de datos operacionales, su utilidad es limitada.

Es de gran importancia que los datos se utilicen eficazmente y no solamente de guardarlos y protegerlos.

Aunado a lo anterior se comprueba la importancia de hacer uso de los datos y no solamente tenerlos guardados, o no aprovechar esa base para generar productos que aumenten la eficiencia a la institución.

Los retos de la administración de datos se resumen en seis premisas:

- Muchos tipos de datos
- Categorías de datos que no son obvios
- Los mismos datos pueden tener muchos nombres
- Los mismos datos pueden tener muchas descripciones y formatos
- Los datos cambian con mucha frecuencia
- Los asuntos de política organizacional complican los asuntos operacionales

Se considerara al administrar los datos su previa organización, catalogados, protegidos y como deberá ser el acceso a los mismos; se debe establecer límites a los derechos y responsabilidades del procesamiento.

Por ende existen distintos objetivos que deben cumplir el diseño de nuestra base de datos:

- Abstracción de la información. Se debe de ahorrar a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de

archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

- Independencia. Donde se considera la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- Consistencia. En aquellos casos donde no se ha logrado esta redundancia nula, será necesario vigilar que la información repetida se actualice de forma coherente, es decir, todos los datos repetidos se actualicen de forma simultánea.
- Seguridad. La información almacenada en una base de datos puede llegar a tener un gran valor. El sistema debe garantizar la seguridad de la información frente a usuarios malintencionados, los cuales intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, para el manejo de la base de datos, se dispone de un complejo sistema de permisos a usuarios y grupos de usuarios, permitiendo otorgar diversas categorías de permisos.
- Integridad. Adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, proteger los datos ante fallos de hardware y software, datos introducidos por usuarios descuidados o cualquier otra circunstancia capaz de corromper la información almacenada.
- Respaldo y recuperación. El sistema debe proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos y de restaurar a partir de estas copias los datos que se hayan podido perder.
- Control de la concurrencia. En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es nos indica el ingreso de muchos usuarios accediendo a la base de datos, para recuperar información o almacenarla. También es frecuente que los accesos se realicen de forma simultánea. Así pues, se debe controlar este acceso concurrente a la información, el cual podría derivar en inconsistencias.
- Tiempo de respuesta. Lógicamente, es deseable minimizar el tiempo que el sistema tarda en darnos la información solicitada y en almacenar los cambios realizados.

Al cumplir con estos objetivos se pueden obtener ciertas ventajas y desventajas dentro del sistema de datos que a continuación se mencionan:

- Ventajas
  1. Facilidad de manejo de grandes volúmenes de información.
  2. Gran velocidad en muy poco tiempo.
  3. Independencia del tratamiento de información.
  4. Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
  5. No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
  6. Integridad referencial al terminar los registros.
  
- Desventajas:
  1. El costo de actualización del hardware<sup>5</sup>.
  2. Costo (salario) del administrador de la base de datos es costoso.
  3. El mal diseño de esta puede originar problemas a futuro.
  4. Un mal adiestramiento a los usuarios puede originar problemas a futuro.
  5. El mal diseño de seguridad genera problemas en esta.

Una vez definidos los objetivos a seguir para el desarrollo de nuestra base de datos y la importancia que éstos representan para nosotros a continuación procederemos a establecer las especificaciones de nuestro sistema con relación a la estructura que se deberá definir para la creación de las tablas.

#### **4.4.1 Especificaciones para la creación de las tablas**

Cada Académico tiene:

Un único número del registro nacional de población CURP

---

<sup>5</sup> En nuestro caso el software está bajo licencia GNU, lo cual nos permite deslindar costo alguno sobre el mismo.

- Un número de expediente asignado por la UNAM
- Un RFC con o sin homo clave
- Datos Generales: nombre (apellidos paterno, materno y nombre), fecha de nacimiento, lugar de nacimiento, sexo, estado civil (soltero, casado), nacionalidad, correo electrónico (e-mail) y página personal WEB.
- Puede tener muchas direcciones correspondientes al domicilio, trabajo en la UNAM o trabajo Externo a la UNAM. Cada dirección se compone de calle, número, colonia, entidad, municipio y CP. Una misma dirección puede corresponder a varios Académicos.
  - Puede tener diferentes números de teléfono (con extensión) correspondientes a su domicilio, trabajo(s) o fax. Un mismo número de teléfono puede corresponder a varios Académicos.
  - Puede tener diferentes correos electrónicos correspondientes a una dirección. Un mismo correo electrónico puede corresponder a varios Académicos.
- Diferentes grados o títulos obtenidos (licenciatura, especialización, maestría, doctorado, postdoctorado), que contiene un nombre, la institución, estado actual, el área, disciplina, subdisciplina correspondientes. Para todos los grados excepto postdoctorado se puede ingresar procedimiento de obtención de grado, título de la tesis y fecha de obtención del mismo. Para licenciatura y maestría además se permite ingresar promedio y si es que obtuvo mención honorífica.
  - Cada grado o título correspondiente al Académico puede contar con varias becas asignadas a el proyecto sobre el cual es desarrolla. La beca contiene un nombre descriptivo y la institución que la otorga. Una beca puede estar asignada a varios grados o títulos, lo cual es estipulado por los reglamentos de la institución que los otorga.
- Diferentes estancias, comisiones o sabáticos definidos por la institución donde se realiza o se realizó, estado actual, fecha de inicio y terminación, comentarios.
  - Cada estancia, comisión o sabático tiene una fecha de inicio y fin, comentarios y el estado en que se encuentra actualmente.
  - Cada estancia puede tener diversas instituciones que brindaron su financiamiento. Definido por nombre del financiamiento e institución financiadora. Un financiamiento puede asignarse a diferentes estancias, comisiones o sabáticos.
- Diversos cursos, seminarios o talleres definidos por su nombre y la institución donde se realizó. Un curso puede haber sido tomado por diversos Académicos.

- Cada curso tiene un estado de concluido o no, fecha de inicio y terminación, si se obtuvo el diploma o constancia.
- Cada curso puede tener una institución financiadora del mismo. Que se define con nombre de la beca e institución.
- Diversos conocimientos en cómputo relacionándolo con el software manejado y descripción del mismo (plataformas, lenguajes de programación, paquetería, otro). Además de ello, un mismo conocimiento de cómputo lo pueden tener diversos Académicos.
- Dominio de diferentes idiomas. Definidos por el nombre del idioma y un porcentaje de dominio en comprensión, expresión oral, lectura, redacción y traducción del mismo. Un mismo idioma lo pueden dominar varios Académicos
- Diferentes áreas de especialización que a su vez se dividen en alguna disciplina y subdisciplina específicas asociadas. Un área de especialización puede ser seguida por diferentes Académicos.
- Diferentes líneas de investigación, principales o no, definidas por un identificador y nombre, un tipo, una catalogación y comentarios. Una línea de investigación puede ser seguida por diferentes Académicos.
- Diversos proyectos de investigación definidos por un nombre, una clave, un Académico perteneciente al sistema que es el responsable del proyecto, si esta vigente o no, fecha de inicio y terminación, tipo, institución responsable y personas participantes divididas en Académicos, técnicos y estudiantes de los niveles de licenciatura, especialidad, maestría y doctorado.
  - Un mismo proyecto de investigación puede estar asignado a diferentes Académicos.
  - Cada proyecto de investigación puede tener diversas instituciones financiadoras. Se definen por su nombre de institución país y tipo.
    - Un financiamiento puede ser asignado a diferentes proyectos de investigación.
- Diversos Intercambios académicos definidos por el nombre del visitante, nacionalidad, institución de procedencia, fecha de inicio y terminación, estado actual comentarios,
  - Cada visitante puede tener diversas instituciones financiadoras. Se definen por su nombre de institución y país.
    - Un financiamiento puede ser asignado a diferentes proyectos de investigación.
  - Un intercambio académico puede ser realizado con diferentes Académicos.

- Diversos programas de estímulos definidos por un identificador (PEPRAC, Prima al desempeño, FOMDOC, otro), nombre, fecha de otorgamiento, nivel si es del SNI y el área a la que pertenece.
  - El SIN, contiene los niveles Candidato, 1, 2, 3 y emérito. Además de siete áreas específicas.
  - PAIPA y PRIDE tienen niveles A, B, C y D
  - Los demás programas de estímulos no contienen niveles ni áreas específicas.
- Diversos reconocimientos definidos por el tipo de premio, nombre del premio, fecha en que se otorgó, fecha de inicio y terminación, si es vitalicio o por periodo en cuyo caso se pide una fecha de inicio y terminación, nombre de la Institución otorgante.

## 4.5 Diseño del reporte de datos

De este punto se partió para generar la interfaz que interactuara con el usuario a sí como la forma para ser administrados los datos y la forma con la cual las dependencias obtendrán información del sistema, basta agregar que el reporte fue basado en los formatos entregados por los académicos para participar en el otorgamiento de estímulos, reconocimientos y premios otorgados por la Facultad de Ingeniería con el fin de motivar su desempeño académico a lo largo de toda su trayectoria docente dentro de la UNAM.

Para el módulo del académico, se cuenta con la opción de generar un reporte general o por secciones de la información registrada, dicho reporte será el equivalente a la presentación del currículum vitae. Considerando que el reporte sea factible de modificar consideramos presentar al mismo en formato “Word”, uno de los formatos más extendidos y de fácil edición mediante el cual el usuario podrá realizar la personalización del mismo.

Para el caso de la dependencia el reporte dependerá de la solicitud realizada ya sea a través de los índices y tabuladores registrados o por medio de una consulta personalizada. En el primero de los casos la respuesta siempre se dará en formato Excel, el cual es uno de los formatos más extendidos para la manipulación de datos en la dependencia, además de contar con la posibilidad de que a partir de este medio se puede conseguir la exportación de la información hacia otras fuentes de almacenamiento de una forma sencilla y económica. Para el segundo de los casos, se puede obtener o no una respuesta, ello dependerá de la coincidencia del patrón proporcionado con uno o más registros dentro de la base de datos, en el caso de que dicha coincidencia



exista se procederá a realizar la presentación de la información en primera instancia solo en pantalla con la opción de poder generar un reporte en formato Excel.

Una vez que se tiene la información detallada del diseño, se continúa con la fase del desarrollo del software. Considerando las especificaciones del software empleado para el desarrollo del sistema. Una vez comenzada la fase de programación, iniciará un proceso en donde se contestan preguntas, aclaran dudas y manejan los problemas que enfrentaremos, al momento de utilizar las especificaciones de diseño establecidas. Todo esto se describe en el siguiente capítulo.

## **5. Configuración y Desarrollo del Sistema de la FI.**

### **5.1 Configuración del servidor**

#### **5.1.1 Sistema Operativo**

##### **5.1.1.1 Instalación**

Para la puesta en Marcha del desarrollo del sistema, se proporcionó por parte de la Secretaría General un equipo de cómputo con las siguientes características:

- DELL Pentium 4
- 2.0 Ghz
- 256 MB Ram
- Sistema Operativo Windows XP

Por tal motivo se procedió a descargar las imágenes de los discos de instalación correspondientes a la distribución de Linux FedoraCore2, para realizar la instalación de dicho sistema en el equipo asignado como servidor, como elección de una configuración de servidor. Se habilitó la instalación del servidor apache y Postgres.

##### **5.1.1.2 Configuración**

Al finalizar la instalación del sistema se configuró el acceso a la red mediante el comando netconfig, que permite la entrada de los parámetros necesarios para su entrada a la red.

##### **5.1.1.3 Actualización**

Mediante la suscripción al servicio Fedora Core Updates se realizaron las actualizaciones disponibles para los paquetes instalados en el sistema, así como la descarga e instalación de los parches para el kernel del sistema, al igual que otras aplicaciones.

#### 5.1.1.4 Seguridad

Además de las actualizaciones realizadas, que ayudan en el incremento de la seguridad en vulnerabilidades descubiertas dentro del sistema, otro aspecto que fue tomado dentro de la seguridad fue el habilitar únicamente los puertos para el acceso vía http (servidor Web) y para el manejo de archivos y de la consola del sistema de manera remota a través del secure shell así como el puerto para el acceso remoto a la base de datos durante la etapa de desarrollo.

#### 5.1.2 Software

Se realizó la descarga de las siguientes aplicaciones hacia la ruta [/usr/local/]:

Apache (1.3.29)	<a href="http://www.apache.org/">http://www.apache.org/</a>
Java (1.4.2.07)	<a href="http://java.sun.com/">http://java.sun.com/</a>
Postgres (7.4.7)	<a href="http://www.postgresql.org/">http://www.postgresql.org/</a>
Apache Ant (1.6.2)	<a href="http://ant.apache.org/">http://ant.apache.org/</a>
JDBC (pg73jdbc3)	<a href="http://jdbc.postgresql.org/">http://jdbc.postgresql.org/</a>
Tomcat (5.0.30)	<a href="http://tomcat.apache.org/">http://tomcat.apache.org/</a>
iReport (0.4.1)	<a href="http://jasperforge.org/sf/projects/ireport">http://jasperforge.org/sf/projects/ireport</a>

La instalación de éstas, se realizó mediante la ejecución de los siguientes comandos, que realizaron la compilación e instalación de las aplicaciones.

##### 5.1.2.1 Instalación de JAVA

- Entrar como super usuario (root)
- Crear la carpeta 'java' en el directorio
  - `[mkdir /usr/local/java]`
- Darle permisos al archivo de ejecución
  - `[chmod +x j2sdk-1_4_2_07-linux-i586.bin]`
- Iniciar la instalación desde la carpeta /usr/local/java.

- `./j2sdk-1_4_2_07-linux-i586.bin`
- Al iniciar la instalación aparece la licencia del programa, para no leerla presionar `[CTRL + C]`. Posteriormente nos pregunta si estamos de acuerdo con la licencia, introducir `[Y]` y `[ENTER]` para continuar con la instalación.
- Una vez terminada la instalación, se debe modificar el archivo `'/etc/profile'` para que todos los usuarios puedan usar java desde su carpeta local. Abrir el archivo y agregar las siguientes líneas:

```
PATH="$PATH:./usr/local/java/bin:./usr/local/java/jre/bin"
JAVA_HOME="/usr/local/java/"
CLASSPATH="./usr/local/java/"
export JAVA_HOME CLASSPATH
```

### 5.1.2.2 Instalación de apache

- Pasar el archivo descargado a la carpeta src `[cp -p httpd-2.2.0.tar.gz /usr/local/src/]`
- Cambiarse al directorio antes definido `[cd /usr/local/src]`
- Descomprimir el archivo `[tar xvzf httpd-2.2.0.tar.gz]`
- Cambiarse a la carpeta creada `[cd /usr/local/src/httpd-2.2.0]`
- Configurar apache `[./configure --prefix=/usr/local/apache --enable-module=most --enable-mods-shared=most]`
- Compilación de apache `[make]`
- Instalación de apache `[make install]`
- Iniciar el servidor `[/usr/local/apache/bin/apachectl start]`
- Detener el servidor `[/usr/local/apache/bin/apachectl stop]`

### 5.1.2.3 Instalación de Tomcat

- Descomprimir el archivo `[tar xvzf apache-tomcat-5.0.30.tar.gz]`
- Cambiarse a la carpeta creada `[cd /usr/local/src/httpd-2.2.0]`
- editar `/etc/profile` para añadir la variable de entorno `CATALINA_HOME`:  
`CATALINA_HOME=/usr/local/tomcat`
- `export CATALINA_HOME`

- Iniciar el servicio de tomcat [*/usr/local/tomcat/bin/startup.sh*]
- Iniciar el servicio de tomcat [*/usr/local/tomcat/bin/shutdown.sh*]

#### 5.1.2.4 Instalación del conector JK

- Pasar el archivo descargado a la carpeta src [*cp -p jakarta-tomcat-connectors-1.2.15-src.tar.gz /usr/local/src/*]
- Cambiarse al directorio antes definido [*cd /usr/local/src*]
- Descomprimir el archivo [*tar xvzf jakarta-tomcat-connectors-1.2.15-src.tar.gz*]
- Cambiarse a la carpeta creada [*cd jakarta-tomcat-connectors-1.2.15-src/jk/native/*]
- Ejecutar [*./buildconf.sh*]
- Configurar conector [*./configure --with-apxs=/usr/local/apache/bin/apxs*]
- Compilación de conector [*make*]
- Instalación de conector [*make install*]
- Crear un nuevo archivo *workers.properties* en */usr/local/apache/conf* con el siguiente contenido:

```
workers.tomcat_home=/usr/local/tomcat/
workers.java_home=$JAVA_HOME
ps=/
worker.list=default
worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
worker.default.lbfactor=1
```

- Editamos el archivo de configuración de Apache [*/usr/local/apache/conf/httpd.conf*] para cargar el módulo. En la sección *LoadModules* añadimos:

```
LoadModule jk_module modules/mod_jk.so
```

Y al final del archivo ponemos las siguientes líneas:

```
JkWorkersFile "conf/workers.properties"
JkLogFile "logs/mod_jk.log"
JkLogLevel warn
```

```
JkMount /jsp-examples default
```

```
JkMount /jsp-examples/* default
```

- Por último iniciamos Apache y Tomcat:

```
/usr/local/apache-tomcat-5.5.16/bin/startup.sh
```

```
/usr/local/apache/bin/apachectl start
```

### 5.1.2.5 Instalación de ANT

- Descomprimir el archivo [`tar xvzf apache-ant-1.6.2-bin.tar.gz`]
- Editar `/etc/profile` para añadir la variable de entorno `ANT_HOME`:

```
ANT_HOME=/usr/local/ant
```

```
export ANT_HOME
```

### 5.1.2.6 Instalación de JDBC

- Colocar el archivo descargado `pg73jdbc3.jar` en la siguiente dirección: `/usr/local/java/lib`

### 5.1.2.7 Instalación de Postgresql

- Entrar como súper usuario(root)
- Crear grupo y usuario postgres: [`groupadd postgres`]; [`useradd -g postgres postgres`]
- Descomprimir [`tar -xvzf postgresql-7.4.7.tar.gz`]
- Cambiarse a la carpeta creada [`cd postgresql-7.4.7`]
- Ejecutar [`./configure -with-java -with-tcl -without-tk -with-openssl=/usr/local/ssl`]
- Compilación de postgres [`make`]
- Instalación de postgres [`make install`]
- Crear un directorio data [`mkdir /usr/local/pgsql/data`]
- Asignar los directorios a sus propietarios: [`chown postgres /usr/local/pgsql/data`]
- Entrar como usuario postgres [`su - postgres`]
- Ejecutar para instalar la BD: [`/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data`]
- Iniciar la BD: [`/usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data`]
- Arrancamos el cliente: [`psql`] y se ejecuta:

- Se crea un usuario: [*Createuser proyectoSIC*]
- Se crea una base: [*Createdb baseSIC*]

### 5.1.2.8 Instalación de iReport

- Descomprimir [*tar xvzf iReport-0.4.1.tar.gz*]
- Copiar el archivo tools.jar de java[*/usr/local/java/lib/tools.jar*] a:[*/usr/local/iReport/lib*]
- Ejecutar [*./configure -with-java -with-tcl -without-tk -with-openssl=/usr/local/ssl*]
- Colocar el jdbc[*pg73jdbc3.jar*] en:[*/usr/local/iReport/lib*]
- Editar */etc/profile* para añadir la variable de entorno *IREPORT\_HOME*:

```
IREPORT_HOME=/usr/local/iReport
export IREPORT_HOME
```

## 5.2 Elaboración de la base de datos

Mediante el lenguaje de definición de datos del postgresql, se generan las sentencias requeridas para la creación de las tablas definidas en la etapa de diseño.

Una vez definidas las entidades y sus relaciones en el diseño, se procedió a dividir las en dos rubros: el primero son los catálogos cuyos nombres están anteceditos por el prefijo “cat\_” y después por el nombre, cabe señalar que en ellos ya se cuenta con información predefinida y a su vez pueden ir almacenando información para su utilización. El segundo son las tablas principales con el formato *inv\_nombre*, en ellas que irán almacenando toda la información proporcionada por los académicos y los catálogos.

Tanto catálogos como tablas primarias tendrán como llave la CURP, la cual es empleada para manejar las relaciones entre catálogos y tablas.

### 5.3 Programación de Clases

Para que se pudiera comenzar con la programación de las páginas del SICFI fue necesario definir la estructura de directorios, las clases y archivos que serían utilizados a partir de la etapa de diseño.

También una parte fundamental fue la creación de archivos y clases iniciales que deben ser cargados por cada página. Las ventajas que se tienen con estos archivos es reducir código y permitir fáciles actualizaciones tanto de configuración como de diseño.

Algunas de las acciones que realizan estos archivos al ser cargados se enlistan a continuación:

- Control de acceso de usuarios a páginas del sistema.
- Definición de variables y constantes, como el url base del sitio, directorio de imágenes, control de errores, etc.
- Conexión a base de datos.
- Definición de funciones empleadas.
- Carga de librerías usadas para la interacción con la base de datos.
- Manejo de la Información contenida en la base de datos

Para llevar un mejor manejo de los datos y se cumplan las acciones antes mencionadas, fue necesario dividir las clase en tres grupos como se estipuló.

En primera instancia se definieron las clases del dominio del problema, considerando los elementos necesarios para su desarrollo, luego, las clases de manejo de datos, considerando los elementos involucrados, se declaran las funciones de manejo de datos que es la manera en la que se va a manipular la información. Por último las de Interfaz, divididas en dos grupos, alta y consulta, junto con ellas las funciones y variables con intervención en los JSP's.



## 5.4 Programación de JSP's

Una vez definidas las clases, sus propiedades y sus métodos es posible proseguir con la programación de las páginas involucradas en el sistema. De esta forma, las páginas, se clasificarán de acuerdo a su funcionalidad en:

- **Control:** Usadas para el alta y edición de los registros. Aquí es donde se recibe información por parte del usuario para que después sea almacenada mediante el empleo de etiquetas HTML y JavaScrip permitiendo la creación de formularios con distintos tipos de entradas de datos.
- **Presentación:** Despliegan la información registrada por los académicos. Dicha información podrá ser presentada en dos formatos en conjunto, uno de ellos es el del HTML mientras que el segundo el formato Word o Excel ya antes mencionado en la etapa de diseño.
- **Reportes:** Estas páginas son empleadas para realizar reportes de los datos contenidos en la base a los usuarios.

### 5.4.1 Formularios

Para el almacenamiento de información proporcionada por los usuarios dentro de la base de datos, se requieren interfaces gráficas de usuario (GUI's) que permitan a éste la introducción y envío de los datos a la base. Para el caso de las aplicaciones Web, se hace uso de formularios en HTML, lo cuales permiten el uso de distintos tipos de entradas de datos. Ejemplo de ello se muestra en la siguiente figura.

Una parte importante en la recepción de datos enviados por los usuarios, es verificar que los datos cumplan con las características establecidas para éstos. Siguiendo un proceso de validación realizado del lado del cliente mediante el empleo de JavaScript y del lado del servidor con JSP y java mediante las clases (Dominio del Problema, Manejo de Datos e InterfazAlta). La validación del lado del cliente brinda la ventaja de no requerir la validación de los datos por parte del servidor, aunque la validación del lado del servidor no debe dejar de realizarse ya que existe la posibilidad de la manipulación de los datos a enviar por parte del usuario y esto provocaría errores al momento de realizar el almacenamiento.

### Datos Personales

#### Consulta

**NOTA:** Los campos con \* son indispensables de llenar


No. Trabajador UNAM*:	049563		
Nombre(s)*:	Gonzalo		
Primer Apellido*:	López	RFC*:	LOHG490425-S60
Segundo Apellido:	de Haro	CURP*:	LOHG490425H0FFRRND8
Fecha de Nacimiento*:	25 - 04 - 1948 (dd - mm - aaaa)		
Sexo*:	Masculino		
Estado Civil*:	Casado		
Nacionalidad*:	Mexicana		
Lugar de Nacimiento*:	México		
Página WEB:			
Celular:	044552481435		

Figura 5.1. Datos personales

Algunas de las validaciones que son realizadas son:

- Los datos obligatorios deben ser incluidos en el envío.
- Un dato deberá contener solamente caracteres numéricos.
- Un dato deberá contener un determinado número de caracteres.
- Una fecha inicial tiene que ser menor o igual que una final.
- Un dato deberá ser único dentro de una tabla en la base de datos.

Las páginas que fueron diseñadas para el alta y actualización de registros

#### 5.4.2 Despliegue de Información

Las páginas encargadas del despliegue de información, hacen consultas a las tablas que forman parte de la base de datos mediante las clases (Manejo de Datos e Interfaz Consulta). Las consultas permiten la extracción de datos requeridos de acuerdo al objetivo de la página.

Una vez realizada la extracción de datos, éstos son colocados en plantillas para ser enviados al usuario y presentarse de una manera entendible y agradable en el navegador. Es importante tomar en cuenta la manera en la cual debe ser presentada la información para ser atractiva a los usuarios.

### 5.4.3 Diseño

Mediante el empleo de editores de HTML y JavaScript fueron creadas las plantillas que son utilizadas para todas las páginas, gracias a esto es posible cambiar o actualizar fácilmente el diseño, para el cambio de colores, tipos y tamaños de letra, encabezados y pies de página.

El uso de plantillas permite separar los elementos de presentación (HTML) del código de la aplicación (JSP). El código de la aplicación se encarga de extraer los datos dinámicos e insertarlos en las plantillas correspondientes que pueden estar siendo utilizadas por varias páginas que son accedidas.

La estructura final de las páginas se muestra en la siguiente ilustración:

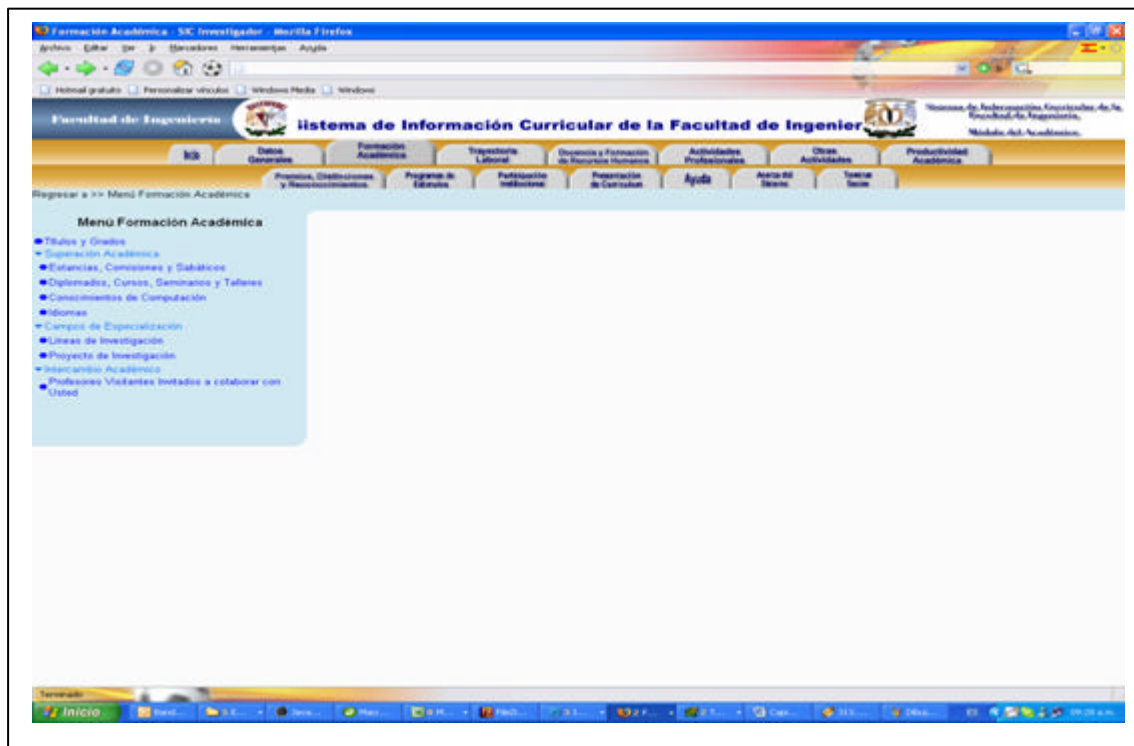


Figura 5.2 Menú

## **6. Implementación del sistema.**

Después de haber diseñado y desarrollado el sistema el siguiente paso es la implementación del mismo en un servidor; las razones fundamentales por la que se requiere un servidor son por el número de usuarios que utilizaran el sistema, la rapidez del mismo, seguridad, capacidad, confiabilidad y recursos. Otros aspectos importantes a considerar en la implementación son la plataforma en la que será implementado el sistema la cantidad de memoria, procesador y disco dura que se requieran para ejecutar los procesos.

### **6.1 Pruebas.**

Durante el desarrollo de un sistema se debe tener contemplado una fase de pruebas para evaluar el desempeño del sistema y poder encontrar fallas o anomalías en el funcionamiento que pudieran repercutir de manera drástica, en el funcionamiento del sistema. Cabe señalar que estas pruebas se basan teniendo en cuenta las especificaciones planteadas desde el inicio, en esta etapa el sistema se emplea de una forma experimental. En esta etapa de pruebas se toman como entradas, conjuntos de datos de prueba para su procesamiento y después se examinan y analizan los resultados.

Dependiendo del porcentaje de información curricular capturada por los académicos y de los comentarios que estos dan del SICFI, es como evaluamos la utilización y el funcionamiento del sistema.

Una vez implementado el sistema es necesario tomar en cuenta los procedimientos:

- Depuración de información
- Actualización de vínculos
- Corrección de errores
- Optimización de las funciones
- Se establecen nuevas herramientas
- Asesoría técnica si es requerido.

El ciclo de vida del desarrollo requiere varios niveles de prueba, iniciando con la prueba unitaria y posteriormente la prueba de integración que tiene el objetivo de demostrar la funcionalidad del sistema en su conjunto.

La prueba unitaria, verifica que un componente funcione adecuadamente dependiendo de los tipos de entrada acordes al diseño del componente. Esta prueba se hace, siempre que sea posible en un ambiente controlado de modo que el responsable de realizar la prueba, pueda ingresarle al componente que se esta probando un conjunto predeterminado de datos y observar que acciones y datos de salida se producen. El siguiente paso es asegurar que las interfaces entre los componentes están definidas y se manejan correctamente.

Además de efectuarse la prueba de integración, que tiene la función de verificar que los componentes del sistema trabajen de manera conjunta conforme a lo establecido previamente en las especificaciones. Cabe señalar que a los módulos ya integrados se les realizan otras pruebas para revisar el funcionamiento del sistema en conjunto, entre las que se encuentran:

#### **6.1.1 Prueba de función.**

Verifica, que el sistema integrado realice todas y cada una de las funciones establecidas en los requerimientos.

#### **6.1.2 Prueba de desempeño (rendimiento).**

Compara el rendimiento de los componentes ya integrados con requerimientos que incluyan la seguridad, exactitud, velocidad y confiabilidad, los cuales restringen de manera significativa las funciones del sistema. Para el SICFI se realizaron las pruebas de seguridad y de restricción de funciones.

La prueba de aceptación consiste en Simular una carga de n-usuarios accediendo y realizando operaciones en el sistema SICFI con la finalidad de evaluar el rendimiento y consumo de recursos del sistema, para el cual empleamos Pruebas de Stress de Microsoft® Web Application Stress Tool (WAS) y Pruebas de Load Balancing los cuales describiremos a continuación.

Microsoft Web Application Stress Tool es una herramienta útil para probar la carga de usuarios. Fue diseñada para simular de manera real varios navegadores que solicitan páginas a una aplicación Web.

Se puede utilizar para generar cargas personalizables en varios servicios de Internet, ofrece un amplio conjunto de características y permite obtener información acerca del rendimiento y estabilidad de un sitio Web determinado.

Su objetivo es crear un ambiente de producción que sea lo más parecido al que se va a utilizar, de esta forma se pueden localizar y eliminar problemas en la aplicación WEB antes de liberar la versión instalable.

WAS de manera sencilla permite grabar una secuencia de pasos de navegación y luego los repite n veces durante un tiempo determinado, se puede indicar el tiempo de retardo entre reintentos y la cantidad de accesos simultáneos.

El Balanceo de carga o Load-Balancing tiene la función de distribuir una tarea a un conjunto de procesadores, en donde cada procesador tiene aproximadamente la misma cantidad de trabajo para ejecutar que los demás.

A medida que crecen las necesidades de la red, el servicio "Load Balancing" permite escalar fácilmente todo un sistema de Internet, tanto para servidores como aplicaciones. De este modo se aprovecha al máximo el uso de cada uno de los servidores de la red llevando al mínimo las posibilidades de caída del sistema.

#### **6.1.2.1 Desarrollo de la prueba.**

Para realizar esta prueba se obtuvo la herramienta WEB application Stress Tool de Microsoft, la versión empleada es gratuita y se obtiene del sitio:

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/intranet/downloads/webstres.asp>.

Para incrementar la confiabilidad, las peticiones deben realizarse desde una maquina que no sea el mismo servidor Web, cabe señalar que uno de los objetivos mas importantes es aprovechar las pruebas para monitorear el uso del procesador, de la memoria, etc.

Para la realización de esta prueba se emplearon las tres maquinas en las que se desarrollo el SICFI, una de ellas tenía la función de Servidor (Server) y en las otras dos se ejecutó el WAS, simulando diversos clientes.

Las características de las PC son las siguientes:

- Pentium 4, 1.2 GHz
- 128 MB RAM
- HD 20GB
- Windows 2000 (Clientes), Linux 7.2 (Servidor)

Para la realización de esta prueba tuvimos la necesidad de crear 1500 usuarios.

Como primera prueba, se ejecuto WAS en una máquina-cliente y el servidor ejecutando TOMCAT 4.1 y Postgresql. En esta prueba se observó que todo funcionaba correctamente, sin embargo el servidor no respondía tan rápidamente a otras peticiones, por lo que se dijo que el sistema sufría una sobre carga. La solución a este problema fue configurar a Postgresql para que aceptara mas conexiones de las que por default se le asignan (32), incrementando hasta 1024 conexiones.

La segunda prueba consistió en separar la base de datos del servidor, para ello, una maquina-cliente funciono como servidor de base de datos mientras que la otra maquina-cliente ejecutaba WAS y hacían peticiones al servidor el cual únicamente estaba ejecutando TOMCAT. En este caso, la respuesta fue muy favorable, pues la carga sobre el servidor no fue tan pesada al separar las operaciones de base de datos del servidor.

La tercera tuvo la finalidad de probar el balanceo de carga sobre un servidor configurado para ejecutar dos instancias de TOMCAT. Para esta prueba se logro que uno los dos clientes utilizaran las dos instancias de TOMCAT. Con esta prueba se confirmo que la aplicación puede ser balanceada en carga, los resultados observados en la base de datos confirmaron que no había tenido ningún tipo de problemas al realizar las operaciones usando load-balancing. Sin embargo, al ejecutarse las dos instancias de TOMCAT sobre un mismo servidor implico que la carga sobre el mismo bajara su tiempo de respuesta.

Por ultimo se realizaron e las cuales se fueron variando diferentes parámetros, por ejemplo se fue variando la memoria RAM, la cual se probó para una capacidad de tanto con 128MB y otra como con 384 MB, otro parámetro variado fue la cantidad de procesos concurrentes (threads) incrementándolos en la siguiente proporción 1,10 y 100, por ultimo se cambiaron los retardos (delay) tanto en hilos como en peticiones. Cabe señalar que estos cambios tiene como finalidad observar el comportamiento del sistema, ante diferentes cambios que se pueden ocurrir.

### **6.1.3 Prueba de Aceptación.**

La prueba de aceptación en ocasiones se realiza en su ambiente real pero a menudo se ejecuta en una ubicación diferente a la definitiva. Es por esta razón que al completar la prueba de aceptación el sistema aceptado se instala en el ambiente en el que será utilizado y se ejecuta una última prueba de instalación para permitir que los usuarios trabajen con él ya puesto el sistema en el sitio real.

Sin importar el tamaño del sistema, el tipo de prueba descrito en cada paso, es necesario para asegurar su correcto funcionamiento

Dependiendo del tamaño de la empresa que usará el sistema y el riesgo asociado a su uso, puede hacerse la elección de comenzar la operación del sistema sólo en un área de la empresa (como una prueba piloto), que puede llevarse a acabo con un departamento ó con una ó dos personas.

Cuando se implanta un nuevo sistema lo aconsejable es que el viejo y el nuevo funcionen de manera simultánea ó paralela con la finalidad de comparar los resultados que ambos ofrecen en su operación, además dar tiempo al personal para su capacitación y adaptación al nuevo sistema.



#### **6.1.4 Prueba de Seguridad.**

Antes de la implementación de un sistema es conveniente considerar un punto de mucha importancia, como lo es la seguridad. Hoy en día existen múltiples amenazas de tipo externo que pueden hacer que un sistema funcione de manera errónea e incluso que deje funcionar.

Por ello, las pruebas de seguridad en este sentido deben ser aplicadas y estudiadas para obtener la mayor cantidad de información relativa a posibles huecos o fallas de seguridad que pudieran presentarse en el sistema. La principal finalidad de estas pruebas es el atacar este tipo de fallas para así obtener un sitio seguro y estable.

En el caso de SICFI existen cuentas de usuario que son restringidas a cierta información y aunque se podría pensar que los académicos no intentarían algún ataque, siempre hay gente externa mal intencionada, así que las pruebas de seguridad dentro del sistema consistieron en identificar la existencia y el funcionamiento de las restricciones necesarias para los diferentes tipos de usuarios, por ejemplo permisos para ver información que sólo le corresponde a cierta área ó usuario, la edición de la misma, el borrado de registros, etc.

Asesoría tecnológica al usuario y dependencias que lo requieran. A través de las estadísticas del sitio se mide su utilización así como el funcionamiento del mismo a través de sugerencias y comentarios.

Otro aspecto importante a tomar en consideración es que el sistema no provoque un desbordamiento de pila; es decir cuando en un sistema se ejecutan más procesos de lo que puede soportar el mismo, esto puede provocar la caída del servidor, y por consecuencia del sistema mismo, por lo cual estaríamos hablando de un sistema poco confiable e inseguro.

Cuando las pruebas se llevan a cabo exitosamente y el sistema trabaja como se estableció en el diseño, se le denomina sistema verificado, y luego se compara con las expectativas del cliente. Si se comprueba satisfactoriamente que el sistema que se ha construido cumple los requerimientos, se tiene un sistema validado.

### **6.1.5 Pruebas de Portabilidad y Usabilidad.**

Debido a que el sistema SICFI se desarrolló en Java, esto nos garantizará un alto porcentaje de compatibilidad en diferentes plataformas (las cuales pueden ser Linux, Windows, etc.) en las cuales se puede ejecutar el sistema.

Es decir un académico que cuente con una terminal con acceso a internet (computadora personal, estación de trabajo, laptop) podrá acceder al sistema SICFI sin ninguna dificultad; todo esto es posible gracias a que este sistema tiene como característica un alto grado de portabilidad, en cuanto a la usabilidad se refiere a que tan fácil es usar el sistema, por cuanto se usará en su construcción elementos estándares de las aplicaciones, bajo interfaz gráfica

## **6.2 Mantenimiento**

El sistema SICFI maneja una gran cantidad de información y dado que la facultad de Ingeniería puede sufrir cambios significativos tanto en su estructura como en sus procesos; es necesario que este sistema se adapte a tales circunstancias las cuales fueron tomadas en cuenta durante la etapa del diseño del sistema. La finalidad de un sistema en cuanto a mantenimiento se refiere; es llevarlo a cabo de una manera fácil y rápida para que en el caso de que se presente una situación que implique nuevos requerimientos o ajustes en la información se lleve a cabo sin mayor contratiempo.

La documentación del sistema SICFI esta conformada por:

- Manual de usuario.
- Manual técnico.
- Manual del administrador.

Estos manuales son necesarios para poder llevar a cabo las tareas de mantenimiento tales como el respaldo de la base de datos, la configuración del sistema operativo y del servidor, restauración de las copias de respaldo en el caso de una contingencia, etc.

Considerando que la construcción del sistema estará basada completamente en la documentación que se generara, se estima que cualquier modificación se podrá ejecutar sin complicaciones.

## **7. Manuales**

### **7.1 Manual Técnico**

#### **7.1.1 Introducción**

El presente manual pretende ayudar al encargado del sistema SICFI, a darle un mantenimiento, y una expansión de una manera metódica la cual le permitirá controlar los elementos involucrados de una forma muy sencilla.

Por esa razón, se dan lineamientos generales, es decir, explicamos el propósito de las actualizaciones, respaldos y mantenimiento adecuado de los elementos que interviene en el sistema.

El encargado del SICFI deberá contar con los siguientes conocimientos como mínimo:

- Ingeniería del software
- Conocimiento de sistemas Cliente/Servidor
- Programación en java
- Conocimiento de base de datos
- Manejo de herramientas para generar reportes en Linux
- Instalación, configuración, mantenimiento de servidor Apache, Tomcat bajo Linux
- Conocimientos de diseño de imágenes
- Conocimiento en el manejo de HTML, JavaScript y XML

#### **7.1.2 Requerimientos del sistema**

Se usará para la elaboración del SICFI las siguientes herramientas tanto primarias como secundarias, ya que se definió que el desarrollo se llevará a cabo mediante software libre, por lo tanto, se buscaron herramientas adicionales que permitieran llevar a cabo el desarrollo del sistema:

### **7.1.2.1 Hardware.**

- Procesador con velocidad mayor a 300 MHz.
- Conexión al Internet
- Memoria al menos de 512 MB
- Espacio libre en disco al menos de 40 GB

### **7.1.2.2 Software Primario**

- Servidor de la base de datos PostgreSQL 7.4.1 ó superior
- Lenguaje de programación JDK 1.4.2 ó superior.
- Servidor Web Apache 1.3.29 ó superior
- Contenedor de JSP's Apache TomCat Versión 5.30 ó superior
- Sistema Operativo Linux ó Windows
- Interfaz entre Java y postgres JDBC para PostgreSQL 7.4.1 ó superior
- Reporteador iReport 0.5.0 o superior

### **7.1.2.3 Comunicaciones**

- Soporte para redes basadas en protocolo TCP/IP y SSL.

### **7.1.2.4 Software secundario**

- Rational Rose versión 98. Se utilizará para el análisis y diseño de los datos.
- DÍA versión 0.9 Se utilizará para realizar la representación grafica del modelado de los datos Entidad – Relación.
- Eclipse. Se utilizará para el diseño y construcción de la Interfaz.
- GIMP. Como Herramientas adicionales

### 7.1.3 Respaldo

El respaldo del sistema involucra respaldar todos los archivos, así como la base de datos.

Para facilitar la distribución e interacción entre JSP's / Servlets estos son agrupados y respaldados en una estructura denominada WAR que se describe así:

- / \*.html \*.jsp \*.css: Este directorio base contiene los elementos típicamente empleados para un sitio web: Documentos HTML, CSS, JavaScript y Graficas; además en este directorio residen los JSP's a utilizarse en el WAR.
- /WEB-INF/web.xml: Este archivo contiene elementos de configuración del WAR como: Página de Inicio, Ubicación ("Mapeo") de Servlets, parámetros para componentes adicionales tales como "Struts" y otros elementos como manejo de errores.
- /WEB-INF/classes/: Este directorio contiene las clases Java utilizadas dentro del WAR, es en este directorio donde *generalmente* residen los Servlets diseñados para el WAR.
- /WEB-INF/lib/: Este directorio contiene los archivos JAR que serán utilizados por la aplicación, estos generalmente corresponden a las clases (JAR's) utilizadas para conectarse a Bases de Datos o aquellas utilizadas por librerías de JSP's.

Para crear un war es necesario ejecutar el siguiente comando:

- `jar cvf SICFI.war *`

Para el respaldo de la base de datos se emplea el siguiente comando:

- `/usr/bin/pg_dump -i -h localhost -p 5432 -U "proyectoSIC" -F c -b -v -f "baseSIC.backup" "baseSIC", asi como2`

Para restaurar una versión de la base de datos respaldada, sólo se ejecuta el siguiente script:

- `/usr/bin/pg_restore -i -h localhost -p 5432 -U "proyectoSIC" -d "baseSIC" -v "/root/baseSIC.backup"`

## 7.1.4 Documentos de referencia

### 7.1.4.1 Documentación de software existente

- <http://www.apache.org/foundation/how-it-works.html>
- <http://java.sun.com/reference/docs/>
- <http://www.postgresql.org/docs/>
- <http://ant.apache.org/manual/index.html>
- <http://tomcat.apache.org/tomcat-5.0-doc/index.html>
- <http://jasperforge.org/sf/wiki/do/viewPage/projects.ireport/wiki/HomePage>
- <http://www-306.ibm.com/software/sw-library/#Rational>
- <http://jdbc.postgresql.org/documentation/docs.html>
- [http://www.fceia.unr.edu.ar/lcc/cdrom/Instalaciones/Oficina/Dia/Documentacion/Dia\\_a\\_dra\\_wing\\_program.html](http://www.fceia.unr.edu.ar/lcc/cdrom/Instalaciones/Oficina/Dia/Documentacion/Dia_a_dra_wing_program.html)
- <http://www.eclipse.org/documentation/>
- <http://www.gimp.org.es/modules/downloadse/singlefile.php?cid=1&lid=81>

### 7.1.4.2 Documentación del sistema

- Introducción
- Estrategia
- Requerimientos
- Diseño
- Implementación
- Pruebas
- Comentarios Finales
- Manuales

## 7.2 Manual del Académico

Con este manual del académico pretendemos dar los lineamientos más generales referentes al uso del sistema SICFI.

Uno de los objetivos principales de este manual es mostrar las funciones de los elementos que en pantalla se mostraran como: los botones, menús y submenús. También se indicará como se puede accederse a las ventanas que componen el sistema así como los elementos que las componen de manera general; no es ni el propósito ni el alcance del presente manual el explicar que información debe ser capturada.

Este manual explica sólo el funcionamiento del programa, el buen o mal uso de la información que este proporcione o la integridad y veracidad de la información que en este se capture dependen completamente del administrador y de las normatividades correspondientes.

A continuación daremos una explicación de manera general del manual del académico, la cual será una buena referencia de uso del funcionamiento del sistema SICFI. Para poder usar el sistema SICFI es necesario tener una computadora con acceso a internet. Una vez que un académico entra a internet, debe teclear esta dirección electrónica en la barra del explorador:

<http://132.248.54.207:8080/SICFI/inicial.jsp>

Esta liga llevará al académico a iniciar su sesión en el sistema SICFI cuya pantalla de inicio se muestra en la figura 7.1.

La primera vez que un académico entre al sistema SICFI con la finalidad de capturar su información personal, deberá seleccionar la opción “Inicio” o el escudo de la Facultad de Ingeniería” que se muestra señalada en la figura 7.1. Una vez seleccionada la opción “Inicio” ó el escudo de la Facultad de Ingeniería”, aparecerá la pantalla de la figura 7.2, en donde proporcionará su CURP y Numero de trabajador de la UNAM para poder darse de alta en el sistema.

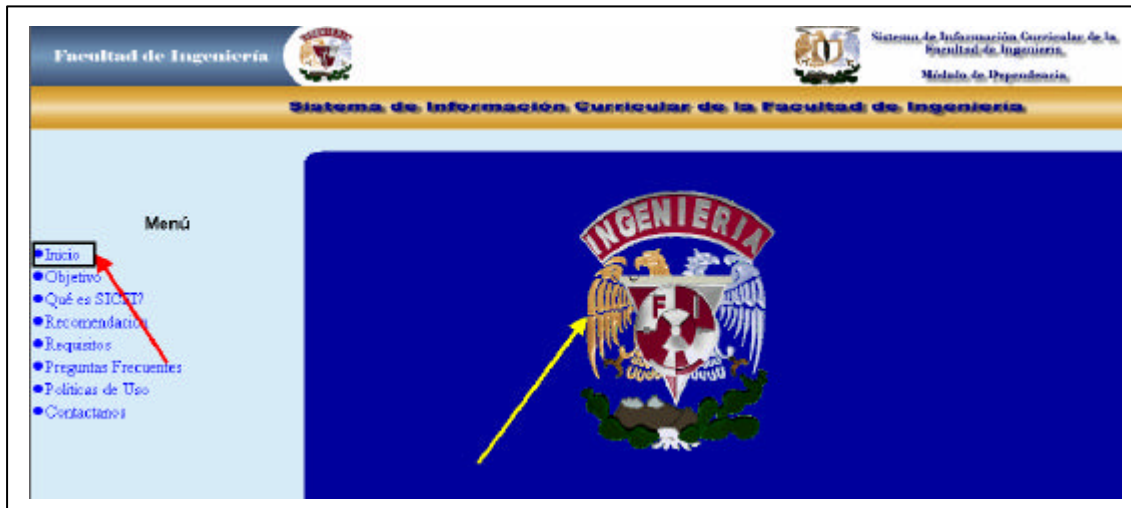


Figura 7.1 Principal

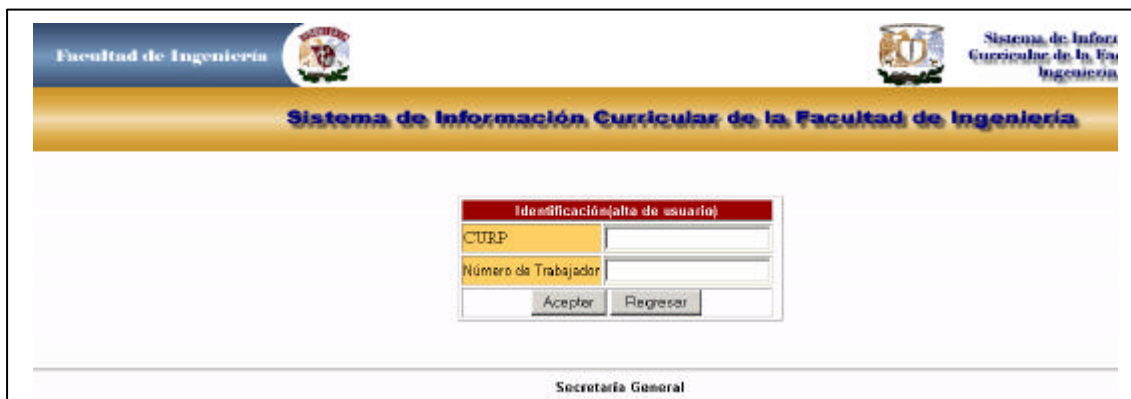



Figura 7.2 Logout

Al ingresar los datos anteriormente mencionados, el sistema SICFI desplegará una pantalla en la cual el académico deberá crear un login y su correspondiente password; los cuales serán validados por el mismo sistema. En esta misma pantalla también deben ingresarse los datos personales, figura 7.3.

Para posteriores accesos; el sistema SICFI únicamente le solicitará al académico el password y el login, figura 7.4.



Facultad de Ingeniería  Sistema de Información Curricular de la Facultad de Ingeniería

**Sistema de Información Curricular de la Facultad de Ingeniería**

**Ingresar Login y Password**

login

Password

Confirmar Password

**Registro de nuevos usuarios**

CUERP MEAC740404HDFDRF026EPI-494848

Número de expediente null

Nombre

Apellido Paterno

Apellido Materno

RFC

Fecha de nacimiento  -  -  ejemplo: 1975-05-25

Sexo  Masculino  Femenino

Estado civil CASADO

Nacionalidad

Lugar de nacimiento

Página WEB

Número de teléfono celular

Realizar alta Registrar

Figura 7.3 Alta

Facultad de Ingeniería  Sistema de Información Curricular de la Facultad de Ingeniería

**Sistema de Información Curricular de la Facultad de Ingeniería**

Bienvenido al Sistema de Información Curricular de la Facultad de Ingeniería.

Si es la primera vez que utilizas esta aplicación o aún no te has registrado, puedes realizar tu registro [aquí](#).

**Identificación**

Login

Password

Aceptar

Figura 7.4 Logout2

## 7.2.1 Secciones del Sistema

Los menús que conforman el sistema SICFI son los siguientes:

### 7.2.1.1 Modulo del Académico

- Datos Generales
- Formación Académica
- Programas de Estímulos
- Premios, distinciones y reconocimientos
- Trayectoria Laboral
- Productividad Académica
- Otras Actividades
- Actividades Profesionales
- Participación Institucional
- Impresión de Reportes Simples

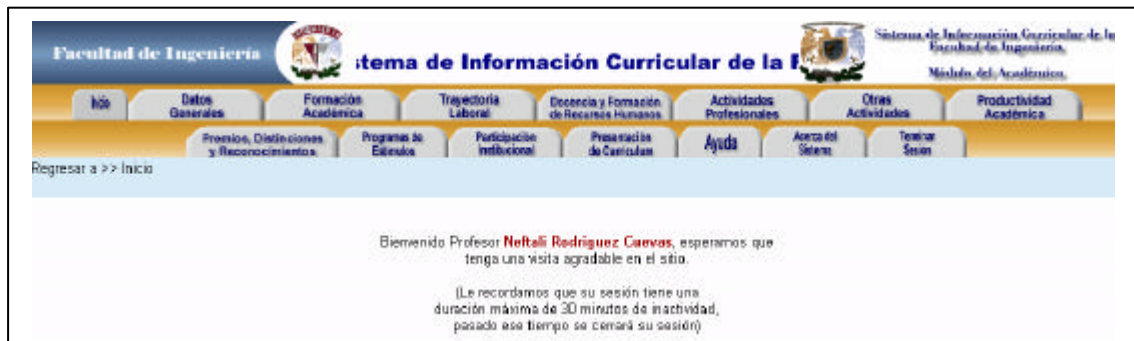


Figura 7.5 Inicio

Todos los menús cuentan con la misma estructura, como por ejemplo:

- Datos Generales
  - Alta
  - Consulta
  - Actualización

- Formación Académica
  - Alta
  - Consulta
  - Actualización
- Programas de Estímulos
  - Alta
  - Consulta
  - Actualización

De forma análoga funcionen el resto de los menús.

De lo anterior se deriva que podemos explicar los puntos principales de un menú (Altas y Consulta) con el propósito de generalizar para todos los menús restantes. Para nuestro caso vamos analizar la opción “Datos Generales”.

Si seleccionamos la opción “Datos Generales”, figura 7.6:



Figura 7.6 Modulos

El sistema SICFI desplegará a su vez un listado de opciones o submenús (figura 7.7). Cada uno de estos submenús tiene a su vez más opciones. Por ejemplo si seleccionamos la opción “Direcciones” (figura 7.8), el sistema SICFI desplegará una pantalla (figura 7.9), en la que se muestran a su vez las opciones Alta y Consulta



Figura 7.7 Menú modulos



Figura 7.8 Menú modulos<sub>2</sub>

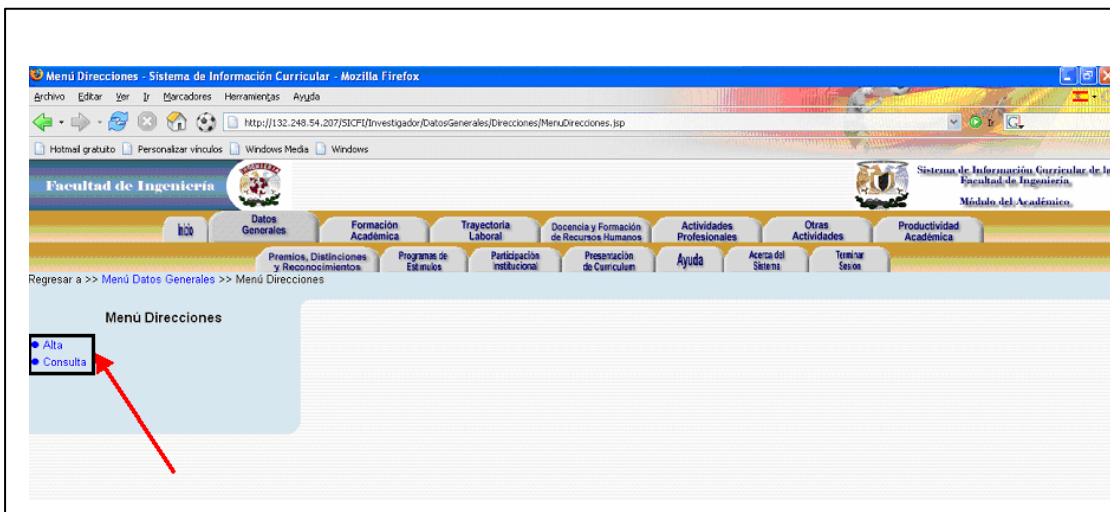


Figura 7.9 Menú modulos<sub>3</sub>

### 7.2.1.1 Alta de Datos

La secuencia que sigue el sistema al seleccionar la opción “Alta” señalada anteriormente es la siguiente:

El académico emite la petición de “Alta” de datos al sistema SICFI; y este a su vez muestra la página de ingreso de datos, figura 7.10.



The screenshot shows the SICFI system interface for data entry. At the top, there is a navigation bar with the following menu items: Inicio, Datos Generales, Formación Académica, Trayectoria Laboral, Docencia y Formación de Recursos Humanos, Actividades Profesionales, Otras Actividades, and Productividad Académica. Below this, there is a secondary navigation bar with: Premios, Distinciones y Reconocimientos, Programas de Estudios, Participación Institucional, Presentación de Currículum, Ayuda, Acceso al Sistema, and Tareas Seleccionadas. The main content area is titled "Direcciones" and "Alta". A yellow warning box states: "NOTA: Los campos con \* son indispensables de llenar". Below this, there is a form field for "Tipo de Dirección\*" with three radio button options: "Particular", "Interna UNAM", and "Externa UNAM". A "continuar" button is located at the bottom right of the form.

Figura 7.10 Alta

A continuación el académico selecciona alguna de las opciones y empieza a capturar la información referente a este campo que el sistema le solicita. Una vez ya concluida la captura por parte del académico este le indicará al sistema que ha concluido esta tarea y a continuación en respuesta a esta petición el sistema procesará la información para verificar su validez bajo los criterios establecidos desde un inicio y posteriormente validará ó emitirá un mensaje de error en el caso de que la información capturada es errónea o incompleta. En caso de que el sistema rechace esta información emitirá un mensaje de error indicando donde la información es errónea y lo regresará nuevamente a la pantalla de captura para la corrección de esta información, figura 7.11. Por otra parte, si la información es correcta; el sistema almacenará la información nueva y emitirá un mensaje de éxito con respecto a la captura de esta información al académico.

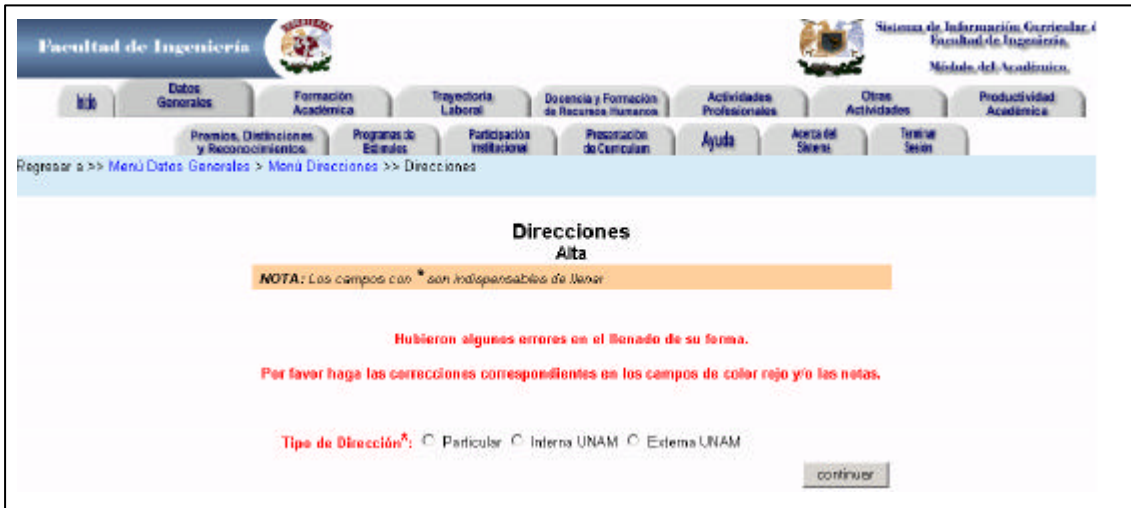


Figura 7.11 Alta error

### 7.2.1.2 Consulta de Datos

La secuencia que sigue el sistema al seleccionar la opción “Consulta” señalada a continuación es la siguiente:

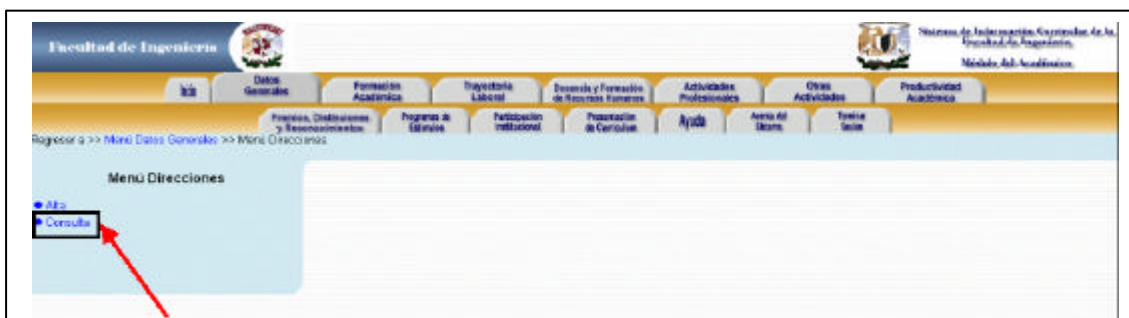


Figura 7.12 Consulta

El académico emite la petición de “Consulta” de datos al sistema SICFI; y este a su vez muestra la página de consulta de datos. En esta parte, el académico deberá seleccionar la opción que quiera consultar. Esta acción se muestra señalada en color rojo en la figura 7.13, Una vez seleccionada una opción, el sistema desplegará los datos correspondientes, figura 7.14

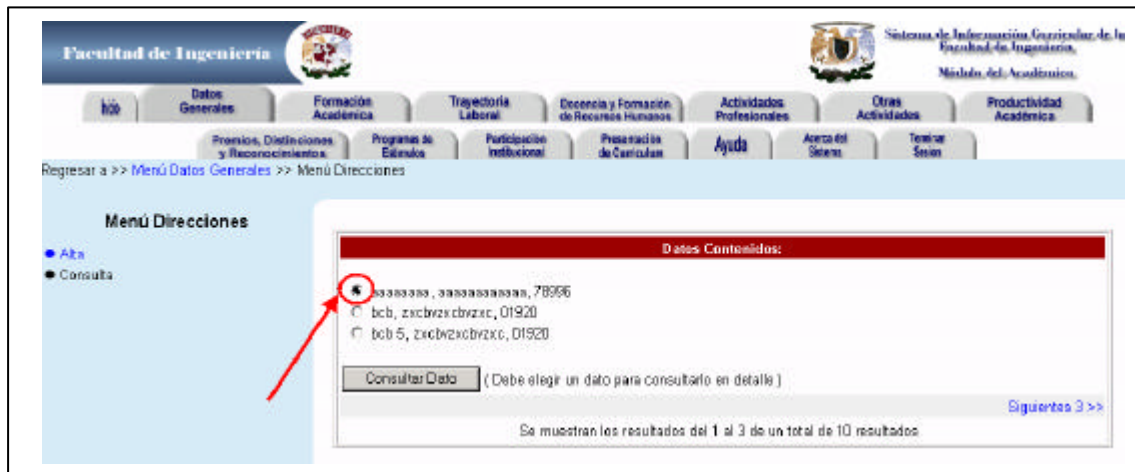


Figura 7.13 Consulta selección



Figura 7.14 Consulta editar

Aquí termina el proceso de “Consulta” por lo que el académico puede seleccionar otra opción.

### 7.2.1.3 Actualizar Datos

Por otra parte, si el académico pretende “Actualizar” Información entonces la secuencia que sigue el sistema para realizar esta acción es la siguiente:

Para iniciar con el proceso de “Actualización” el académico deberá emitir la petición de “Actualización” de datos al sistema SICFI presionando el botón “Editar” el cual se encuentra señalado en color rojo mostrado en la figura 7.14. El sistema a su vez muestra la pagina con los datos que tiene almacenados y habilita los campos necesarios para llevar acabo la actualización. En la siguiente imagen se muestra la consecuencia de presionar el botón “Editar”. Ahora ya están habilitados dos botones; uno llamado “Modificar” cuya función es la de seleccionar dentro de las opciones a la cual se le harán modificaciones (por ejemplo dentro de los tipos de direcciones existentes; están Dirección particular, Dirección interna UNAM o Dirección Externa UNAM). Y otro botón llamado “Guardar” cuya función es la de validar la información, por parte del académico una vez ya modificada y de esa manera actualizar la base de datos del sistema con la información nueva.

**Direcciones**  
**Edición**

**NOTA:** Los campos con \* son indispensables de llenar

Tipo de Dirección\*: INTERNA UNAM

Calle y Num.\*: aaaaaaaaa

Colonia\*: aaaaaaaaaaaaa

Delegación (ó Municipio)\*: aaaaaaa C.P.\*: 78996

Estado\*: aaaaaaaaaaaaa

Pais\*: México

Teléfonos:

Teléfono 1 (sólo dígitos):  Extensión de Tel. 1:

Teléfono 2 (sólo dígitos):  Extensión de Tel. 2:

Fax:

Correos Electrónicos:

Correo Electrónico 1:

Correo Electrónico 2:

Figura 7.15 Consulta modificar

Por ejemplo supongamos que en la pantalla anterior decidimos cambiar los campos: Código Postal actual, Calle y Número, Colonia, Delegación ó Municipio, Estado, y correo electrónico 2; por los



siguientes: 78900, AVENIDA INSURGENTES 545, Colonia Nápoles, Benito Juárez, México DF y [neftali666@yahoo.com.mx](mailto:neftali666@yahoo.com.mx) respectivamente.

Una vez que el académico termino de hacer los cambios en la información; deberá indicarle al sistema que desea cambiar la información actual oprimiendo el botón “Guardar” como se muestra a continuación.

**Direcciones Edición**

**NOTA:** Los campos con \* son indispensables de llenar

Tipo de Dirección\*: INTERNA UNAM modificar

Calle y Num.\*: AVENIDA INSURGENTES

Colonia\*: COLONIA NAPOLES

Delegación (ó Municipio)\*: BENITO JUARE C.P.\*: 78900

Estado\*: MEXICO DF

País\*: México

Teléfonos:

Teléfono 1 (sólo dígitos):  Extensión de Tel. 1:

Teléfono 2 (sólo dígitos):  Extensión de Tel. 2:

Fax:

Correos Electrónicos:

Correo Electrónico 1:

Correo Electrónico 2:

**Guardar**

Figura 7.16 Guardar

A continuación el sistema en respuesta a esta petición de validación de la información por parte del académico; procesará la información para verificar su validez bajo los criterios establecidos desde un inicio y posteriormente validará o emitirá un mensaje de error en el caso de que la información capturada es errónea o incompleta. En caso de que el sistema rechace esta información emitirá un mensaje de error en el cual le indicará al académico cual es el punto ó que información es errónea y lo regresará nuevamente a la pantalla de captura para la corrección de esta información.

Por otra parte, si la información es correcta; el sistema almacenará la información nueva y emitirá un mensaje de éxito con respecto a la captura de esta información al académico.

Las acciones principales del sistema que explicamos en párrafos anteriores son la de “Alta”, “Consulta” y “Actualización” de datos, todos los menús están conformados de manera muy similar por lo que aquí dimos un bosquejo general de cómo funciona el sistema y de los menús en general.

## 7.2 Manual del Administrador.

Para el caso del administrador; el sistema SICFI únicamente se le solicitará el password y el login como en el caso del Académico sin embargo la validación del nivel correspondiente al administrador le llevara a la siguiente página:

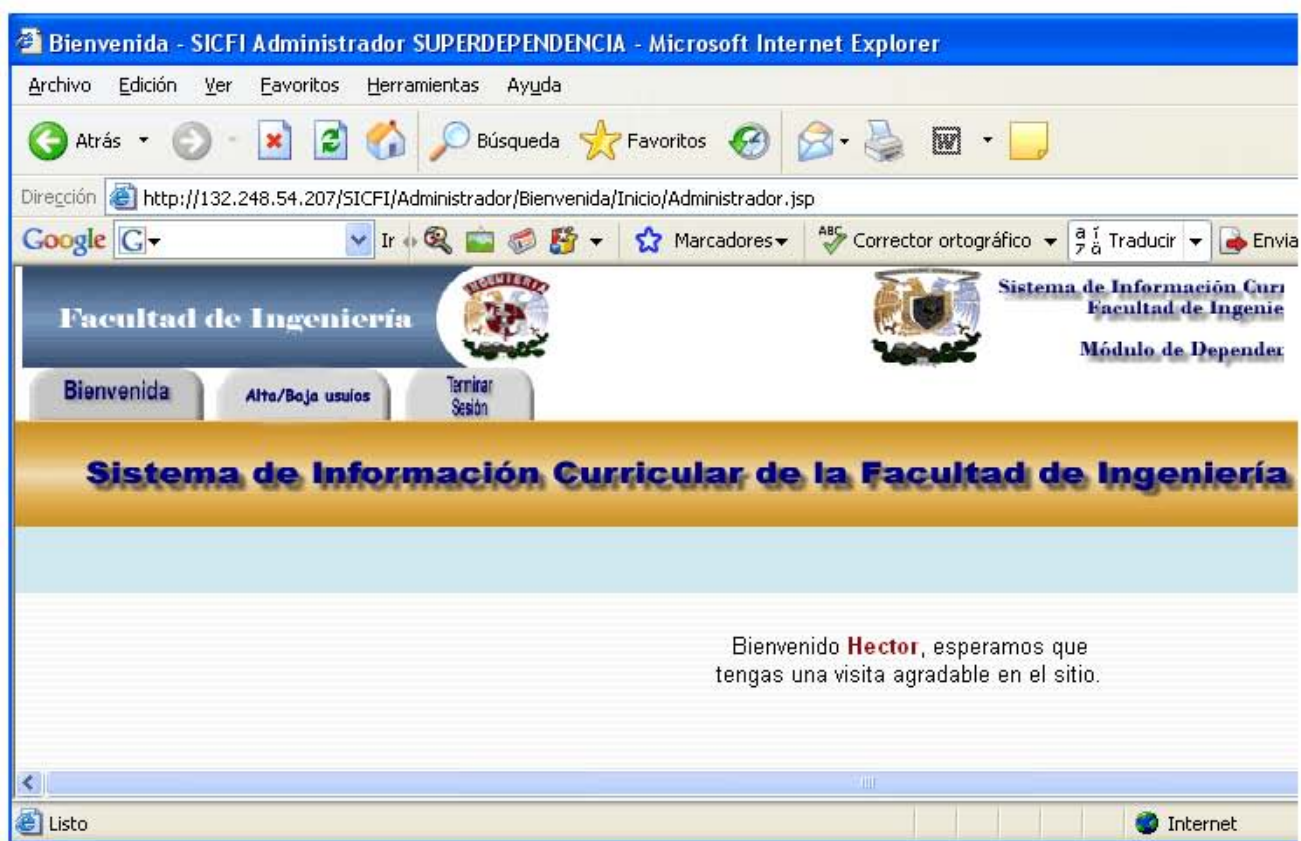


Figura 7.17 Administrador

Este módulo contara con las opciones para dar de Lata y baja de usuarios así como con la opción de salir del sistema, opción que remitirá al usuario a la página de entrada.

Para el caso del alta y baja de usuarios se tendrá siguiente página, en donde el usuario deberá de ingresar tanto la CURP como el número de trabajador correspondiente a usuario que ha de ser agregado o eliminado del sistema, bajo las siguientes consideraciones:

- Alta de Usuarios:
  - En el caso de que el usuario ya haya sido dado de alta, se indicara al administrador que el usuario ya esta registrado en el sistema. En caso de que no sea el caso se pedirá de forma simultánea que revise los datos ingresados y en caso de que los datos del usuario sean los correctos, el administrador deberá de realizar las comprobaciones pertinentes directamente en la base de datos, bajo una estricta revisión que permita mantener la integridad de los datos.
  - En caso de que el usuario sea nuevo y no haya problema con la CURP ingresada ni con el número de trabajador, al momento de realizar el alta del usuario se desplegara un mensaje, que le indicara al usuario que el registro se ha efectuado.
- Baja de Usuarios
  - En el caso de que el usuario no exista, se indicara al administrador que el usuario no se encuentra registrado en el sistema. En caso de que no sea el caso se pedirá de forma simultánea que revise los datos ingresados y en caso de que los datos del usuario sean los correctos, el administrador deberá de realizar las comprobaciones pertinentes directamente en la base de datos, bajo una estricta revisión que permita mantener la integridad de los datos.
  - En caso de que los datos ingresados, CURP y el número de trabajador, sean los correctos al momento de realizar la baja del usuario se desplegara un mensaje, que le indicara al usuario que se ha eliminado al usuario del sistema.

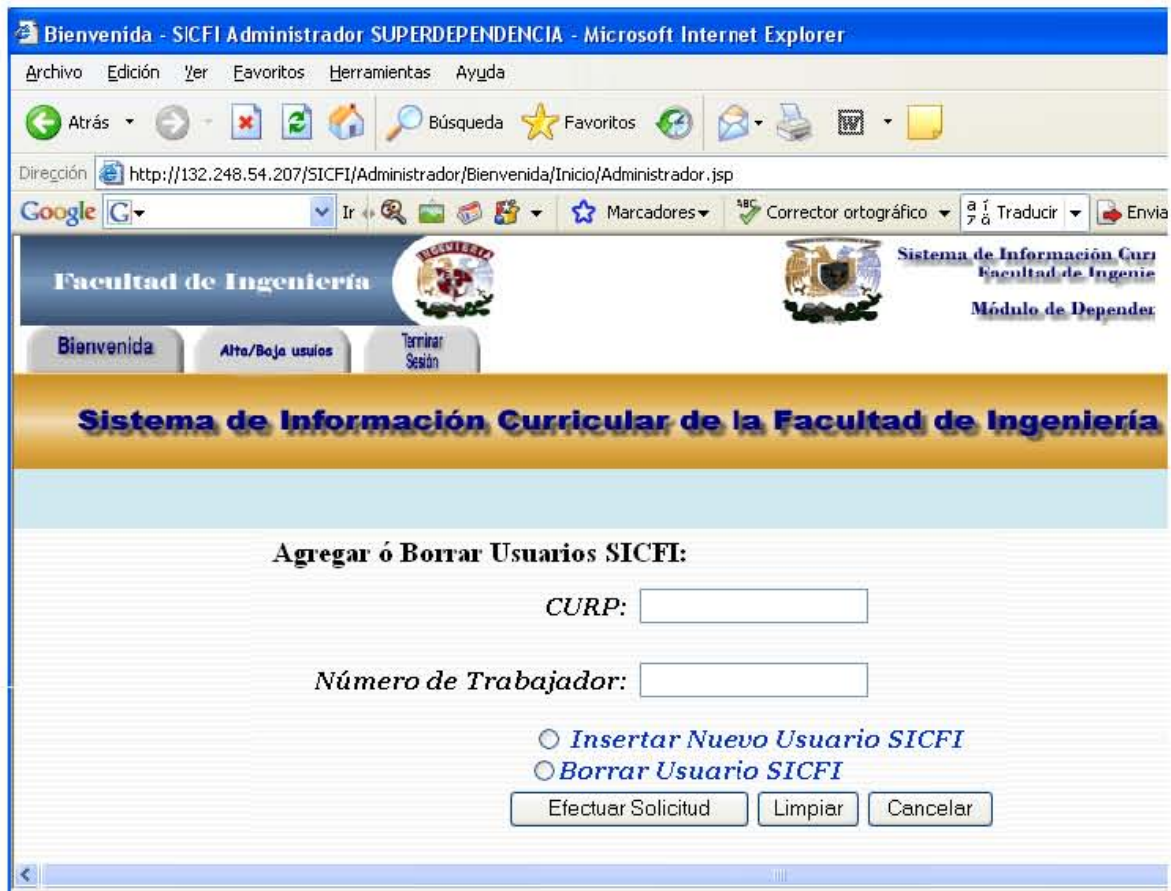


Figura 7.18 Administrador alta

## **8. CONCLUSIONES.**

La investigación realizada señala que las universidades juegan un papel muy importante en el desarrollo de la educación. La UNAM, en específico la Facultad de ingeniería, no está exenta de ello. Al hablar de la internacionalización se presenta un verdadero reto y proceso de transformación en cuanto a mejorar sus planes de desarrollo, programas, procedimientos, y políticas, que implica que la institución deba ser más competitiva al ofrecer una formación más extensa y de calidad a sus estudiantes.

Para ello, se requiere fortalecer los esquemas de actualización, protección y ampliación de las tecnologías de información y comunicación, consolidando con esto los procesos de gestión académica y administrativa así como los modelos de enseñanza-aprendizaje empleados actualmente en la Facultad.

En parte, la solución a este tipo de crecientes necesidades de la institución, implica el poder administrar de forma eficiente los recursos con los que se cuenta, en especial la planta docente encargada de la formación de los estudiantes así como de múltiples proyectos de investigación, equipamiento e infraestructura dentro de la UNAM

Centrándonos en este punto, conocer a dicho personal, su formación académica y laboral, nos permite saber el tipo de profesionistas con los que contamos, permitiendo casi en forma transitiva, la asignación de actividades específicas o proyectos de acuerdo al perfil que demanden dichas tareas o proyectos.

De aquí la importancia y relevancia del diseño y desarrollo del Sistema de Información Curricular para la Facultad de Ingeniería, sistema, que como su nombre lo indica, nos permitirá tener acceso a la información detallada y actualizada del personal académico de la Facultad de ingeniería.

Se espera que el impacto dentro de la Facultad sea de gran relevancia, ya que la obtención de la currícula de los académicos sería obtenida de una manera expedita y confiable. Además se podría disponer en cualquier momento de la información detallada de la planta docente que conforman a cada una de las áreas de la Facultad de Ingeniería, permitiendo la toma de decisiones de una forma mas fehaciente e inmediata por parte de los directivos para poder asignar el personal adecuado a las

tareas o proyectos que requieran un perfil específico dentro de los diferentes ámbitos de la Facultad de Ingeniería ó de la propia UNAM, tareas que requieren de personas calificadas para ser llevadas a cabo.

Por otra parte, conocer tanto la formación académica como laborar de personal que conforma la planta docente de la facultad, implican poder realizar una mejor evaluación del mismo, con el fin de brindarle los mejores apoyos (premios, estímulos, reconocimientos etc.) con base en su desarrollo profesional, motivándoles con el fin de que puedan brindar más de su conocimiento y experiencia para beneficio no sólo de la Facultad y de la UNAM en sí, si no también en beneficio de los nuevos profesionistas que se forman actualmente en las aulas de esta gran institución, brindando un gran beneficio a la sociedad.

De esta forma podemos decir que el Sistema de Información Curricular para la Facultad de Ingeniería (SICFI), aún cuando es una herramienta en apariencia sencilla, que se encarga de recabar la información personal, académica y laboral de la plantilla docente; dentro de sus alcances se tienen el poder proporcionar información actualizada, completa y estandarizada, que puede ser capturada, importada y exportada a otros sistemas o herramientas con el fin de ser más explotada, tal es el caso de la Memoria Estadística, el sistema ARIES o el banco de datos del CONACYT. Sistemas que cuentan con información en común, la cual puede ser aprovechada por el Sistema de Información Curricular.

Tomando en cuenta que toda la información contenida en estos sistemas ya ha sido previamente validada y actualizada, el Sistema de Información Curricular no tendría información duplicada y el académico sólo tendría que revisar periódicamente su información y lo mejor es que solamente tendría un solo currículo, el cual estará disponible para las entidades que estén interesadas en él, las cuales solo tendrían que buscar su información en el SICFI y tomar las decisiones sobre que personal es el que cubre sus expectativas.

Más aún, la presentación de los formatos institucionales para premios y estímulos, por ejemplo PRIDE, PAPIME, etc., por medio del Sistema de Información Curricular, permitiría minimizar el tiempo que el académico emplea para su generación al mismo tiempo que se eliminaría la duplicidad de información.

Es así que el sistema como tal, tiene como fin primordial el proporcionar información íntegra y veraz a quien lo solicite, información que se hallará disponible en cualquier momento, aún más, dadas las características del diseño del mismo, se cuenta con un sistema portable, diseñado en módulos separados que interactúan entre sí, dichas características ofrecen capacidades de crecimiento según lo requiera las necesidades administrativas de cada una de las entidades que conforman a la Facultad de Ingeniería.

Estas características fueron alcanzadas gracias al uso de las más recientes herramientas de programación, así como las tecnologías más avanzadas dentro de Cliente/Servidor

En la liberación del sistema, podemos decir que el sistema que se entrega cumple con la normatividad informática que actualmente se pide en todos y cada uno de los sistemas de Información a nivel comercial y con una integración profesional, el decir además que el sistema fue implantado con calidad, esto es fue entregado en tiempo y forma en cuanto a las necesidades del cliente (Facultad de Ingeniería)

Hacemos énfasis que el sistema puede ser mejorado y actualizado en un futuro cuando así se requiera, además de que puede recibir la información de manera directa y en tiempo real, en el momento en el que se alimenta el sistema de información, esta puede ser procesada casi inmediatamente si así se necesitara. Tomando en cuenta que el desarrollo es basado para la Web, podemos hacer que el sistema pueda ser utilizado alrededor del mundo.

Finalmente concluimos que el realizar sistemas de información, es importante en los procesos de automatización, dentro y fuera de la UNAM.

## 9. BIBLIOGRAFIA

- (1). Zorrilla Arena, Santiago Torres Xammar (2003). Guía para elaborar Tesis (2ª.Ed.) México, Océano Pág. 62 y 63.
- (2). Plan de Desarrollo 2003-2007 Facultad de Ingeniería
- (3). Programa de Mejoramiento al Profesorado, [http://: promep. sep. gob. mx](http://promep.sep.gob.mx)
- (4) Zermeño Flores, Ana Isabel (2002) Currículum Vitae
- (5) Pressman, Roger S. (2002). Ingeniería del Software: Un enfoque práctico; Quinta edición. McGraw-Hill, Madrid, España.
- (6) Charre Ojeda, Francisco. La biblia de HTML. Anaya Multimedia. Madrid, España.
- (7) Delgado Sanz, Soledad. JavaScript. McGraw-Hill. Madrid, España.
- (8) Lewis, Bil & J. Berg, Daniel. Programación Multithread en Java. Madrid, España.
- (9) Flanagan, David. Java In A Nutshell, 5th Edición. O'Reilly.
- (10) M. Burke, Eric. Java and XSLT, O'Reilly.
- (11) E. Johnson, Ralph A Little Java, a Few Patterns. MIT Press.
- (12) Bergsten, Hans. Java Server Pages, tercera edición, O'Reilly.
- (13) W. Perry, Bruce. Java Servlet & JSP Cookbook. O'Reilly.
- (14) Moodie, Matthew. Pro Apache Tomcat. Apress