

01132  
60



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

---

---

FACULTAD DE INGENIERÍA.

SISTEMA DE CONTROL DE EVALUACIÓN DE  
DOCENCIA Y DESEMPEÑO ACADÉMICO  
POR INTERNET.

T E S I S

QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACIÓN.

P R E S E N T A N

MARTÍNEZ LÓPEZ ROSALBA.  
PÉREZ LÓPEZ GRACIELA.  
TREJO MUNGUÍA ISABEL.

DIRECTOR DE TESIS

ING. JUAN MANUEL MARTÍNEZ VILLALOBOS.



MÉXICO, D.F.

2004

A



Universidad Nacional  
Autónoma de México



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

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

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

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

## ***AGRADECIMIENTOS***

A mis padres y hermanos,  
que me han dado su amor y apoyo en todo momento.

A mi director de tesis,  
al Ing. Juan Manuel Martínez Villalobos  
por su paciencia y apoyo en el desarrollo del presente trabajo.

A mi Señor y Salvador Jesucristo,  
por su infinito amor y eterna misericordia.

*"Porque de tal manera amó Dios al mundo,  
que ha dado a su Hijo unigénito,  
para que todo aquel que en él cree,  
no se pierda, mas tenga vida eterna."*

*Juan 3:16*

GRACIAS.

Rosalba

B

## AGRADECIMIENTOS

### A MI PADRE:

*Gracias padre por creer en mi y haberme brindado tu cariño y confianza, por apoyarme en las decisiones que he tomado y motivarme a seguir adelante.*

*Te quiero mucho.*

*Gracias Padre por existir.*

### A MI MADRE:

*Gracias Corazón*

*Por darme la vida, tu cariño, ternura y comprensión, por enseñarme las cosas bellas de la vida y ayudarme constantemente a ser perseverante.*

*Te adoro chiquita.*

*Gracias por ser mi Madre.*

### A MIS HERMANOS:

*Gracias por tener la dicha de estar con ustedes de compartir su sensibilidad, ternura y amor les agradezco todo el apoyo incondicional que me han brindado.*

*Gracias, los quiero mucho.*

### A MIS ABUELOS:

*Aunque ya no están, pero les doy las gracias por su bondad, humanidad y sencillez.*

*Por ese amor tan grande que tuvieron para nosotros.*

*A MIS SOBRINOS:*

*Gracias por cada momento de alegría  
que han llevado a nuestro hogar y  
espero que esa sensibilidad que los  
caracteriza perdure por siempre.*

*ING. JUAN MANUEL:*

*Que gracias a su tiempo y dedicación  
hizo posible que esta tesis se terminara.*

*A MIS CUÑADOS Y AMIGOS:*

*Con los cuales he contado incondicionalmente  
y en momentos difíciles apoyaron mis decisiones  
y recibí de ellos una palabra de aliento.*

*HONDO, MUY HONDO,  
DENTRO DEL CUERPO HABITA EL ALMA,  
NADIE LA HA VISTO NUNCA  
PERO TODOS SABEN QUE EXISTE.  
Y NO SOLO SABEN QUE EXISTE,  
SABEN TAMBIEN LO QUE HA Y EN SU INTERIOR.*

*MIJAL SNUNIT*

*GRACIELA PÉREZ LÓPEZ*

# **AGRADECIMIENTOS.**

A la Universidad Nacional Autónoma de México:

Por haberme brindado una formación profesional y la oportunidad de llevarme los más hermosos recuerdos.

A mis padres y hermanos:

Gracias por esperar con paciencia, la búsqueda de una solución que me tomó más tiempo del esperado, pero he terminado y ustedes han contribuido a la elaboración de cada uno de mis proyectos, y ahora finalmente participaron en mi proyecto de tesis, especialmente:

Mi mamá:

Por estar presente en cada etapa que he vivido, siempre preocupada por qué no abandone las cosas sin antes terminarlas, gracias por ayudarme a lograr cada uno de mis objetivos, eres la persona más importante que tengo.

Mi papá:

Por aguardar el momento en que pueda decirte, gracias por venir y estar conmigo.

José Luis:

Por qué como siempre, con tus valiosas aportaciones realizadas de principio a fin, logré encontrar una solución a los problemas con los que me enfrente, durante la realización de este proyecto de tesis.

Daniel:

Gracias por el gran apoyo que siempre he encontrado en ti.

**Cecilia:**

Por recordarme que el tiempo es tan breve que se va a cada instante, y de pronto no hemos terminado lo que debía ya estar concluido.

**A mi director de tesis:**

Ing. Juan Manuel Martínez Villalobos por su confianza, paciencia y apoyo brindado, facilitando la inicialización y culminación de este proyecto de tesis.

**A los profesores:**

Que resolvieron pacientemente mis dudas, haciendo posible el seguimiento de este proyecto, especialmente a Carlos A. Román Zamitiz por su amabilidad ofrecida, cada vez que me acerque para solicitarle alguna consulta.

**A todas aquellas personas:**

Que me proporcionaron amablemente el material necesario, para el desarrollo de este trabajo de tesis.

GRACIAS.

**ISABEL TREJO MUNGUÍA.**

F

# ÍNDICE.

---

OBJETIVO.....	1
INTRODUCCIÓN.....	2
<b>CAPÍTULO I ANTECEDENTES.</b>	
I.I ANTECEDENTES .....	3
I.II EVALUACIÓN AL PROFESOR.....	6
I.III ALUMNO .....	14
I.IV PARÁMETROS A EVALUAR.....	15
<b>CAPÍTULO II INGENIERÍA DEL SOFTWARE.</b>	
II.I INGENIERÍA DEL SOFTWARE.....	17
II.II PARADIGMAS DE LA INGENIERÍA DEL SOFTWARE.....	18
II.II.I CICLO DE VIDA CLÁSICO.....	18
II.II.II PROTOTIPO.....	19
II.II.III MODELO EN ESPIRAL.....	21
II.II.IV TÉCNICAS DE CUARTA GENERACIÓN (4GT) .....	23
II.II.V METODOLOGÍA UTILIZADA .....	24
<b>CAPÍTULO III ANÁLISIS DEL SISTEMA.</b>	
III.I REQUERIMIENTO DE INFORMACIÓN .....	25
III.II BASES DE DATOS .....	35
III.II.I MODELO ENTIDAD-RELACIÓN .....	35
III.II.II NORMALIZACIÓN .....	39
III.II.III DIAGRAMA RELACIONAL DE BASE DE DATOS .....	40
III.II.IV DICCIONARIO DE DATOS .....	42
III.III ANÁLISIS DEL SISTEMA .....	43
III.IV ANÁLISIS DEL REQUERIMIENTO .....	53
<b>CAPÍTULO IV DISEÑO DEL SISTEMA.</b>	
IV.I PRINCIPIOS DEL DISEÑO DE SOFTWARE .....	62
IV.II ARQUITECTURA DE LA APLICACIÓN .....	67
IV.III ESTRUCTURA DE PROCESOS .....	75
IV.IV INTERFAZ CON EL USUARIO .....	92

**CAPÍTULO V CONSTRUCCIÓN.**

V.I CONSIDERACIONES DEL AMBIENTE DE DESARROLLO.....93  
V.II CONSTRUCCIÓN DE LA BASE DE DATOS .....94  
V.III CODIFICACIÓN UTILIZANDO ACCESS Y JAVA .....98  
V.IV SEGURIDAD .....102

**CAPÍTULO VI ENFOQUE PARA LA PRUEBA DEL SOFTWARE.**

VI.I VERIFICACIÓN Y VALIDACIÓN.....106  
VI.II PRUEBA DEL SISTEMA .....109  
VI.III PRUEBA DE SEGURIDAD .....112

**CAPÍTULO VII MANTENIMIENTO Y DOCUMENTACIÓN.**

MANTENIMIENTO Y DOCUMENTACIÓN .....113  
CONCLUSIONES .....140  
BIBLIOGRAFÍA .....141

H

## **OBJETIVO.**

---

La actualización de un sistema de cómputo de la Facultad de Medicina Veterinaria y Zootecnia, que permita mejorar y agilizar el proceso de evaluación de profesores actualmente utilizada; de tal manera que el alumno pueda acceder de forma sencilla y rápida vía internet.

# INTRODUCCIÓN.

---

El desarrollo de este trabajo de tesis, consistirá en la actualización del sistema de cómputo de evaluación de profesores de la Facultad de Medicina Veterinaria y Zootecnia, de tal manera que el alumno pueda acceder de forma sencilla y rápida, a través de una computadora conectada a Internet.

Con la implementación de este sistema se pretende satisfacer los requerimientos del usuario en cuanto a disponibilidad, distribución de información y acceso; necesidades que ya no son cubiertas por el sistema actualmente utilizado.

La realización de este trabajo, se llevó a cabo en una serie de etapas que a continuación se describen.

En el **capítulo I** se efectúa una investigación detallada del entorno en el que opera el sistema antiguo, utilizando todos los recursos necesarios que permitan ejemplificar de una manera clara esta situación.

En el **capítulo II** se mencionan los paradigmas de la ingeniería de software, que consisten en diversos métodos, caminos o procedimientos mediante los cuales puede desarrollarse un sistema.

En el **capítulo III** se inicia con la etapa de análisis del sistema, estableciendo primeramente los requerimientos de información, necesarios para lograr un mejor entendimiento durante esta fase.

Para efectuar el planteamiento del problema se describe el funcionamiento del antiguo sistema, de esta forma se determinan las necesidades por cubrir y las herramientas de desarrollo para el nuevo sistema.

En el **capítulo IV** de diseño se utilizan las herramientas mencionadas en el capítulo anterior, para documentar y efectuar diagramas. Se muestra el diseño de la base de datos, modelos y objetos del sistema.

En el **capítulo V** construcción, se construye la base de datos y la programación de los objetos mencionados en el capítulo IV.

En el **capítulo VI** se reportan los resultados correspondientes a las pruebas realizadas al sistema, así como la instalación y configuración de las herramientas necesarias para su funcionamiento.

En el **capítulo VII** se incluye toda la documentación requerida para lograr un buen entendimiento del desarrollo y funcionamiento del sistema.

# **CAPÍTULO I**

## **ANTECEDENTES.**

---

### **I.1 ANTECEDENTES.**

Dentro del proceso enseñanza-aprendizaje, es común la evaluación realizada por el profesor hacia el alumno, pero efectuar la evaluación de manera inversa, es decir, del alumno hacia el profesor requiere de un análisis detallado para plantear todo el conjunto de factores que serán involucrados.

En febrero de 1997, la Universidad Nacional Autónoma de México acordó la creación de la Dirección General de Evaluación Educativa (DGEE), a la cual encomendó el desarrollo de marcos de referencia, programas, métodos, instrumentos y estudios de evaluación indispensables para mejorar la función educativa de la Universidad.

Para atender el cumplimiento de sus objetivos, la Dirección General de Evaluación Educativa ha establecido cuatro áreas de trabajo; una dedicada al desarrollo de instrumentos e interpretación de los resultados de los exámenes de admisión, diagnóstico y egreso; otra orientada a explorar sistemáticamente aspectos particulares del proceso educativo que permitan una comprensión objetiva de la trayectoria escolar, el egreso, la titulación, la ubicación y desempeño de los egresados en el trabajo; una más destinada al apoyo de las diversas acciones de reflexión institucionales que exige la evaluación de procesos y programas académicos; y otra dedicada al impulso y coordinación de programas institucionales de apoyo y desarrollo educativo.

El área de Evaluación de procesos y programas académicos, a solicitud de las diversas Facultades proporcionan:

1. Asesorar en el diseño y métodos e instrumentos para evaluar a la docencia.
2. Asesoría en el diseño, operación e interpretación de resultados de procesos de licenciatura y posgrado.
3. Talleres sobre evaluación de la docencia y evaluación de programas académicos.

Actualmente en colaboración con la Dirección General de Evaluación Educativa, las Facultades de Arquitectura, Medicina, Química, Medicina Veterinaria y Zootecnia, Ciencias en el área de Biología, Posgrado de Neurobiología y Posgrado de Ciencias Biológicas aplican un cuestionario de evaluación al profesor para saber el desempeño académico de su Facultad o Institución.

Existen otras Facultades, por ejemplo Ingeniería y Contaduría, entre otras, que realizan sus evaluaciones sin la colaboración de la DGEE, debido a que cuentan con el personal especializado que elabora los cuestionarios y procesa los resultados.

Los rubros utilizados para la elaboración de los cuestionarios aplicados se basan en estrategias pedagógicas de enseñanza-aprendizaje como son:

- Evaluación.
- Cumplimiento.
- Conocimiento.
- Motivación.
- Responsabilidad y,
- Actualmente apoyo a la formación de alumnos.

Cada una de estas características que permitirán evaluar al profesor, fueron establecidas de acuerdo a una encuesta realizada a profesores y alumnos, para determinar las cualidades que debe tener un buen docente.

En la Facultad de Medicina Veterinaria y Zootecnia, el proceso de evaluación dio inicio en 1975, con la creación de una sección didáctica, cuyo objetivo fue el de apoyar a los profesores en su práctica docente y la evaluación sólo se efectuaba a solicitud de los mismos.

Tomando como base estos puntos, se concluyó en la Facultad de Medicina Veterinaria y Zootecnia establecer un sistema, donde el profesor fuera evaluado por el alumno, por sus colegas, por su jefe inmediato y por la administración de la Facultad.

En 1990 se implementó un cuestionario de evaluación al profesor, para otorgar el premio a la docencia, "El premio Manuel Cabrera Valtierra", utilizado sólo por 2 años debido a que los resultados ya no eran viables.

Como respuesta a esta problemática, se formó una comisión que propusiera un proyecto de evaluación de los profesores, por los alumnos.

El sistema de evaluación resultante consistió en 7 etapas.

1. Una encuesta para determinar según la opinión de profesores y alumnos las características de un buen docente.
2. Determinación de las características a evaluar a partir de la encuesta realizada.
3. Elaboración de un cuestionario preliminar.
4. Validación de ese cuestionario.
5. Corrección de errores detectados en el proceso anterior.
6. Aplicación del cuestionario.
7. Detección de factores que influyen en la calificación y ajuste.

Estos son los pasos que se realizarán para la elaboración de los cuestionarios, que posteriormente deberán ser procesados para conocer los resultados y proporcionarlos al profesor correspondiente.

A continuación se ejemplifica en el diagrama 1 el proceso mencionado anteriormente.

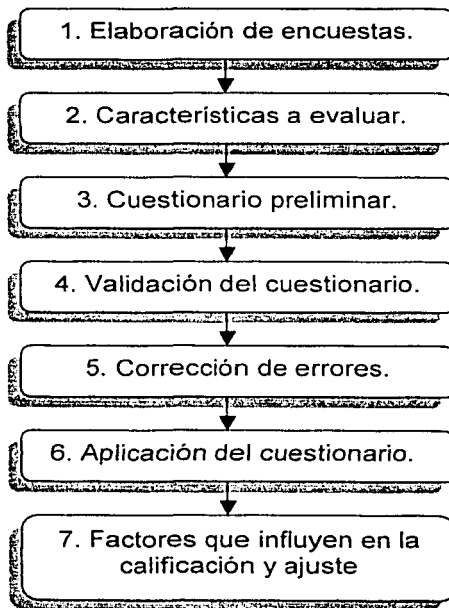


Diagrama 1. Etapas del Sistema de Evaluación.

TESIS CON  
FALLA DE ORIGEN

### **I.II EVALUACIÓN AL PROFESOR.**

El objetivo principal de una evaluación es retroalimentar el proceso enseñanza-aprendizaje; esto significa que los datos obtenidos en la evaluación servirán a los que intervienen en dicho proceso (docentes-alumnos) en forma directa, para mejorar las deficiencias que se presenten en la realización del proceso e incidir en el mejoramiento de la calidad; y en consecuencia el rendimiento en el Proceso Enseñanza-Aprendizaje.

La Facultad de Medicina Veterinaria y Zootecnia, consciente de que el docente universitario se enfrenta al reto de proporcionar una educación de formación continua, propuso un sistema de evaluación cuyo contenido estuviera compuesto con la participación de los profesores, los cuales serían seleccionados de manera aleatoria.

Como resultado del sistema de evaluación, compuesta por 7 etapas ya mencionadas anteriormente se obtuvo el siguiente cuestionario.

**CUESTIONARIO PILOTO DE EVALUACIÓN A LA DOCENCIA.  
FACULTAD DE MEDICINA VETERINARIA Y ZOOTECNIA.  
NOMBRE DEL PROFESOR \_\_\_\_\_**

**CUESTIONARIO PILOTO DE  
EVALUACIÓN AL PROFESOR DE  
LICENCIATURA.**

**INSTRUCCIONES:**

1. No maltrate este hoja.
2. Utilice unicamente lápiz del No. 2 ó 2.5
3. En caso de error borre completamente y limpiamente
4. Marque su respuesta llenando completamente el óvalo correspondiente
5. Sólo debe marcarse una respuesta por pregunta.

MATERIA \_\_\_\_\_ GRUPO \_\_\_\_\_

- |         |         |
|---------|---------|
| 0. 0000 | 0. 0000 |
| 1. 0000 | 1. 0000 |
| 2. 0000 | 2. 0000 |
| 3. 0000 | 3. 0000 |
| 4. 0000 | 4. 0000 |
| 5. 0000 | 5. 0000 |
| 6. 0000 | 6. 0000 |
| 7. 0000 | 7. 0000 |
| 8. 0000 | 8. 0000 |
| 9. 0000 | 9. 0000 |

1) Los conocimientos que tiene el profesor sobre su asignatura son:

- Excelentes.
- Buenos.
- Regulares.
- Deficientes.

2) El profesor relacionó los temas de su curso con otros temas de la carrera.

- Siempre.
- Frecuentemente.
- Pocas veces
- Nunca.

3) El profesor compartió en clase experiencias de su vida profesional para dar a usted una visión más amplia de la carrera.

- Siempre.
- Frecuentemente.
- Pocas veces.
- Nunca.

4) Los temas del curso que el profesor domina son:

- Todos.
- Casi todos.
- Algunos.
- Ningunos.

5) La exposición que el profesor hizo de los temas fue:

- Brillante.
- Interesante.
- Aceptable.
- Aburrida.

6) El profesor lo motivo para que ampliara sus conocimientos sobre la material.

- Frecuentemente.
- Algunas veces.
- Pocas veces.
- Nunca.

**TESIS CON  
FALLA DE ORIGEN**

7) El profesor estuvo dispuesto a ayudarle a resolver sus dudas.

- Siempre.
- Frecuentemente.
- Pocas veces.
- Nunca.

8) El profesor asistió puntualmente y cumplió con el tiempo destinado a su clase.

- Siempre.
- Casi siempre.
- Casi nunca.
- Nunca.

9) El profesor expuso en forma clara los temas del programa.

- Siempre
- Frecuentemente.
- Pocas veces.
- Nunca.

10) El método de enseñanza que uso el profesor durante el curso fue:

- Excelente.
- Bueno.
- Regular.
- Malo.

11) El profesor cumplió con los criterios de evaluación que estableció al principio del semestre.

- Siempre.
- Casi siempre.
- Casi nunca.
- Nunca.

12) El trato del profesor fue imparcial y respetuoso tanto con usted como con sus compañeros.

- Siempre.
- Casi siempre.
- Casi nunca.
- Nunca.

13) La frecuencia con la que el profesor asistió al curso.

- Más del 90%.
- Entre el 75% y 90%.
- Entre el 50% y 75%.
- Menos del 50%.

14) Clasifique en términos generales al profesor.

- Excelente.
- Bueno.
- Regular.
- Malo.

ALUMNOS.

15) La frecuencia con que asistió a clases fue:

- Más del 90%
- Entre el 75% y el 90%
- Entre el 50% y el 75%

16) Yo cumplí con las tareas o lecturas asignadas por el profesor.

- Siempre.
- Frecuentemente.

17) Yo recomendaría a este(a) profesor(a) a mis compañeros.

Si.

No.

18) Considerando el esfuerzo que hice durante el curso la calificación que obtuve me pareció.

Justa.

injusta.

19) Sexo del alumno.

M

F

Si usted tiene comentarios serán de mucha utilidad:

Las primeras 12 preguntas califican al profesor, y se encuentran organizadas de la siguiente manera.

- Las preguntas 1 y 4 evalúan Conocimientos.
- Las preguntas 2 y 3 evalúan Experiencia Profesional.
- Las preguntas 5 y 6 evalúan Motivación.
- Las preguntas 9 y 10 evalúan Comunicación.
- Las preguntas 11 y 12 evalúan Justicia e Imparcialidad.
- La pregunta 7 Accesibilidad del profesor.
- Finalmente la pregunta 8 Puntualidad y asistencia.

Existen 3 preguntas que a continuación se describen, que no tienen ningún peso en la calificación, pero se utilizan para validar algunas respuestas de preguntas anteriormente realizadas.

- La pregunta 13 es una pregunta control de la 8, que nos permite validar en este caso a la pregunta 8, siempre que el resultado de ambas sea congruente.
- La pregunta 17 es una medida de popularidad del profesor.
- Finalmente la 18 es un control de la 11 y 12.

TESIS CON  
FALLA DE ORIGEN

También se colocó una pregunta de calificación global, es decir la respuesta de esta pregunta debería coincidir con la suma total, obtenida en las 12 preguntas que califican al profesor.

- La pregunta 14 es un control de todo el cuestionario.

Por último se tienen 2 controles para los alumnos, los cuales son de utilidad para validar o no el cuestionario; porque un alumno que falta el 50% o más al curso y no cumpla con las tareas asignadas, no puede evaluar al profesor.

- Las preguntas 15 y 16 son controles de autoevaluación del alumno.

Los pesos específicos de cada una de las respuestas que califican, se decidió de la siguiente manera:

- Todas las respuestas A), tienen el 100%.
- Las respuestas B), las 2/3 partes del 100%.
- Las C), 1/3 parte del 100%.
- Y las D) un valor de 0.

Los cuestionarios ya contestados se pasaban por el lector óptico, obteniéndose un primer archivo que constaba de alrededor de 8000 registros. En éste se detectaban y eliminaban los cuestionarios con errores. Posteriormente este archivo era introducido a un programa de conteo y obtención de promedios.

A continuación se describe esquemáticamente la elaboración de este proceso, en el diagrama 2.

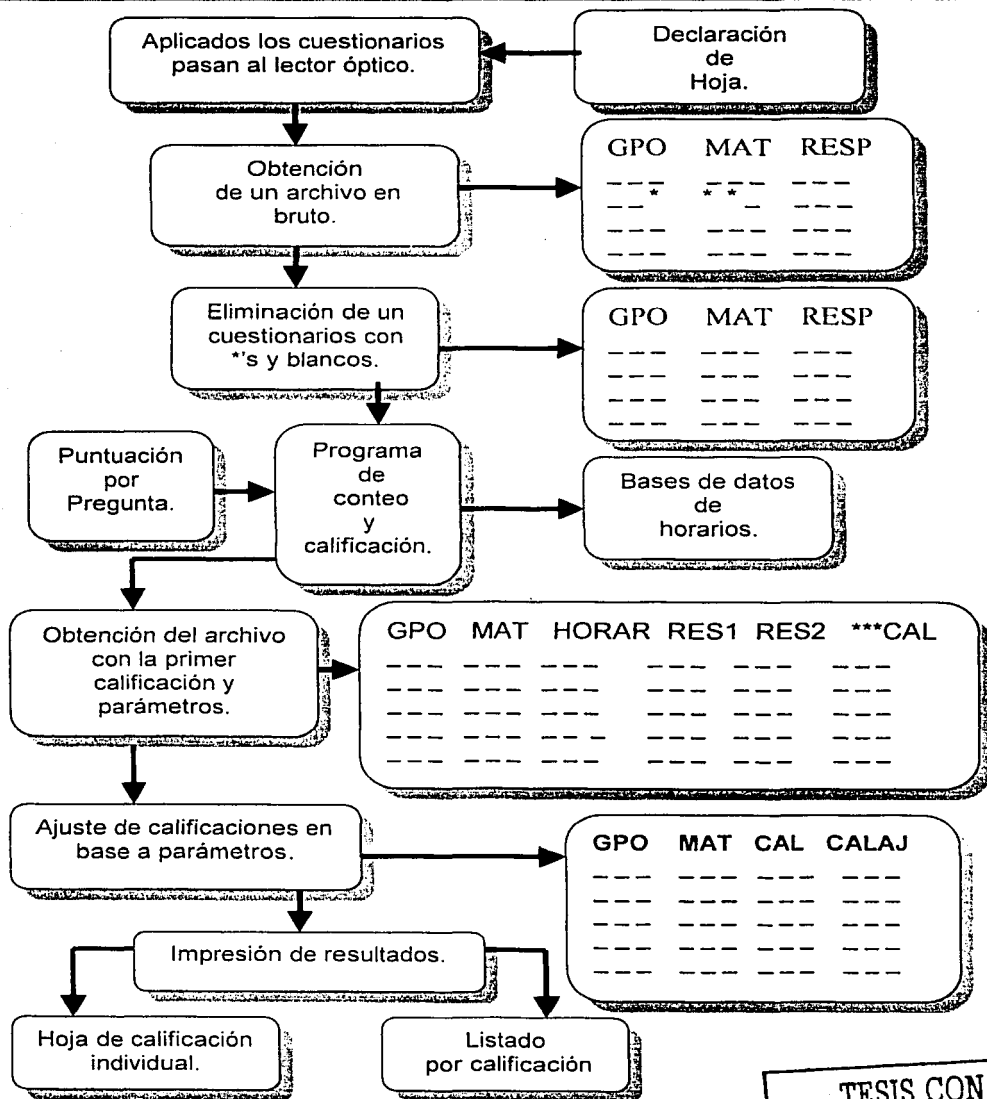


Diagrama 2. Proceso de Evaluación al Profesor

TESIS CON FALLA DE ORIGEN

Actualmente el sistema de evaluación al profesor por los alumnos se efectúa de manera automatizada, sustituyendo de esta forma los procesos manuales. La aplicación del cuestionario se realiza al inicio del siguiente semestre, como antecedente para que el alumno efectúe su inscripción; llenando el número de cuestionarios correspondientes a las materias, en las cuales estuvo inscrito el semestre anterior.

Una vez contestados los cuestionarios de evaluación por los alumnos y realizando los promedios correspondientes se generan las hojas de resultados, a continuación se muestra el formato 1 de la hoja de resultados.

**FACULTAD DE MEDICINA VETERINARIA Y ZOOTECNIA.**  
**UNIDAD DE PLANEACIÓN.**  
**EVALUACIÓN DE PROFESORES REALIZADA POR LOS ALUMNOS EN EL SEMESTRE.**

**NOMBRE DEL PROFESOR: XXXXXXXXXX.****MATERIA: XXXX XXXXX.****GRUPO: XXXXX.**

**CONOCIMIENTOS.** Evalúa los conocimientos que el profesor tiene de los temas que consta el curso.  
**COMUNICACIÓN.** Este concepto evalúa la capacidad del maestro para exponer con claridad su materia, si la presentación de su clase es organizada y lógica, si mantiene la atención e interés del alumno, si sus clases no son monótonas, si invitan al alumno a asistir y si especifica claramente sus objetivos.  
**EXPERIENCIA PROFESIONAL.** Evalúa la capacidad del profesor para permitir la relación del conocimiento de la materia con la vida diaria, ejemplos, experiencias u otro tipo de actividades.  
**MOTIVACIÓN.** De acuerdo con lo expuesto en los cuestionarios, los alumnos evalúan positivamente al profesor dinámico que trata de entusiasmar a sus alumnos. Trata en lo posible que el alumno estudie por su cuenta.  
**ACCESIBILIDAD.** Evalúa la capacidad del profesor de establecer en forma personal la orientación, comunicación y asesoría a los alumnos.  
**JUSTICIA E IMPARCIALIDAD.** Trato imparcial, abarca tanto evaluación como relación personal con el grupo. El profesor tiene un criterio de calificación el cual respeta.

AREAS.	CALIFICACIÓN OBTENIDA.	PROMEDIO DE LA FACULTAD.	MÁXIMO QUE PUEDE OBTENER.
COMUNICACIÓN.	XXXX	XXXX	XXXX
CONOCIMIENTOS.	XXXX	XXXX	XXXX
EXPERIENCIA PROFESIONAL.	XXXX	XXXX	XXXX
PUNTUALIDAD Y ASISTENCIA.	XXXX	XXXX	XXXX
MOTIVACIÓN.	XXXX	XXXX	XXXX
ACCESIBILIDAD.	XXXX	XXXX	XXXX
JUSTICIA E IMPARCIALIDAD.	XXXX	XXXX	XXXX
TOTAL.	XXXX	XXXX	XXXX

**AUTOEVALUACIÓN DE LOS ALUMNOS (PORCENTAJES)**

	MÁS DEL 90%	ENTRE 75% Y 90%	ENTRE 50% Y 75%	MENOS DEL 50%
La frecuencia con que asistí a clases fue...	XXXX	XXXX	XXXX	XXXX
Yo cumplí con las tareas o lecturas asignadas por el profesor.....	SIEMPRE XXXX	FRECUENTEMENTE XXXX	POCAS VECES XXXX	NUNCA XXXX
Yo recomendaría este profesor a mis compañeros.....	SI XXXX	NO XXXX		
Considerando el esfuerzo que realicé durante el curso, la calificación que obtuve merecía.....	JUSTA XXXX	INJUSTA XXXX		

**NÚMERO DE ALUMNOS QUE EVALUARON: XX**

Formato 1.

**TESIS CON  
FALLA DE ORIGEN**

### **I.III ALUMNO.**

El alumno es el principal constructor del aprendizaje, participando activamente dentro del aula en colaboración con el profesor y, resulta ser el principal afectado o beneficiado de las actividades del docente, cuya finalidad es conducir al aprendizaje.

Para establecer su participación fueron seleccionados 3 grupos de un total de 19 grupos de la Facultad de Medicina Veterinaria y Zootecnia.

Estos 3 grupos fueron divididos en 3 estratos formados de la siguiente manera.

1. El primer estrato formado por alumnos de primer ingreso.
2. El segundo por alumnos de segundo y tercer año.
3. El tercero por alumnos de cuarto y quinto año.

Posteriormente se les aplicó una encuesta para que determinaran las características que debe tener un buen docente.

Contribuyeron a la validación de un cuestionario preliminar, basado en la determinación de las características anteriormente especificadas, y finalmente a la corrección adecuada, que permitió la realización de un cuestionario de evaluación para un docente de licenciatura.

**I.IV PARÁMETROS A EVALUAR.**

Como resultado del proceso realizado para diseñar el cuestionario, que permitiría evaluar a los docentes de la Facultad de Medicina Veterinaria y Zootecnia, se determinaron las características mencionadas con mayor frecuencia.

CARACTERÍSTICAS MECIONADAS POR EL 40% O MÁS DE LOS CUESTIONARIOS	PESO PROMEDIO (AJUSTADO AL 100%) POR CARACTERÍSTICA.
COMUNICACIÓN.	XXXX
CONOCIMIENTOS.	XXXX
EXPERIENCIA PROFESIONAL.	XXXX
PUNTUALIDAD Y ASISTENCIA.	XXXX
MOTIVACIÓN.	XXXX
ACCESIBILIDAD.	XXXX
JUSTICIA E IMPARCIALIDAD.	XXXX

Por lo tanto los factores que influyen en la calificación y ajuste de evaluación al profesor, son los siguientes.

Habilidad compuesta por los siguientes parámetros.

- 1) **Conocimientos.** Evalúa los conocimientos que el profesor tiene de los temas de que consta el curso.
- 2) **Comunicación.** Permite saber la capacidad del maestro para exponer con claridad su materia, la presentación de sus clases es organizada y lógica, mantiene la atención e interés del alumno, sus clases no son monótonas, invitan al alumno a asistir y específica claramente sus objetivos.
- 3) **Experiencia Profesional.** Evalúa la capacidad del profesor para permitir la relación del conocimiento de la materia con la vida diaria, ejemplos, experiencias u otro tipo de actividades.
- 4) **Motivación.** Esta área permite saber la facilidad que tiene el docente para dirigir y mantener el interés en su clase, así como transmitir ideas que permitan una mejor asimilación de la materia.

- 5) **Justicia e Imparcialidad.** Trato imparcial, abarca tanto evaluación como relación personal con el grupo. El profesor tiene un criterio de calificación el cual respeta.
- 6) **Accesibilidad del profesor.** Evalúa la capacidad del profesor de establecer en forma personal la orientación, comunicación y asesoría a los alumnos.
- 7) **Puntualidad y Asistencia.** Evalúa si el profesor asiste puntualmente.

Oportunidad se refiere a las condiciones como:

- 1) **Tamaño del grupo.**
- 2) **Las horas frente al grupo.**
- 3) **Semestre en que está situada la materia.**
- 4) **Turno en que se imparte.**

# **CAPÍTULO II**

## **INGENIERÍA DEL SOFTWARE.**

---

### **II.1 INGENIERÍA DEL SOFTWARE.**

La ingeniería del software es la disciplina, técnica y de gestión que se ocupa de la invención, producción y mantenimiento sistemático de productos de software de alta calidad, desarrollados a tiempo y a un mínimo costo.

La ingeniería del software contempla tres elementos fundamentales:

- Métodos
- Herramientas
- Procedimientos

Los métodos de la ingeniería del software nos dicen cómo hacer la construcción del software, incluyen una amplia gama de tareas como: planeación y estimación del proyecto, análisis de los requerimientos en cuanto al sistema y el software, diseño de estructuras de datos, algoritmos, procedimientos, codificación, prueba y mantenimiento.

Las herramientas se integran de tal forma que la información creada por una herramienta puede ser usada por otra, se establece un sistema para soporte de desarrollo de software llamado CASE (Ingeniería de Software Asistida por Computadora).

CASE combina software, hardware y una base de datos de ingeniería de software (depósito que contiene información importante sobre el análisis, diseño y construcción de programas y pruebas).

Los procedimientos son el vínculo que se mantiene entre métodos y herramientas, definen la secuencia en que se aplican los métodos técnicos, salidas requeridas (modelos, documentos, datos, informes, formularios, etc.), controles que nos ayudarán a tener una buena calidad y coordinación de cambios.

## II.II PARADIGMAS DE LA INGENIERÍA DEL SOFTWARE.

Para el desarrollo de todo sistema es necesario establecer un plan de trabajo, que nos permita incorporar una estrategia de desarrollo, por ello es necesario plantear una metodología o paradigma a seguir.

Se elige un paradigma basándose en la naturaleza del proyecto y la aplicación, los métodos y herramientas a usar, y los controles y salidas que se requieren. Los diferentes modelos de procesos para la ingeniería del software son cuatro:

- Ciclo de vida clásico.
- Prototipo.
- Modelo de espiral.
- Técnicas de cuarta generación.

### II.III.I CICLO DE VIDA CLÁSICO.

Ciclo de vida clásico llamado también modelo de cascada, permite un enfoque sistemático y secuencial del desarrollo de software, que comienza en el nivel de planeación del sistema y avanza con el análisis, diseño, codificación, prueba y mantenimiento. Cada etapa se explica y representa mediante la figura 2.1.

- **Planeación del sistema.** Es la etapa en la que se determina si el proyecto es o no factible de realizar y se determinan tiempos y costos aproximados, estableciendo así la ruta crítica de cada actividad. Esto es porque la falta de planeación de un sistema es la causa principal de retrasos en programación, incremento de costos, poca calidad, y altos costos de mantenimiento en los desarrollos de productos de software.
- **Análisis.** Se analizan los requerimientos del cliente y se centra especialmente en el software. Para comprender la naturaleza del programa a construir, el ingeniero de software debe tener claro el dominio de la información de software, al igual que el rendimiento y la interfaz.
- **Diseño.** El diseño de software se enfoca en cuatro atributos, como son estructura de datos, arquitectura de software, representaciones de interfaz y detalle de procedimientos.
- **Generación de Código.** El diseño se debe traducir en código, si se lleva a cabo el diseño en una forma detallada, la generación de código se realizará mecánicamente.

- **Pruebas.** Una vez generado el código se realizan pruebas del programa, el proceso se basa en los procesos lógicos internos del software.
- **Mantenimiento.** El software debe de adaptarse para acoplarse a los cambios de su entorno externo. Esta etapa nuevamente aplica cada uno de los pasos del ciclo de vida al programa existente en lugar de tener que generarse un nuevo sistema.

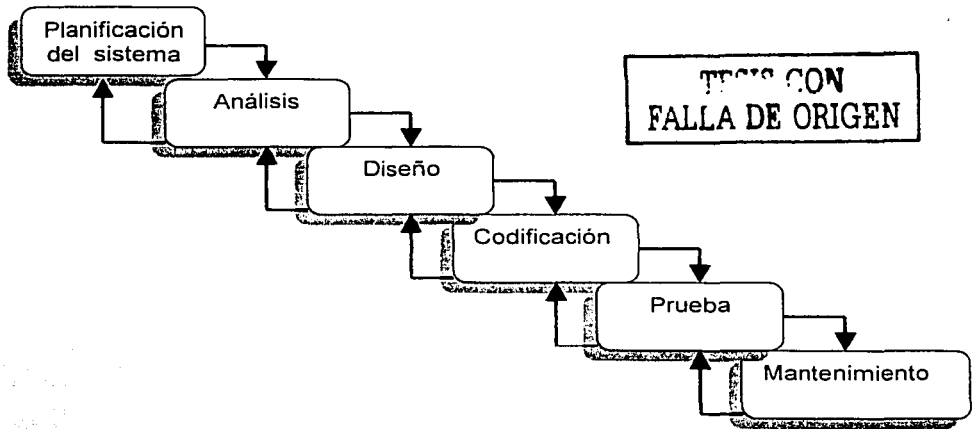


Figura 2.1. Ciclo de vida clásico.

### II.II.II. PROTOTIPO.

Este paradigma comienza con la colección de requerimientos, el cliente y el desarrollador definen los objetivos globales para el software, identifican las necesidades y enmarcan las áreas del esquema, pero no analizan detalladamente los requerimientos de entrada, procesamiento o salida de datos, estos factores contribuyen a generar un diseño rápido, este diseño rápido lleva a la construcción de un prototipo que es evaluado por el cliente y usado para refinar los requerimientos de software, finalmente se tiene un prototipo que se va perfeccionando con base a las necesidades del cliente. A continuación se muestra en la figura 2.2.

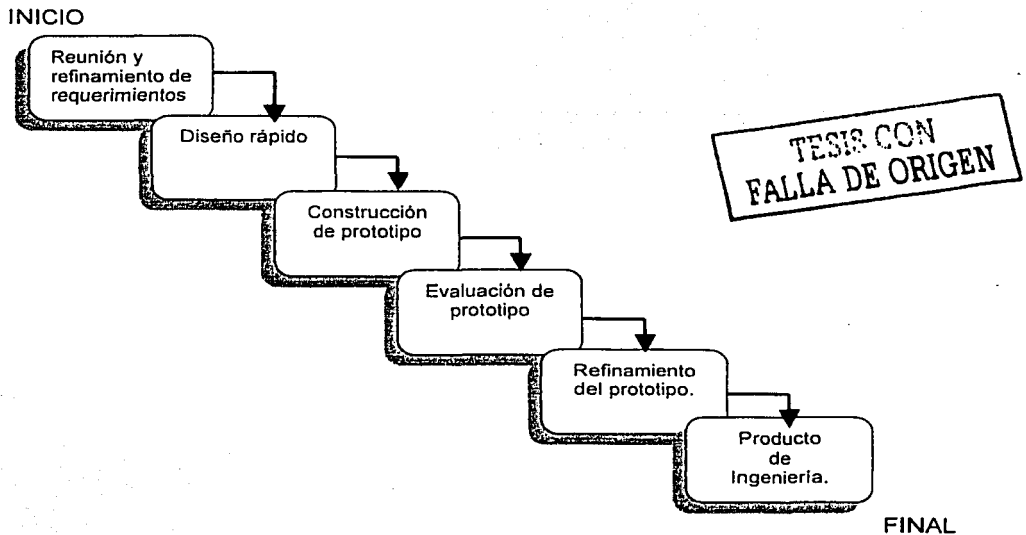


Figura 2.2 Prototipo.

Hay dos modos de usar los prototipos:

- En las fases de requisitos o de diseño de un modelo en cascada **Prototipado rápido, o desechable**.
- En un modelo de **prototipado evolutivo**, también llamado desarrollo incremental con mejora iterativa, o programación exploratoria.

**Prototipado rápido.**

Para que sea efectivo:

- Debe ser un sistema con el que se pueda experimentar.
- Debe ser comparativamente barato (< 10%).
- Debe desarrollarse rápidamente.
- Hacer énfasis en la interfaz de usuario.
- Utilizar equipo de desarrollo reducido.
- Emplear herramientas y lenguajes adecuados.
- El prototipado es un medio excelente para recoger el 'feedback' (realimentación) del usuario final.

### Modelo de Prototipado Evolutivo.

- Permite la construcción de una implementación parcial que cubre los requisitos conocidos, para ir aprendiendo el resto y, paulatinamente, incorporarlos al sistema.
- Reduce el riesgo y aumenta la probabilidad de éxito.
- No se conocen niveles apropiados de calidad y documentación.
- Se evitan problemas de gestión de configuración.

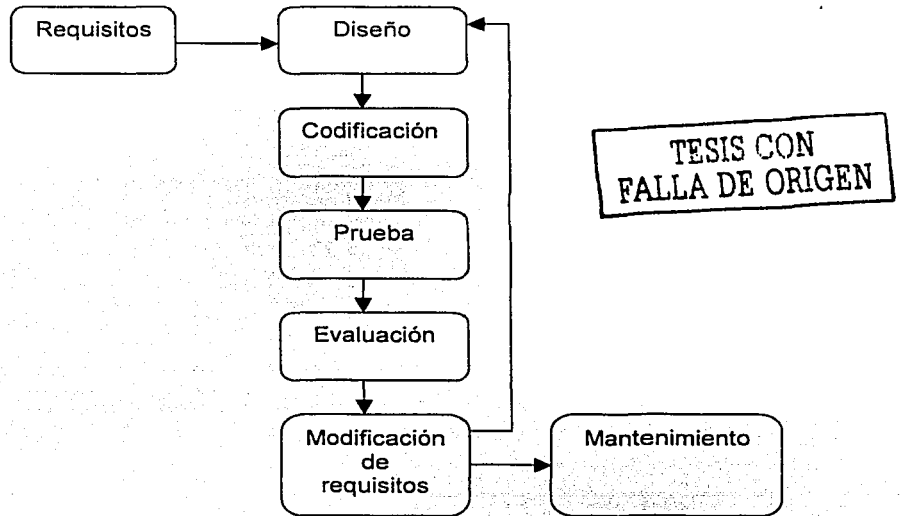


Figura 2.3. Modelo de Prototipado Evolutivo.

### II.II.III MODELO ESPIRAL.

El modelo en espiral propuesto por Boehn es un modelo de proceso de software evolutivo, que ha sido desarrollado para cubrir las mejores características, tanto el ciclo de vida clásico como la creación de prototipos, añadiéndole un nuevo elemento, el análisis de riesgo.

El modelo en espiral se divide en actividades estructurales o regiones de tareas que están representadas a continuación en la figura 2.4.

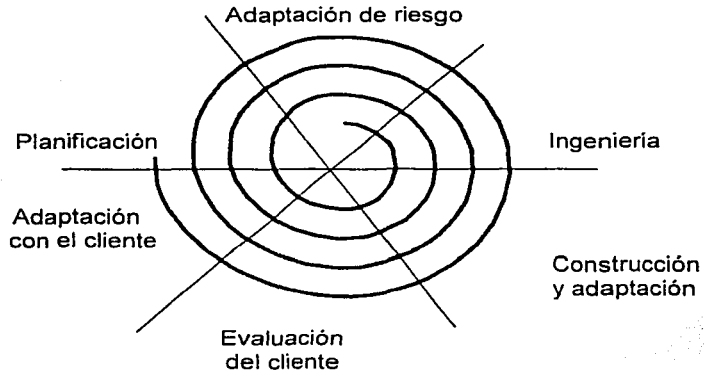


Figura 2.4 Modelo espiral.

Existe una comunicación con el cliente que se establece entre el desarrollador y el cliente, se planifican las tareas requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto, se evalúan riesgos técnicos y de gestión, en la ingeniería se requieren de las tareas para construir una o más de las representaciones de la aplicación, en la construcción y adaptación se debe de probar e instalar el software y finalmente se tiene la evaluación del cliente tomando como base los resultados.

TESIS CON  
FALLA DE ORIGEN

### II.II.IV TÉCNICAS DE CUARTA GENERACIÓN (4GT).

Permiten al desarrollador especificar algunas características de software a un alto nivel. La herramienta genera automáticamente el código fuente basándose en la especificación del analista. El paradigma 4GT (Técnicas de cuarta generación) se orienta a la especificación de software usando formas de lenguaje especializado o notaciones gráficas que describan el problema a resolver en términos que el cliente los entienda.

Actualmente el paradigma 4GT puede incluir las siguientes herramientas:

- Consulta de bases de datos.
- Generación de informes.
- Manejo de datos.
- Iteración y definición de pantallas.
- Generación de códigos y capacidades gráficas de alto nivel.
- Capacidad de hoja de cálculo.

El modelo 4GT se describe de la siguiente manera en la figura 2.5.

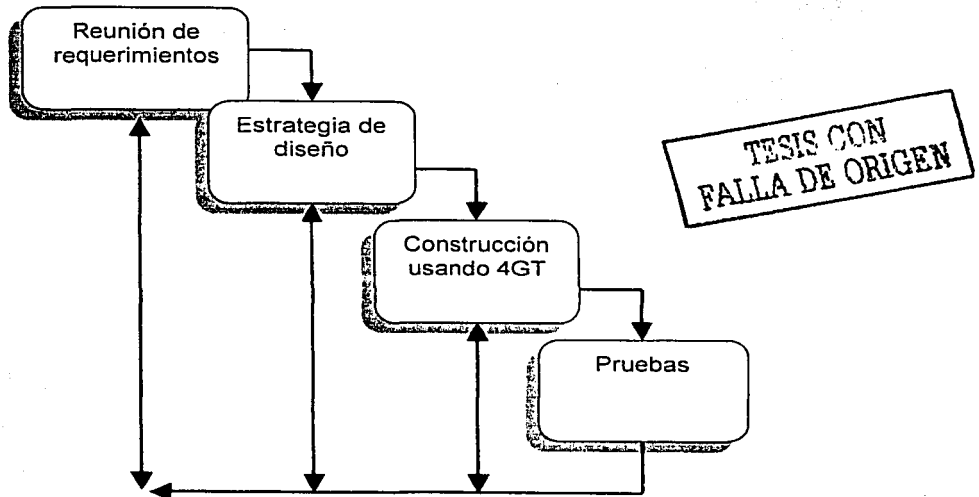


Figura 2.5 Técnicas de Cuarta Generación (4GT).

Este paradigma comienza con la reunión de requisitos donde el cliente es el que los establece, para aplicaciones pequeñas se puede usar un lenguaje de procedimientos (4GT), para aplicaciones más grandes, es necesario desarrollar una estrategia de diseño para el sistema.

La construcción usando 4GT permite al Ingeniero representar los resultados deseados de tal forma que se genere el código automáticamente y produzca buenos resultados.

### **II.II.V METODOLOGÍA UTILIZADA.**

Después del análisis efectuado sobre las diferentes metodologías presentadas anteriormente, se determinó utilizar el modelo en cascada para el desarrollo del Sistema de Control de Evaluación de Docencia y Desempeño Académico por Internet, debido a que dicha metodología presenta un proceso con progresos ordenados y coherentes.

En el ciclo de vida clásico del desarrollo de sistemas se expone:

- Un paradigma que demande un enfoque sistemático y secuencial del desarrollo de software que comienza con el nivel de ingeniería de sistemas y avanza hacia el análisis, diseño, codificación, prueba y mantenimiento

En la ingeniería de sistemas se establecen los requerimientos para los elementos del sistema.

En el diseño de un sistema de información se producen los detalles que establecen la forma en que el sistema cumplirá con los requerimientos identificados en la fase de análisis.

Con el desarrollo del software se obtendrá la codificación, programación y desarrollo de los diferentes módulos.

Las pruebas del sistema deberán realizarse para efectuar las correcciones necesarias antes de llegar a una prueba final.

Finalmente en la implementación y evaluación del sistema se deberán presentar los resultados esperados o bien tratar de cubrir las expectativas que se plantearon.

# CAPÍTULO III

## ANÁLISIS DEL SISTEMA.

---

### III.I REQUERIMIENTOS DE INFORMACIÓN.

#### INTERNET.

Internet se puede entender como una tecnología y como una forma de comunicación comparable con el sistema telefónico.

Ambos sistemas se basan en un determinado hardware que es el que utilizan, y ambos tienen miles de cables y otros componentes que sirven para unirlos con otros lugares.

Igual que dos personas necesitan hablar el mismo idioma para poder tener una conversación, todas las computadoras que se encuentran conectadas a una red necesitan tener un lenguaje común con el fin de poder comunicarse.

Para que sea posible el intercambio de información entre múltiples redes, es necesario que todas las computadoras y las redes que las conectan utilicen las mismas reglas para comunicarse unas con otras. Al lenguaje capaz de hacer que las computadoras puedan interactuar las unas con las otras se denomina *protocolo*.

El protocolo es un conjunto de procedimientos para interactuar que se encuentran perfectamente definidos y que deben utilizar todas las computadoras.

Existen dos protocolos básicos que hacen posible la comunicación y el intercambio de datos entre computadoras dentro de Internet.

- Protocolo de Control de Transmisión (TCP).
- Protocolo de Internet (IP).

Debido a que estos dos protocolos son básicos para el funcionamiento de Internet, se hace referencia a ellos como TCP/IP.

Incluso aunque las diferentes redes que forman Internet trabajan de manera diferente, TCP/IP proporciona el lenguaje necesario para poder llevar a cabo la comunicación y el intercambio de información entre ellas.

#### REDES DE COMPUTADORAS.

Una red de computadoras es un sistema de computadoras, dispositivos de red, impresoras y aplicaciones de software interconectados mediante comunicaciones por cable, fibra o radio.

- Una red dentro de un área limitada, como un edificio o una universidad es una red de área local o LAN (Local Area Network).

- Aquella que abarca una gran zona geográfica se llama red de área amplia o WAN (Wide Area Network).
- Una red de área metropolitana MAN conecta varias redes LAN dentro de una gran ciudad o región metropolitana.

### **PROTOCOLO TCP/IP.**

TCP/IP toma la información destinada a Internet y la divide en pequeños bloques (paquetes) haciendo posible la transmisión electrónica de los datos a través de las conexiones eléctricas. Los paquetes son dirigidos de la computadora fuente a la computadora destino, donde se reconstruye toda la información una vez que todos los paquetes en la transmisión; en este caso, TCP/IP pide a la computadora fuente vuelva a enviar el paquete receptor. Por lo tanto podemos concluir que los protocolos TCP/IP son responsables de las siguientes tareas.

- TCP es el protocolo encargado de dividir la información en paquetes, que serán enviados de una computadora a otra, y de volverlos a unir cuando éstos sean recibidos en la computadora destino.
- IP es el protocolo encargado de transmitir la información y de encontrar el camino óptimo para enviarlos de una computadora a otra.

### **MODELO OSI.**

Al principio del desarrollo de la informática cada fabricante establecía los procedimientos de comunicación entre sus computadoras de forma independiente, siendo muy difícil, la comunicación entre computadoras de fabricantes distintos.

Poco a poco se fue haciendo necesario disponer de unas normas comunes que permitiesen la intercomunicación entre computadoras.

De todos los protocolos propuestos destaca el modelo OSI (Interconexión de Sistemas Abiertos), que fue propuesto por la Organización Internacional de Normalización (ISO).

El modelo OSI, trata de establecer las bases para la definición de protocolos de comunicación entre sistemas informáticos.

Propone dividir en niveles todas las tareas que se llevan a cabo en una comunicación entre computadoras.

**Nivel 1 (Física).**

- Se encarga de transmitir los datos por el medio de transmisión (por ejemplo cable).

**Nivel 2(Enlace).**

- Permite que los datos se codifiquen como señales eléctricas en el nodo de transmisión , se decodifiquen en el nodo de recepción y se verifiquen para comprobar los errores.

**Nivel 3 (Red).**

- Encargada de controlar el paso de paquetes (unidad de datos) por la red.

**Nivel 4 (Trasporte).**

- Garantiza que los datos se envían de manera fiable desde el nodo de transmisión hacia el nodo destino.

**Nivel 5 (Sesión).**

- Es responsable de la continuidad de la conexión o sesión entre dos nodos.
- Establece la conexión y asegura que éste se mantiene mientras dure la sesión de comunicación.

**Nivel 6 (Presentación).**

- Este nivel se encarga de formatear los datos. Cada tipo de red utiliza un esquema de formato particular.
- Una forma de ver la capa de presentación es como un verificador de sintaxis.

**Nivel 7 (Aplicación).**

- Representa los accesos a las aplicaciones y a los servicios de red de los usuarios de ordenador, tal como las bases de datos.

## **ARQUITECTURA TCP/IP.**

### **(Modelo de Internet).**

La arquitectura TCP/IP distribuye el procesamiento entre los clientes y los servidores en la red.

Los clientes solicitan la información de los servidores mientras que los servidores almacenan los datos y los programas.

Este modelo se describe en cuatro capas estableciendo las correspondencias posibles con el modelo de referencia OSI.

### **Nivel 1 (Interfaz de Red).**

- Se encarga del intercambio de datos entre un host, la red y entre los dispositivos de la misma red.

Nota: host (computadora que tiene un sistema operativo que permite que varias computadoras accedan a él al mismo tiempo).

- Engloba las capas de Enlace de datos y Física del modelo OSI, aunque no incorpora las funciones de control de flujo y de errores propios de control de enlaces.

### **Nivel 2 (Red).**

- Corresponde con la capa de red del modelo OSI.
- Los protocolos de esta capa tienen como propósito el direccionamiento, el empacamiento y el encaminamiento.

### **Nivel 3 (Transporte).**

- Equivale al nivel del transporte del modelo OSI.

### **Nivel 4 (Aplicación).**

- Abarca las funciones de las 3 últimas capas del modelo de referencia OSI: sesión, presentación y aplicación.

La figura 3.1 ejemplifica las capas que componen el modelo de referencia OSI, así como los niveles utilizados en el modelo de Internet, también llamada arquitectura TCP/IP.

NIVEL 7	APLICACIÓN	APLICACIÓN.	NIVEL 4
NIVEL 6	PRESENTACIÓN		
NIVEL 5	SESIÓN	TRANSPORTE.	NIVEL 3
NIVEL 4	TRANSPORTE		
NIVEL 3	RED	INTERNET.	NIVEL 2
NIVEL 2	ENLACE	INTERFAZ DE RED	NIVEL 1
NIVEL 1	FÍSICA		

Figura 3.1 Modelo de referencia OSI y Modelo Internet.

### ARQUITECTURA CLIENTE/SERVIDOR.

El cliente es la entidad por medio de la cual un usuario solicita un servicio, realiza una petición o demanda el uso de recursos. Este elemento se encarga, básicamente, de la presentación de los datos o información al usuario en un ambiente gráfico.

El servidor es la entidad física que provee el servicio y devuelve resultados; ejecuta el procesamiento de datos, aplicaciones y manejo de información o recursos.

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

A continuación se presenta la siguiente figura 3.2 que ejemplifica la arquitectura Cliente/Servidor.

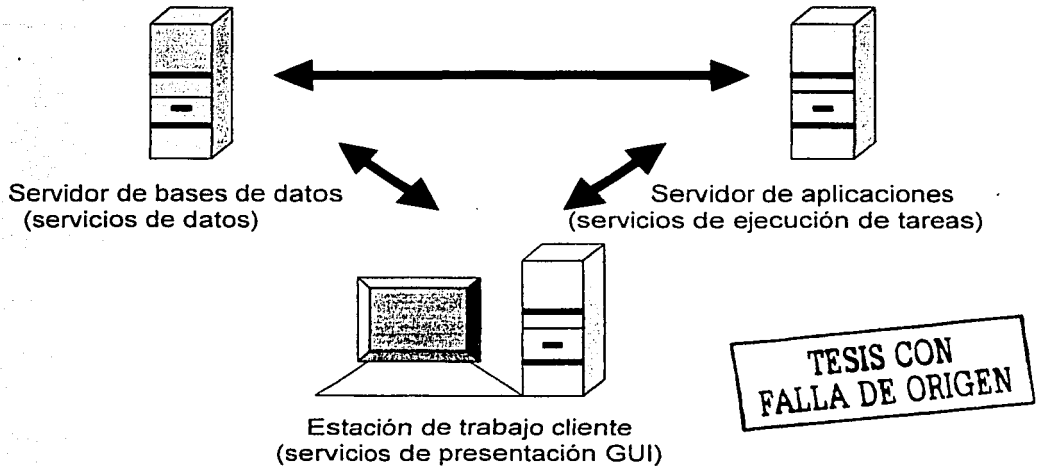


Figura 3.2 Sistema Cliente/Servidor.

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- Una computadora cliente es la computadora que accede a una computadora central, que permite el acceso a múltiples usuarios.
- Una computadora servidor tiene un sistema operativo que permite que varias computadoras accedan a él, al mismo tiempo.
- 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, del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

### **SERVIDORES WEB.**

Los Servidores Web son aquellos que permiten a los clientes compartir datos, documentos y multimedia en formato Web.

Aunque parte de la tecnología Cliente/Servidor, el servidor Web aporta las siguiente ventajas.

- El Web se crea normalmente como un sistema abierto al que cualquiera puede contribuir y acceder desde cualquier punto de la red de la empresa; no requieren de logins ni password como los sistema tradicionales Cliente/Servidor.
- Los Servidores Web vuelcan información con un simple clic del ratón a través de un proceso de hipervínculo.
- La información servida puede ser de cualquier tipo (datos, documentos, multimedia, etc).
- Dado que el Servidor Web es de tan fácil acceso, hace posible publicar información de forma instantánea.
- En una aplicación Cliente/Servidor el procesado de las tareas se realiza en el cliente, o bien en uno o más servidores, para lograr el mejor rendimiento y minimizar la carga en la red.

### **SISTEMAS OPERATIVOS.**

Un Sistema Operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas.

El objetivo principal de un Sistema Operativo es, entonces, lograr que el Sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

### **TIPOS DE SISTEMAS OPERATIVOS.**

- OS/2.
- UNIX.
- LINUX.
- WINDOWS 9X.

## LENGUAJES DE PROGRAMACIÓN.

La programación ejecutada desde el servidor web permite construir sitios web dinámicos e interactivos para nuestros clientes, incluyendo el pedido de productos on-line, la obtención de información de los clientes que les visitan o incluso la posibilidad de personalizar lo que sus clientes ven cuando visitan sus páginas.

A continuación se mencionan los lenguajes más utilizados recientemente que nos permiten desarrollar este tipo de aplicaciones.

- **PHP.** Lenguaje de programación preprocesador de hipertextos.
- **JAVA.** Es un lenguaje que permite realizar aplicaciones para que se ejecuten en el server, llamadas Servlets, o aplicaciones que se ejecutan en la máquina cliente, llamada applets.
- **ASP.** Proporciona la facilidad para conectarse con una Base de datos y extraer datos de la misma dinámicamente visualizándolos en el navegador.
- **JAVA SERVLETS.** Son programas que se ejecutan en un servidor Web y construyen páginas Web.
- **JAVA JSP (Páginas de Servidor Activas).** Es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente.

## BASE DE DATOS.

Cada vez más, los sitios web están incorporando aplicaciones dinámicas que necesitan una integración con bases de datos. Las bases de datos son los motores ocultos detrás de los sitios web que permiten llevar a cabo desde la exposición de información hasta las complicadas aplicaciones de comercio electrónico.

La base de datos a utilizar debe ser accesible desde Internet, de tal manera que permitan:

- Buscar, añadir, modificar o borrar registros de sus bases de datos sin que sea necesario conocimiento alguno de programación. Todo ello se puede hacer a través de un simple formulario.

- Actualizar las páginas web directamente a través de un simple formulario.
- Almacenar gran cantidad de información en un servidor.
- Proporcionar un rápido acceso a los datos en el momento de su actualización, consulta o petición.

### PROGRAMACIÓN ORIENTADA A OBJETOS.

La Programación Orientada a Objetos es un método de implementación en el cual los programas son organizados como grupos cooperativos de objetos, cada uno de los cuales representa una instancia de alguna clase, y estas clases, todas son miembros de una jerarquía de clases unida vía relaciones de herencia.

Los fundamentos de la programación orientada a objetos son los siguientes:

- **Objeto** es aquello que tiene un estado (propiedades más valores), comportamiento (acciones y reacciones a mensajes) e identidad (propiedad que lo distingue de los demás objetos).
- **Clase** es un conjunto de objetos que comparten una estructura y comportamiento en común.
- **Abstracción** es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros.
- **Encapsulación** es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales.
- **Modularidad** es la propiedad de un sistema que ha sido descompuesto en un conjunto de módulos coherentes e independientes.
- **Jerarquía o Herencia** es el orden de las abstracciones organizado por niveles.
- **Tipificación** es la definición precisa de un objeto de tal forma que objetos de diferentes tipos no puedan ser intercambiados, o cuando mucho, puedan intercambiarse de manera muy restringida.

- **Concurrencia** es la propiedad que distingue un objeto que está activo de uno que no lo está.
- **Persistencia** es la propiedad de un objeto a través del cual su existencia trasciende el tiempo (es decir, el objeto continua existiendo después de que su creador ha dejado de existir) o el espacio (es decir, la localización del objeto se mueve del espacio de dirección en que fue creado).

### **UML (LENGUAJE UNIFICADO DE MODELADO).**

El UML es una herramienta que permite a los desarrolladores de sistemas generar diseños que capturen sus ideas de una manera convencional y fácil de comprender para comunicarlas a otras personas.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, cuya finalidad es presentar diversas perspectivas de un sistema, llamado modelo.

Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Los principales diagramas del UML son los siguientes:

- **Diagramas de caso de uso.**
- **Diagramas de clases.**
- **Diagrama de secuencias.**

Cada uno de estos diagramas serán explicados con gran detalle, al momento de ser utilizados en capítulos posteriores.

### III.II BASES DE DATOS.

El procesamiento de información ha constituido una de las tareas básicas de cualquier civilización. Debido al crecimiento económico y demográfico, existen necesidades crecientes de administrar grandes cantidades de datos interrelacionados. Éstos datos interrelacionados considerados en un conjunto forman lo que se llama un sistema. La parte medular de cualquier sistema de información la constituyen sus datos almacenados.

Las bases de datos permiten definir sistemas de cómputo que almacenan grandes cantidades de información proporcionando métodos eficientes para la manipulación de los datos.

**Base de Datos:** Conjunto de datos relacionados entre sí que guardan una estructura en común y que se encuentran totalmente organizados.

**Sistema de Manejo de Base de Datos:** Un sistema de manejo de bases de datos consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos.

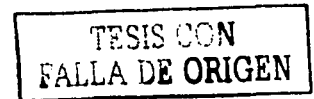
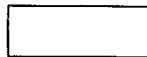
El objetivo primordial de un sistema de manejo de base de datos es crear un ambiente en que sea posible guardar y recuperar información de la base de datos en forma eficiente.

#### III.II.I MODELO ENTIDAD-RELACIÓN.

La estructura lógica general de una base de datos puede expresarse en forma gráfica por medio de un diagrama E-R que se integra de la siguiente forma:

**Entidad:** Una entidad es un objeto que existe y puede distinguirse de otros objetos. La distinción se logra asociando a cada entidad un conjunto de atributos que describen al objeto. Una entidad puede ser concreta, como por ejemplo, una persona o un libro, o abstracta, como un concepto.

Una entidad se representa con un rectángulo.



**Atributo:** Es cada una de las características o propiedades de una entidad. Una entidad puede tener muchos atributos como por ejemplo una persona puede tener como atributos el nombre, el apellido, dirección, teléfono, etc. Para cada atributo existe un rango de valores permitidos, llamado dominio del atributo.

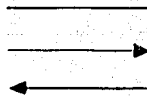
Se representa con una elipse.



**Relación :** Una relación es una asociación entre varias entidades. Un conjunto de relaciones es un grupo de relaciones del mismo tipo. Se representa con un rombo.



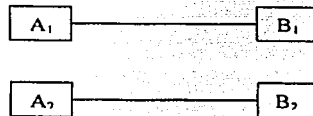
**Conexión:** Para conectar los atributos a los conjuntos de entidades, y los conjuntos de entidades a los conjuntos de relaciones se utilizan líneas



**Cardinalidad de mapeo:** Expresa el número de entidades con las que puede asociarse otra entidad mediante una relación.

a) Relación Uno a Uno (1:1):

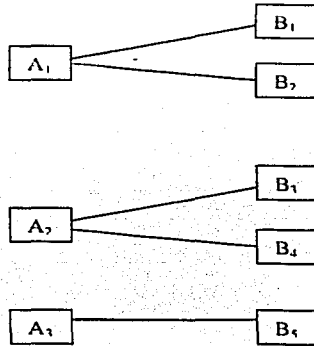
Una entidad en A está asociada únicamente con una entidad en B, y una entidad en B está asociada sólo con una entidad en A.



TESIS CON  
FALLA DE ORIGEN

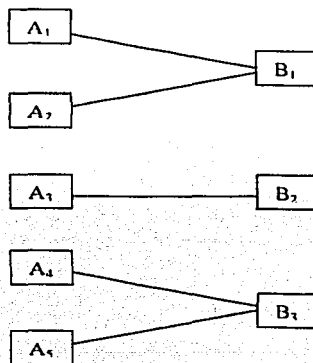
b) Relación Uno a Muchos(1:M) :

Una entidad en A está asociada con cualquier número de entidades en B, pero una entidad en B puede asociarse únicamente con una entidad en A.

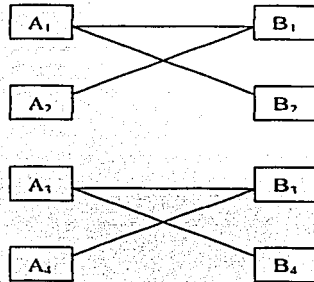


TESIS CON  
FALLA DE ORIGEN

c) Relación Muchos a Uno (M:1): Una entidad en A esta asociada únicamente con una entidad en B; pero una entidad en B está asociada con cualquier número de entidades en A.



d) Relación Muchos a Muchos (M:M): Una entidad en A está asociada con cualquier número de entidades en B, y una entidad en B está asociada con cualquier número de entidades en A.



- **Llave candidato.** Uno o más atributos que permiten identificar en forma única y mínima a una entidad.
- **Llave primaria:** Llave candidato seleccionada por el programador o diseñador.

TESIS CON  
FALLA DE ORIGEN

### **III.II.II NORMALIZACIÓN.**

El proceso de normalización parte de las formas normales definidas por E.F. Codd (1970) creador de las bases de datos relacionales. Codd formuló tres formas normales (1FN), (2FN) y (3FN).

Este proceso permite crear una estructura relacional apropiada para almacenar la información.

La normalización minimiza la redundancia en los datos, y permite agrupar los datos de forma sencilla.

#### **FORMAS NORMALES**

La teoría de la normalización está basada en el concepto de formas normales. Se dice que una relación está en una forma normal particular si satisface cierto conjunto específico de restricciones.

##### **Primera Forma Normal (1FN):**

Una relación está en primera forma normal (1FN) si los valores en la relación son atómicos para cada atributo en la relación. Esto quiere decir que los valores de los atributos no pueden ser un conjunto de valores o un grupo repetitivo.

##### **Segunda Forma Normal (2FN):**

Una relación se dice que está en 2FN si y sólo si cumplen con dos condiciones:

1. Se encuentra en 1FN.
2. Todo atributo secundario depende totalmente de la llave completa y no de una parte de ella.

##### **Tercera Forma Normal (3FN):**

Una relación está en 3FN si y sólo si cumple con dos condiciones:

1. Se encuentra en 2FN.
2. Un atributo secundario sólo se debe conocer a través de la llave principal de la relación y no por medio de otro atributo no primario.

### **III.II.III DIAGRAMA RELACIONAL DE BASE DE DATOS.**

El modelo relacional fue definido en 1970 por E.F. Codd quien propone un modelo de datos basado en la teoría de las relaciones, en donde los datos se estructuran lógicamente en forma de relaciones (TABLAS). El modelo relacional (MR) propone como principal objetivo aislar al usuario de las estructuras físicas de los datos, finalidad perseguida desde los inicios de las bases de datos.

#### **Objetivos del modelo relacional.**

- Independencia física.
- Independencia lógica.
- Flexibilidad.
- Uniformidad.
- Sencillez.

#### **Independencia física.**

Es decir que el modo en que se almacenan los datos no influye en su manipulación lógica y, por lo tanto, los usuarios que acceden a esos datos no tengan que modificar sus programas por cambios en el almacenamiento físico.

#### **Independencia lógica.**

Esto es, que el añadir, eliminar o modificar objetos de la base de datos no repercute en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos.

#### **Flexibilidad.**

En el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.

#### **Uniformidad.**

Las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos, por parte de los usuarios.

#### **Sencillez.**

Las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil.

Para cumplir los objetivos citados se introduce el concepto de relación (TABLA) como estructura básica del modelo.

Todos los datos de una base de datos se representan en forma de relaciones cuyo contenido varía en el tiempo. Formalmente una relación es un conjunto de filas en terminología relacional.

### Objetos del modelo relacional.

Existe una serie de objetos utilizados en el modelo relacional los cuales son:

**Relación (Entidad).** Conjunto de filas, pueden asociarse a lo que se conoce como tabla con ciertas propiedades.

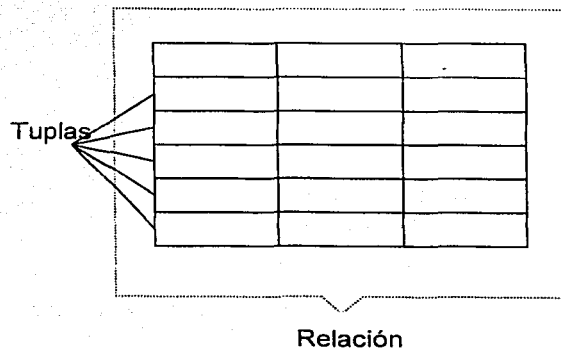
**Tupla.** Corresponde a una fila de esa tabla. Al número de tuplas se le denomina cardinalidad.

**Atributo.** Se refiere a una columna de esa tabla. La cantidad de atributos determina el grado de la relación.

**Dominio.** Es el conjunto de donde los atributos toman sus valores. Puede suceder que dos atributos distintos de una misma relación tomen sus valores del mismo dominio.

**Grado.** Es el número de atributos o columnas que posee una relación.

**Clave.** Definimos como clave de una relación a aquellos atributos que nos determinan de forma unívoca y mínima a una tupla de esa relación.



TESIS CON  
FALLA DE ORIGEN

TESIS CON FALLA  
DE  
ORIGEN

### III.II.IV. DICCIONARIO DE DATOS.

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

Un diccionario de datos debe ayudar al usuario de una base de datos a:

- Comunicarse con otros usuarios.
- Controlar los campos de datos de manera simple y efectiva, esto es, a introducir nuevos campos en los sistemas o a cambiar las descripciones de los campos.
- Reducir la redundancia e inconsistencia de los datos.
- Determinar el impacto de los cambios en los campos de datos sobre la base de datos total.

Un diccionario de datos está compuesto de las siguientes partes:

**Campo de datos:** Atributo que describe a una entidad. Cada campo tiene un nombre único o etiqueta. Éstos deben ser descriptivos y deben consistir de palabras claves o abreviaciones tomadas de listas autorizadas.

### III.III ANÁLISIS DEL SISTEMA.

El análisis del sistema es un estudio detallado de la forma como funciona el proceso operacional de la información en una organización.

Los principales propósitos de esta fase son aclarar los objetivos, necesidades y restricciones del sistema.

Para llevar a cabo nuestro análisis se realizó un estudio detallado del sistema de evaluación a profesores por los alumnos, actualmente utilizado en la Facultad de Medicina Veterinaria y Zootecnia, con la finalidad de detectar las deficiencias del sistema.

Posteriormente se efectuó un análisis de los requerimientos de información, especificando las herramientas a utilizar para el desarrollo del sistema de evaluación a profesores por los alumnos.

En el sistema de evaluación de profesores por los alumnos en la Facultad de Medicina Veterinaria y Zootecnia intervienen los siguientes departamentos.

**Departamento de Estudios Profesionales.** Procura la solución de los problemas y necesidades relacionadas con la administración escolar de los alumnos de licenciatura. Estudia cómo incrementar la eficiencia de los profesores de administración escolar. Planea, organiza, dirige y supervisa la revisión permanente del plan de estudios de la carrera y propone alternativas de actualización.

**Unidad de Planeación.** Ayuda al cumplimiento de los fines sustantivos de la Facultad y fortalece los vínculos con el país y sus problemas. Logra la participación efectiva de la comunidad en los programas, proyectos y acciones a través de los órganos y mecanismos institucionales. Colabora con la dirección de la Facultad en el desarrollo de planes y proyectos de la institución. Procesa y organiza los datos e información con el objeto de ser utilizado para la toma de decisiones por parte del Director de la Institución. Estudia el perfil del personal académico de la Facultad de Medicina Veterinaria y Zootecnia. Participa en la revisión y modificación de los programas de asignatura en el plan de estudios de la carrera.

**Secretaría General.** Coordina los asuntos académicos-administrativos que atañen al personal docente. Coordina el desarrollo de los trabajos de las secretarías y divisiones a su cargo, así como de las jefaturas del colegio. Manteniendo actualizadas, en coordinación con la secretaría administrativa, los nombramientos y demás movimientos referentes al personal académico mediante los mecanismos contemplados en la legislación universitaria. Coordina las actividades de las divisiones de estudios profesionales de posgrado, del sistema de universidad abierta, la unidad de planeación, la secretaría de producción animal, y de la coordinación de educación continua.

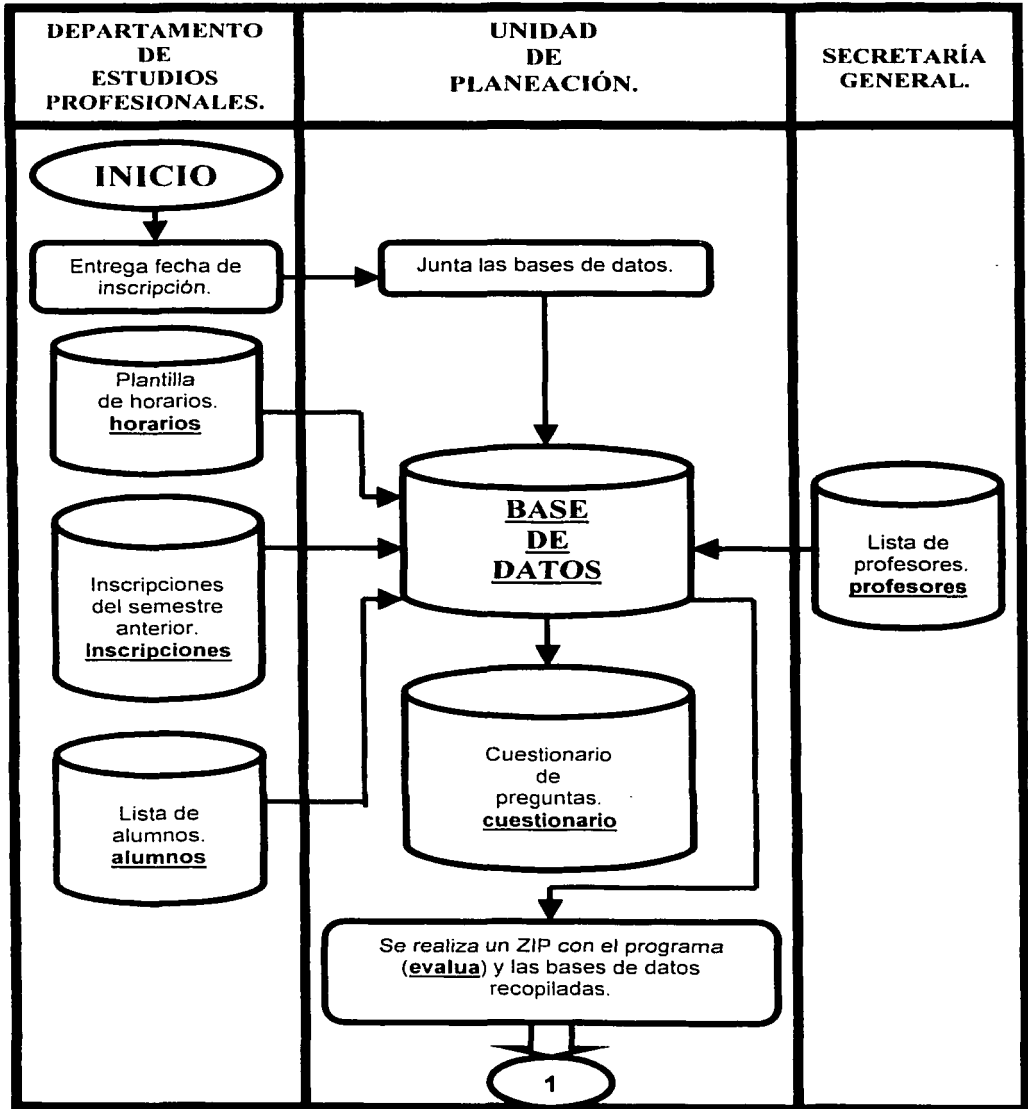
El procedimiento actual para la evaluación de profesores por los alumnos en la Facultad de Medicina Veterinaria y Zootecnia se describe a continuación.

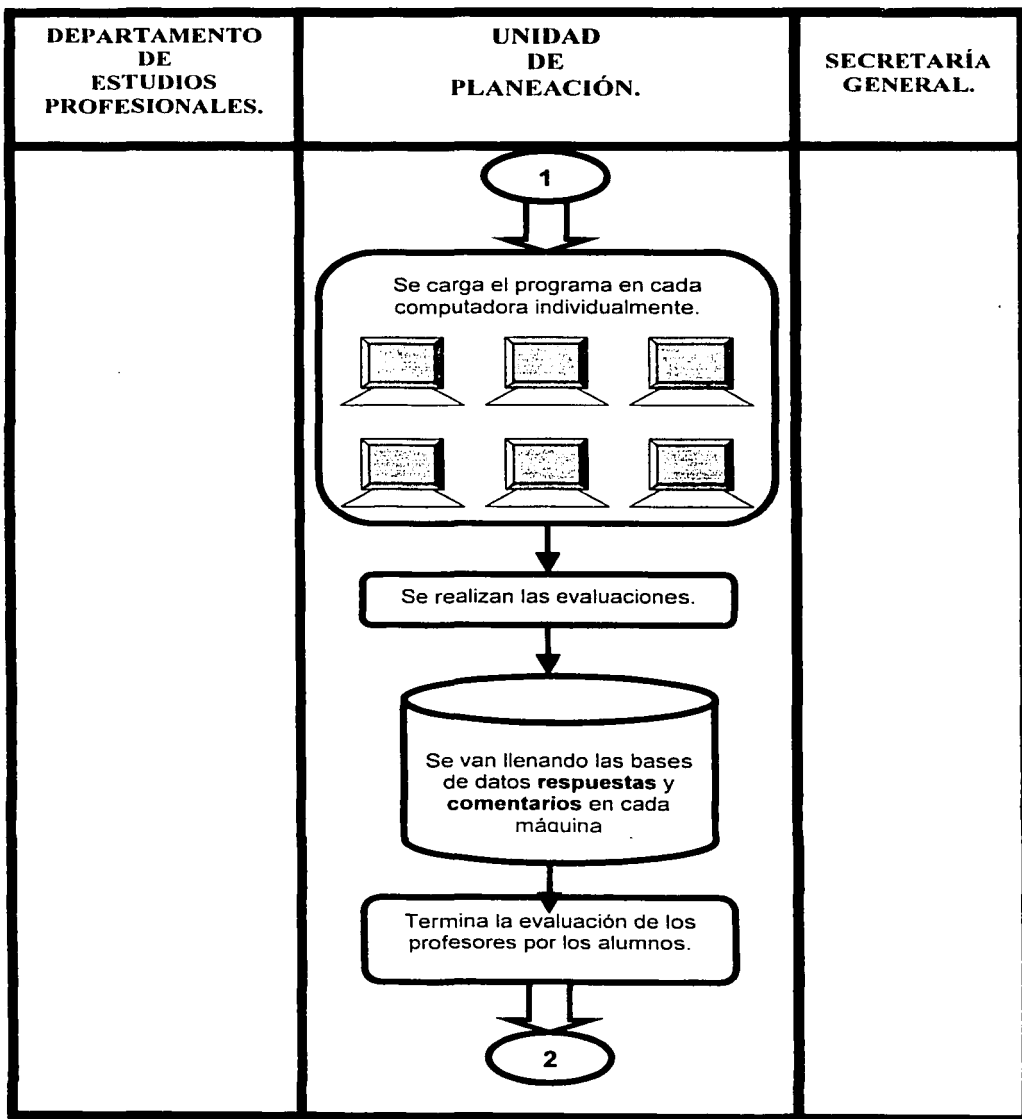
1. El Departamento de Estudios Profesionales entrega la fecha de inscripción al Departamento de Planeación, debido a que las evaluaciones a los profesores se decidió realizarlas al inicio del siguiente semestre, cuya justificación fue que en ese período, es donde se tiene la mayor concentración de alumnos.
2. El Departamento de Planeación se encarga de conseguir las siguientes bases de datos.
  - Plantilla de horarios del semestre, contenida en el archivo **horarios**, proporcionada por el Departamento de Estudios Profesionales.
  - La Secretaria General proporciona la base de datos que contiene la lista de profesores, dentro del archivo **profesores**.
  - La base de datos que contiene los cuestionarios se encuentra en el archivo **cuestionario**, obtenida del Departamento de Planeación.
  - Las inscripciones del semestre que terminó se encuentran en la base de datos **Inscripción**, otorgada por el Departamento de Estudios Profesionales.
  - Finalmente la ultima base de datos donde se encuentra el registro de alumnos, se obtiene del Departamento de Estudios Profesionales en el archivo **alumnos**.
3. Se juntan todas las bases de datos antes mencionadas, con el programa ejecutable de evaluación de profesores y se realiza un zip (compresión de archivos) generando un archivo **evalua**.
4. Se carga el archivo evalua.exe en cada una de las computadoras individualmente.
5. Se realizan las evaluaciones por el alumno en cada computadora.
6. Se van llenando las bases de datos **respuestas** y **comentarios**, las cuales contienen las respuestas y comentarios efectuados por los alumnos en el cuestionario aplicado.
7. Termina la evaluación de profesores por los alumnos.
8. El Departamento de Planeación, recolecta de cada máquina las bases de datos de los resultados y los junta en una sola.

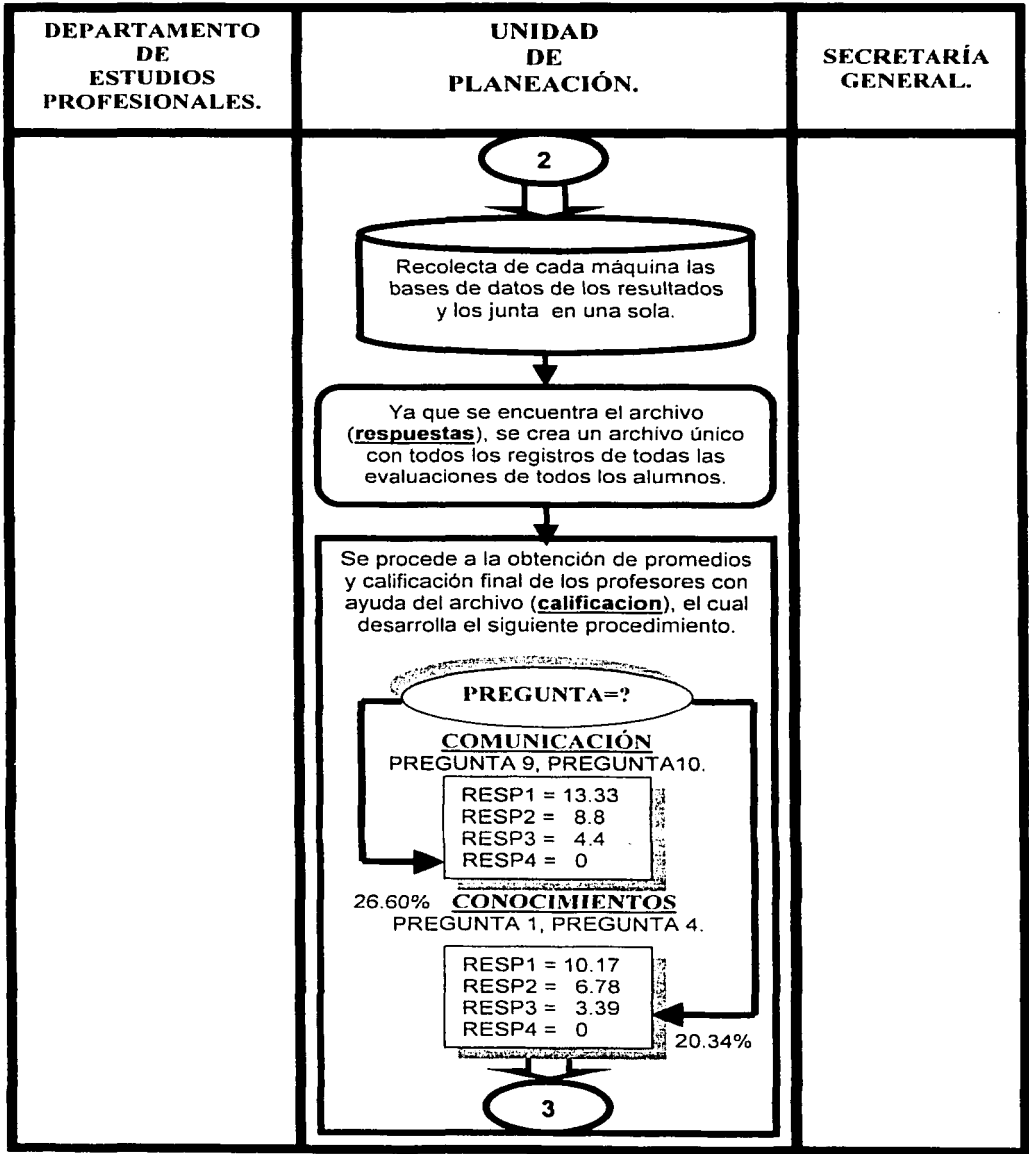
9. Ya que se crea el archivo **respuestas** que contiene los resultados de la evaluación, se crea un archivo único con todos los registros de todas las evaluaciones de todos los alumnos, aproximadamente se tienen 165,000 registros.
10. Se procede a la obtención de promedios y calificaciones de los profesores utilizando el archivo (**calificación**), el cual analiza los 165,000 registros.
11. Se genera el archivo de base de datos (**calificaciontotal**) con todos los resultados de todas las preguntas, obteniendo aproximadamente 350 registros.
12. Se realiza un ajuste de la calificación final, utilizando un modelo estadístico.
13. Se imprime una hoja personalizada que se entrega al profesor, la cual contiene: nombre del profesor, materia, grupo, calificación obtenida en cada una de las áreas, autoevaluación del alumno y calificación final ya ajustada.
14. Finalmente se obtiene un listado con los resultados obtenidos de la evaluación a los profesores de la Facultad de Medicina Veterinaria y Zootecnia, las cuales son entregadas al Departamento de Estudios Profesionales y a la Secretaría General.

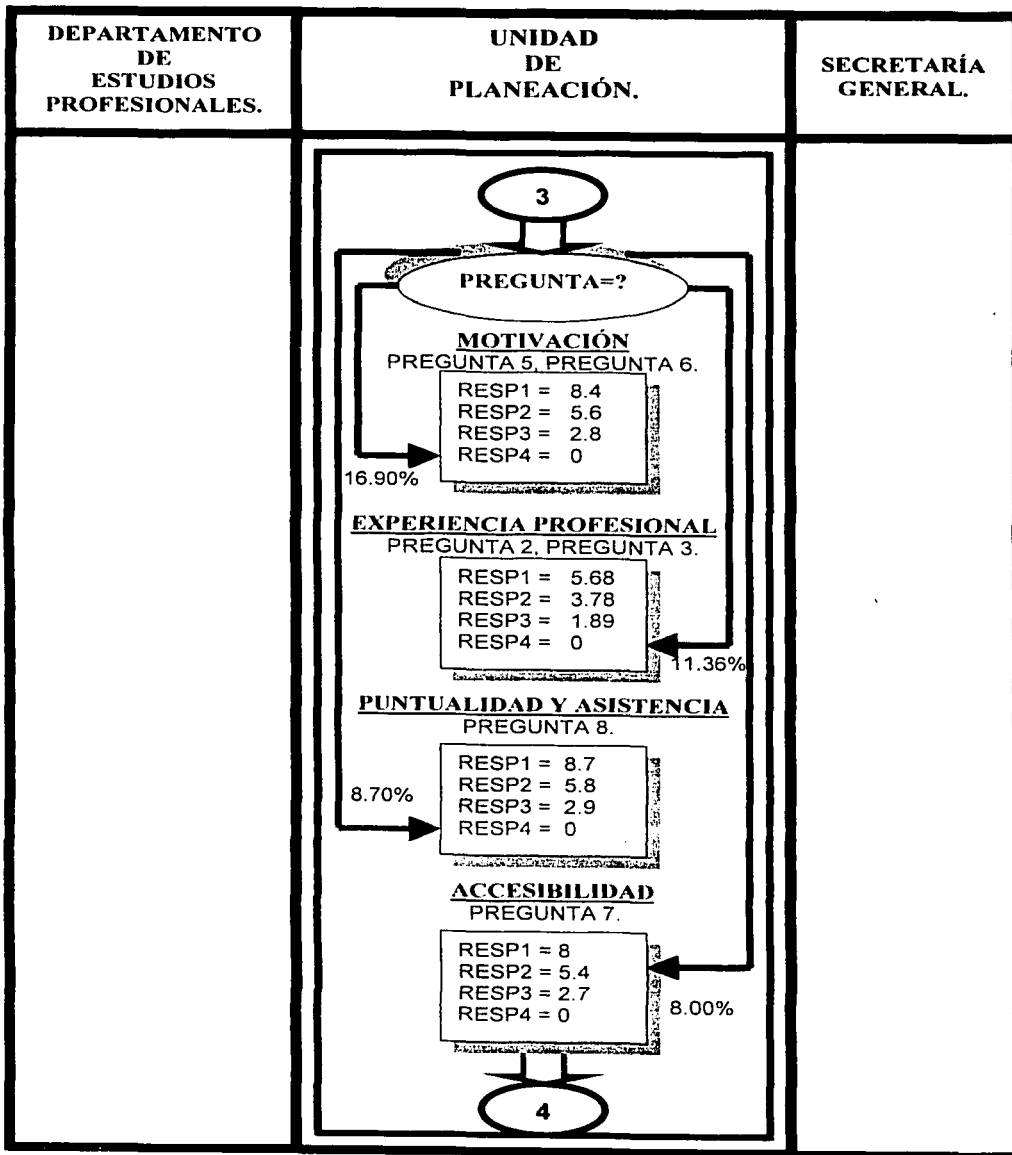
A continuación se describe mediante el diagrama de la figura 3.3, el proceso anteriormente mencionado.

FIGURA 3.3 DIAGRAMA DE FLUJO DE INFORMACIÓN DE LA SITUACIÓN ACTUAL.

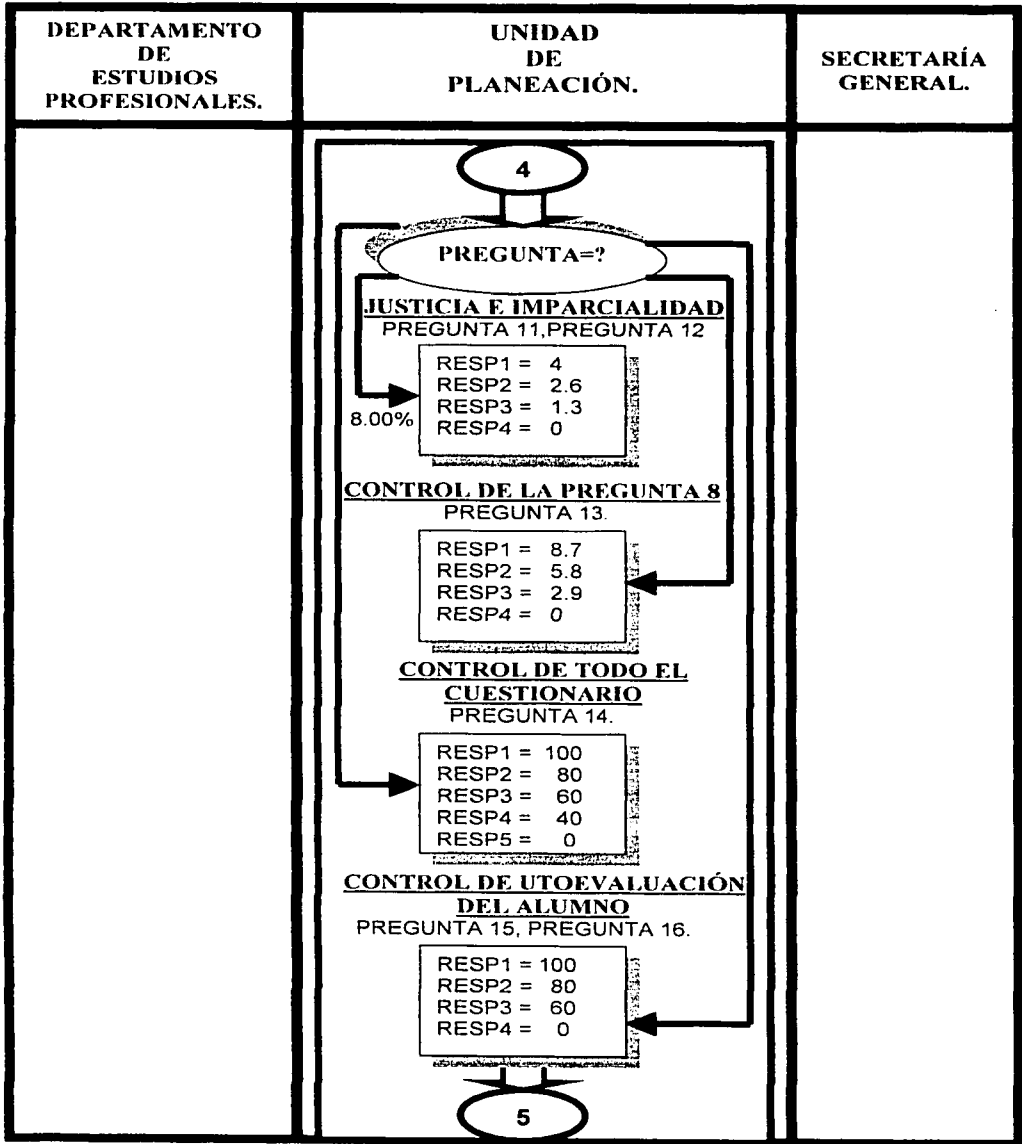




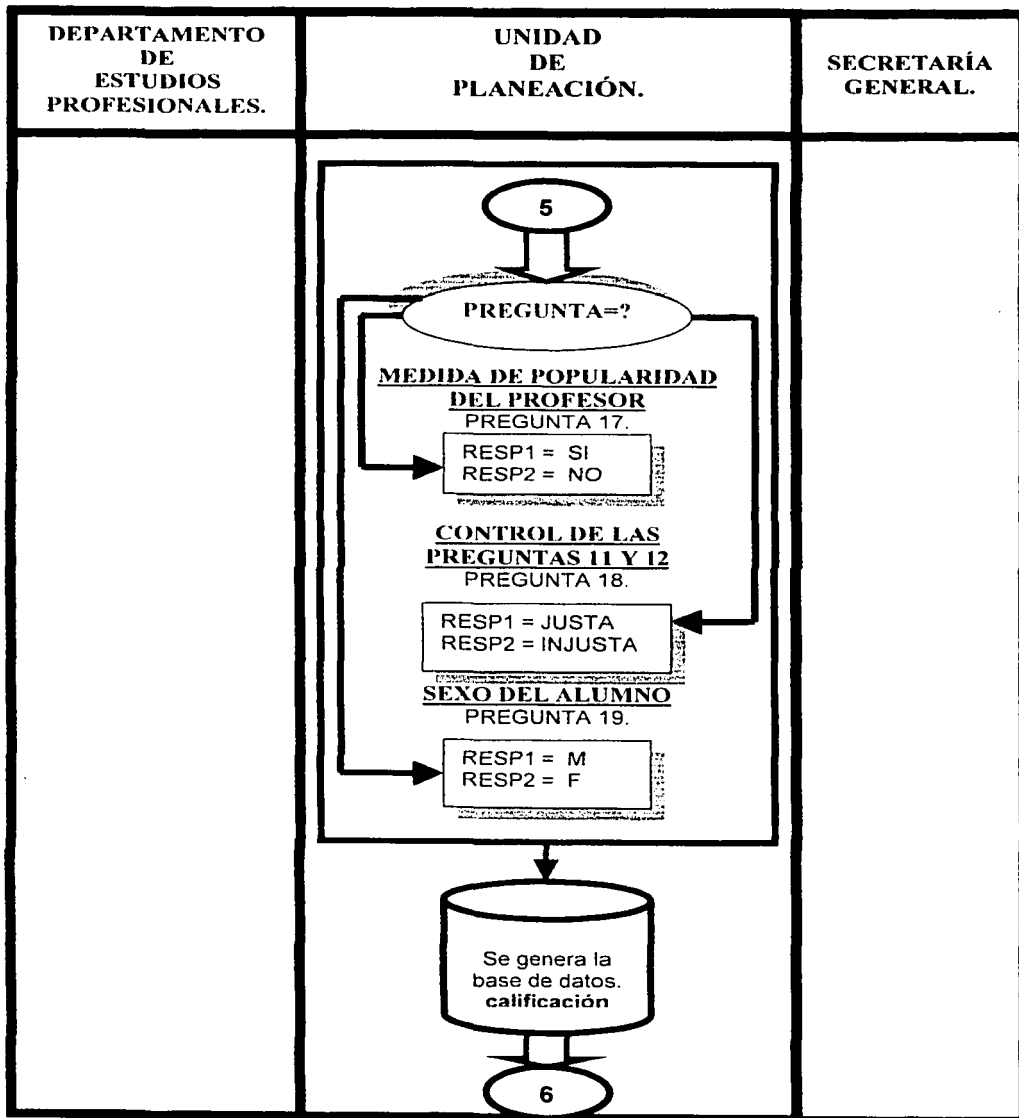




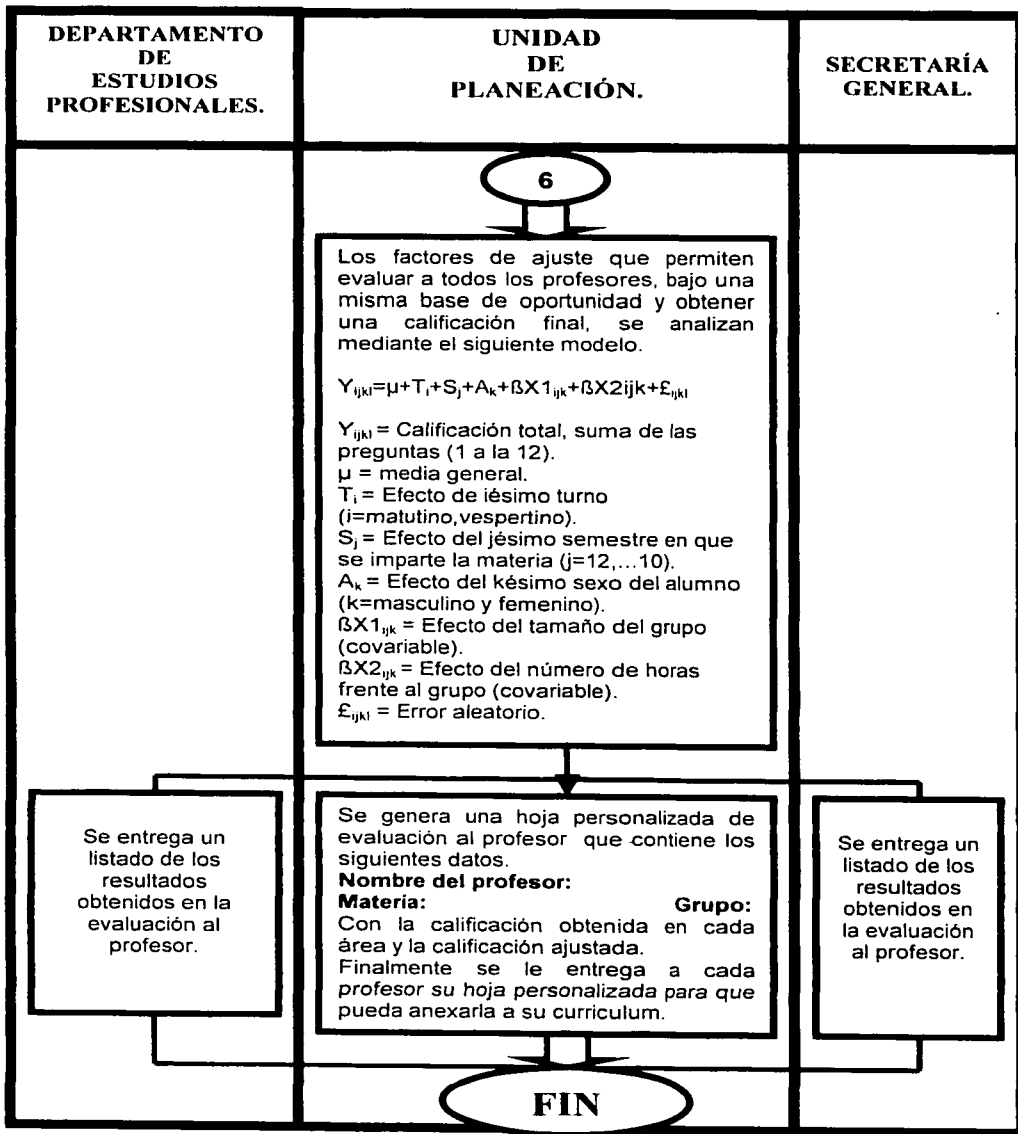
**TESIS CON FALLA DE ORIGEN**



TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN



TESIS CON FALLA DE ORIGEN

### III.IV ANÁLISIS DEL REQUERIMIENTO.

Los requerimientos de los alumnos de la Facultad de Medicina Veterinaria y Zootecnia, se exponen a continuación.

- Claridad en la información
- Incluir la ayuda necesaria que describa el correcto seguimiento del proceso.
- No existan problemas con el tamaño de información.
- Fácil manipulación de información.
- No exista pérdida de información.
- Seguridad en la información.
- Rápido acceso a la información.
- Capacidad de revisar resultados de diferentes semestres.
- Rapidez en la obtención de los resultados.

De acuerdo a las necesidades planteadas anteriormente, se concluyó establecer un Sistema de Evaluación vía Internet, que cuente con un servidor de base de datos encargado de la transferencia de archivos, permitiendo a los usuarios un acceso a la información más cómoda, así como facilitar el proceso de dicha información para la obtención de resultados.

Todos estos elementos establecerán comunicación mediante el protocolo TCP/IP, en la figura 3.4 se muestra la relación establecida entre cada uno de ellos.

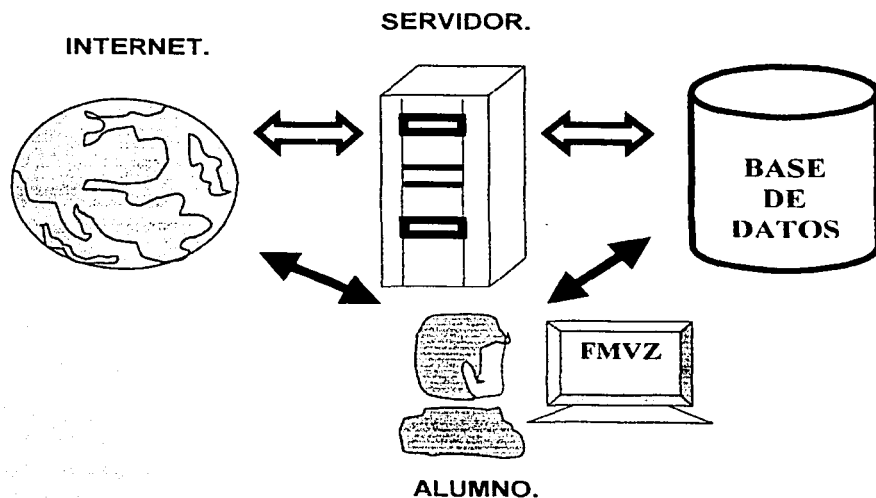


Figura 3.4 Sistema Propuesto.

TESIS CON  
FALLA DE ORIGEN

Debido a que la herramienta de programación a utilizar, para el desarrollo del Sistema de Evaluación de Profesores de la Facultad de Medicina Veterinaria y Zootecnia, será un lenguaje orientado a objetos (Java). Para documentar, visualizar y especificar cada una de las partes del desarrollo del software se utilizará el Lenguaje Unificado para la Construcción de Modelos (UML). En esta etapa de análisis utilizaremos 2 tipos de diagramas, uno para ejemplificar los requerimientos del sistema (diagramas de casos de uso) y el otro para describir el comportamiento del sistema (diagrama de secuencias).

### DIAGRAMA DE CASOS DE USO.

El diagrama de casos de uso representa la forma en como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso).

Un diagrama de casos de uso consta de los siguientes elementos.

#### Actor.

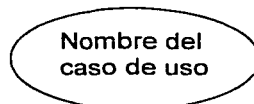
Una definición previa, es que un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.



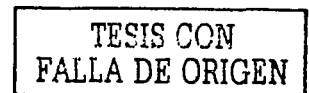
Actor.

#### Caso de uso.

Es una operación/tarea específica que realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

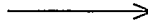


Caso de uso.



**Relaciones.**

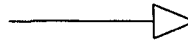
- **Asociación.** Este tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.



- **Dependencia o Instanciación.** Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.



- **Generalización.** Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de uso (<<uses>>) o de herencia (<<extends>>).



**DIAGRAMA DE CASOS DE USO DEL SISTEMA.**

A continuación se presenta en la siguiente figura el diagrama de casos de uso, el cuál nos permite organizar y modelar el comportamiento del sistema.

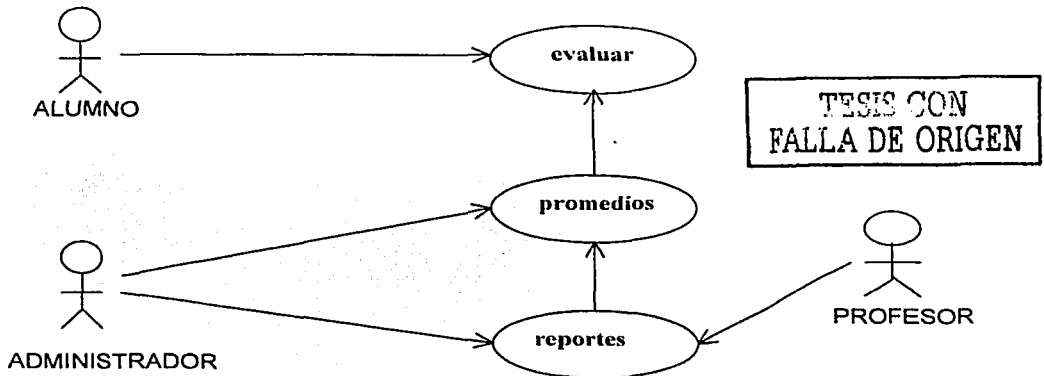


Figura 3.5 Diagrama de casos de uso del Sistema.

Para una mejor comprensión se presenta a continuación una tabla, con la descripción de cada caso de uso mostrados en el diagrama de la figura 3.5.

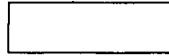
<b>CASO DE USO</b>	<b>ACTOR</b>	<b>DESCRIPCIÓN</b>
<b>evaluar</b>	<b>Alumno</b>	Para comenzar con el registro de evaluaciones, es necesario que el alumno introduzca su número de cuenta, posteriormente se realizarán consultas a la base de datos para determinar su posible validación y con esto el acceso al sistema.
<b>promedios</b>	<b>Usuario final (Administrador)</b>	Una vez concluido el proceso de evaluación, las respuestas obtenidas serán almacenadas en una base de datos, para un posterior cálculo de promedios correspondiente, a la asignatura, grupo y profesor evaluado. Para la realización de este proceso se realizará, primeramente la validación de la clave correspondiente al usuario en cuestión.
<b>reportes</b>	<b>Usuario final (Administrador) (Profesor)</b>	Finalmente deberá realizarse un reporte por asignatura, grupo y profesor que muestre los resultados obtenidos en las diferentes áreas evaluadas. De igual manera que en el caso de uso anterior, para realizar este proceso se tendrá que validar la clave o rfc correspondiente al usuario en cuestión.

Tabla de Casos de uso.

### DIAGRAMA DE SECUENCIAS.

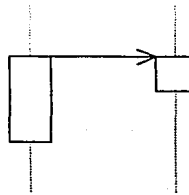
Un diagrama de secuencias, representa la forma en como un cliente (actor) u objetos (clases) se comunican entre si en petición a un evento. Esto implica recorrer toda la secuencia de llamadas, de donde se obtienen las responsabilidades claramente.

Los componentes de un diagrama de secuencias son:



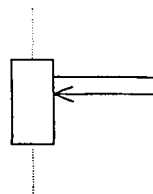
El rectángulo representa una instancia de un objeto en particular, y la línea punteada representa las llamadas a métodos del objeto.

#### Mensaje a otro objeto.



Se representa por una flecha entre un objeto y otro, representa la llamada de un método (operación) de un objeto en particular.

#### Mensaje al mismo objeto.



TESIS CON  
FALLA DE ORIGEN

A continuación se muestran los siguientes diagramas de secuencias que describen el comportamiento del sistema de evaluación de profesores.

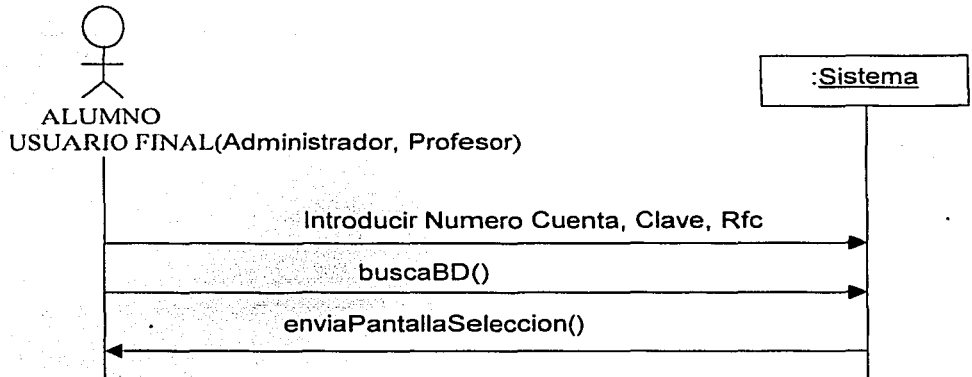


Figura 3.6 Diagrama del Sistema para el proceso validación(acceso).

La figura 3.7 muestra la secuencia de NoAcceso, cuando el usuario introduce su número de cuenta, clave o rfc el sistema debe ser capaz de no permitir el acceso en caso de ser incorrecto.

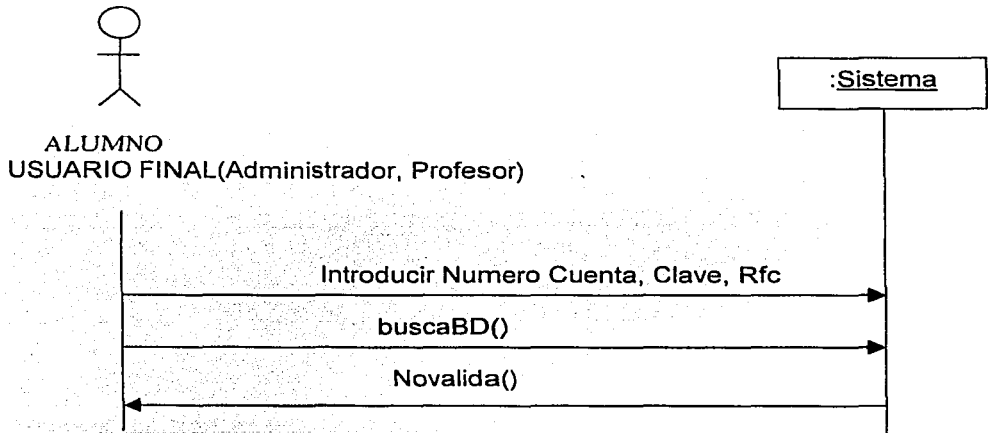


Figura 3.7 Diagrama de secuencia para el proceso validación(NoAcceso).

Una vez que el usuario tenga acceso al sistema, se desplegara una página que contendrá la información sobre su inscripción del semestre anterior, donde el alumno deberá seleccionar al profesor a evaluar.

Cuando se ha concluido la selección de respuestas del cuestionario de evaluación, se procede al almacenamiento de información A continuación se muestra este proceso en la figura 3.8.

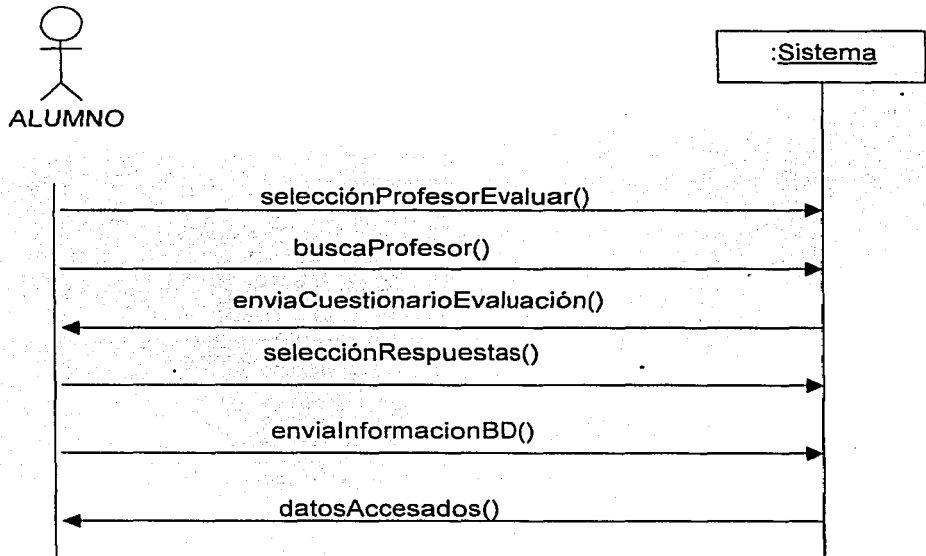


Figura 3.8 Diagrama del Sistema para el proceso evaluar.

TESIS CON  
FALLA DE ORIGEN

La información almacenada en la base de datos será procesada, mediante un Servlet que se encargará de realizar los cálculos necesarios, para la obtención de promedios y ajuste de la calificación final del profesor evaluado. Este proceso se describe en la figura 3.9.

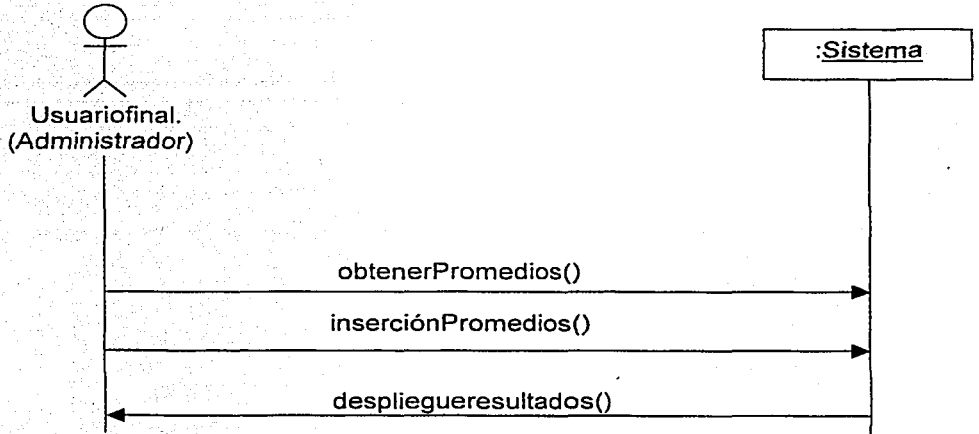


Figura 3.9 Diagrama del Sistema para el proceso Promedios.

Al final de este proceso se podrá efectuar un reporte a solicitud del Administrador o bien del Profesor, cada uno para tener acceso deberán proporcionar una clave y de ser correcta, tendrán que elegir el criterio de búsqueda.

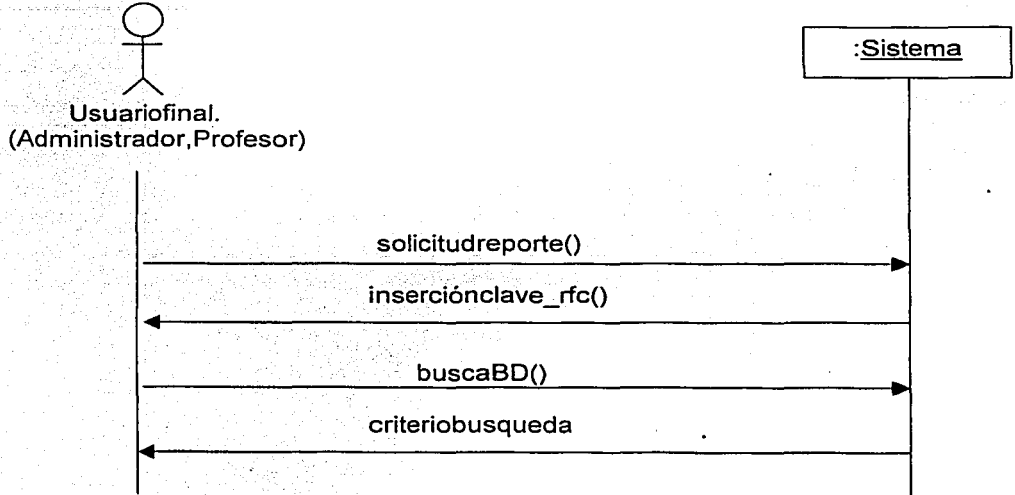


Figura 3.10 Diagrama del Sistema para el proceso Reportes.

# **CAPÍTULO IV**

## **DISEÑO DEL SISTEMA.**

---

### **IV.1 PRINCIPIOS DEL DISEÑO DE SOFTWARE.**

El Diseño del Sistema se define como el proceso de aplicar técnicas y principios con el propósito de definir un sistema con suficientes detalles como para permitir su interpretación y realización física.

A partir de la especificación de los flujos de datos resultantes del análisis, nos muestra en forma general los procesos que habrán de ser implementados.

En ésta fase de diseño se detalla y materializa con precisión los requerimientos del cliente.

Antes de comenzar con el proceso de diseño, primeramente se analizarán los recursos disponibles para determinar la plataforma de trabajo.

Herramientas de Software a utilizar para la elaboración del Sistema de Control de Evaluación de Docencia y Desempeño Académico por Internet.

#### **Sistema Operativo.**

*Para una aplicación de Internet es necesario disponer de un sistema operativo que brinde servicios a múltiples usuarios y que sea capaz de atender varias tareas al mismo tiempo.*

Los principales tipos de sistemas operativos multitarea y multiusuario son los de Windows NT de Microsoft y los de Unix. Los Sistemas Operativos de tipo Windows NT, incluyen las versiones de Windows NT y 2000.

El sistema operativo se instala en el equipo que servirá como servidor, donde se requiere atender a varios usuarios, que consulten la información a través de Internet.

En este caso el sistema operativo disponible en la Unidad de Planeación, departamento encargado de realizar la mayor parte del proceso de Evaluación, a los Profesores de la Facultad de Medicina Veterinaria y Zootecnia, es Windows 2000 elegido por su fácil manejo y popularidad.

#### **Sistemas manejadores de bases de datos.**

Los sistemas manejadores de bases de datos son programas que organizan información de manera que su consulta, almacenamiento y modificación se realiza de forma más eficiente.

Los dos principales tipos de sistemas manejadores de bases de datos son: Sistemas Manejadores de Bases de Datos Relacionales y los, Orientados a Objetos. Sus diferencias radican en la forma de administrar la base de datos. Dado que el sistema operativo a utilizar para el desarrollo del sistema es Windows 2000, la base de datos relacional empleada será Access 2000, descrita más detalladamente a continuación.

### **Access 2000.**

Microsoft Access 2000 es una aplicación de bases de datos relacionales, cuyas características la hacen fácil de usar y compartir información de una base de datos.

#### **Características.**

**Asistentes.** Los asistentes ayudan a crear tablas y consultas.

**Autocorrección de nombres.** Cuando se renombra un campo (columna) en una tabla, el cambio es automáticamente actualizado en todas las tablas dependientes, consultas y formas.

**Arrastrar y pegar a Excel.** Se puede exportar información de Access 2000 a Excel 2000 simplemente arrastrando los objetos.

**Compactación.** Las bases de datos pueden crecer en tamaño rápidamente y ocupa espacio del disco duro. Access automáticamente comprime una base de datos.

**Páginas de acceso de datos.** Se puede crear páginas Web de los objetos de la base de datos, es un proceso simple usando esta nueva características de Access 2000.

#### **Ventajas.**

- Prácticamente es gratuito, ya que forma parte de Microsoft Office y como es del conocimiento general, estos programas los poseen casi todas las empresas.
- Es fácil de usar y administrar sus tablas, además permite el rápido desarrollo de aplicaciones.
- Es poderoso para proyectos pequeños y medianos.
- Compacta automáticamente la base de datos.
- Su base de datos es escalable a SQL Server, el cual puede soportar proyectos mayores.

#### **Desventajas.**

- Existen en el mercado manejadores de bases de datos que pueden soportar proyectos mayores, sin embargo Access puede escalar la base de datos a este tipo de manejadores.

### **Servidor Web.**

El servidor de Web es un programa encargado de enviar el código HTML y otros componentes al navegador de Web.

Es también un elemento importante que permite interactuar al cliente con el servidor. El cliente realiza las peticiones al servidor y el servidor procesa la petición y regresa una respuesta.

Existen distintos servidores de Web que pueden ser comerciales o de distribución libre.

Para elegir un servidor Web, es necesario tomar en cuenta lo siguiente:

- La plataforma en la que funciona.
- Su rendimiento.
- Si es libre o comercial y,
- La herramienta de programación utilizada, ya que va totalmente ligada a la elección del servidor.

Tomando como base lo anteriormente citado, el servidor elegido para el desarrollo de este sistema es el Servidor Tomcat.

### **Servidor Tomcat.**

El servidor Tomcat es una aplicación web basada en Java creada para ejecutar servlets y páginas JSP.

Por definición como antecedente, antes de entrar en la instalación y configuración de Tomcat, es necesario tener un conocimiento básico del concepto de Aplicación Web.

Una aplicación web es una colección de servlets, páginas JSP, clases Java, documentos estáticos: HTML, XHTML, e imágenes.

Una vez realizada la instalación de Tomcat, el sistema estará en condiciones de arrancar el servidor web y atender peticiones a través de Tomcat.

## Lenguaje de Programación.

### Java.

Es un lenguaje de programación independiente de la plataforma, debido a que se ha creado una Máquina de Java (compilador) para cada sistema que hace de puente entre el sistema operativo y el programa de Java.

Las principales características de Java son:

- **Portabilidad.** Debido a la máquina virtual de Java que interpreta el código de bytes y lo transforma en código que puede correr en equipo real del ordenador, las diferencias individuales de las plataformas de los ordenadores, pueden ser reconocidas y adaptadas localmente mientras el programa se está ejecutando.
- **Compatibilidad.** No se requieren versiones específicas de programas para cada plataforma.
- **Seguridad.** La máquina virtual de Java realiza una serie de verificaciones de cada objeto para checar su integridad, evitando que una instrucción pueda tener la dirección de almacenamiento de datos en otra aplicación o en el sistema operativo, ocasionando de esta forma la caída del sistema.
- **Objetos.** Java es un lenguaje de programación orientado a objetos, de tal forma que objetos similares pueden aprovechar el hecho de que son parte de una misma clase y heredar código en común.

La elección de este lenguaje como herramienta para el desarrollo del Sistema de Evaluación, se debió a la importancia que tiene para Internet por su independencia de plataforma, característica que soluciona el problema ocasionado con el acceso a Internet con ordenadores distintos.

El lenguaje Java proporciona varias formas de usarlo dentro de un servidor Web, dos de ellas se mencionan a continuación.

Mediante **servlets** pequeños programas en Java que se ejecutan de forma persistente en el servidor, y que por lo tanto tienen una activación muy rápida, cuyas características son las siguientes:

- Tienen un marco de trabajo similar al lenguaje de programación Java.
- Son independientes de la plataforma y están soportados en la mayoría de servidores web.
- Una vez que son llamados por primera vez, quedan activos en la memoria del servidor, logrando con esto una mayor eficiencia en cuanto a tiempo de respuesta hacia el usuario del sistema.

- Se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, permitiendo con esto el desarrollo de aplicaciones distribuidas y fácilmente escalables.
- Permiten redireccionar peticiones de servicios a otros servlets, en la misma máquina o en una máquina remota.
- Los Servlets no tienen entorno gráfico ya que se ejecutan en el servidor.
- Reciben unos datos y su salida o respuesta son principalmente ficheros de texto HTML.
- Los servlets son desarrollados utilizando el API Java Servlet, que es una extensión de Java.

## **IV.II ARQUITECTURA DE LA APLICACIÓN.**

La arquitectura describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo.

En la arquitectura se estructuran los elementos de un software para:

- Comprender el sistema.
- Organizar el desarrollo.
- Fomentar la reutilización y
- Hacer evolucionar el sistema.

### **Arquitectura de capas en Sistemas de Información.**

Existen propuestas de arquitecturas de capas para Sistemas de Información, donde las capas a veces reciben el nombre de niveles:

- Arquitectura de tres capas;
- Arquitectura de dos capas;

### **Arquitectura de tres capas.**

La arquitectura de tres capas abarca una interfaz para el usuario y el almacenamiento de datos.

A continuación se realiza una descripción de las capas que se muestran en la figura 4.9.

1. Capa de presentación.
  - Involucra presentar los resultados a los usuarios del sistema y recoger las entradas del usuario.
2. La capa lógica de aplicación.
  - Involucra proveer funcionalidad específica a la aplicación, por ejemplo en un sistema de banco, las funciones del banco como abrir una cuenta, cerrar una cuenta, etc.
3. La capa de almacenamiento de datos.
  - Involucra administrar las bases de datos del sistema.

**Característica de las arquitecturas de 3 capas:**

- Posibilidad de crear diferentes interfaces para la misma lógica de negocio.

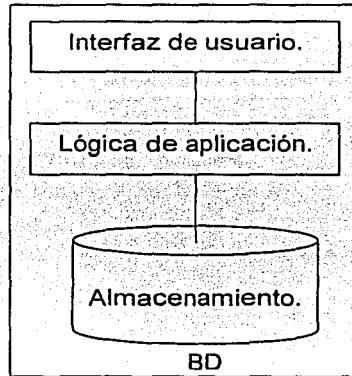


Figura 4.1 Esquema de arquitectura de 3 capas.

**Arquitectura de 2 capas.**

**Características de las arquitecturas de 2 capas:**

- Mucha carga en el cliente.
- Poca carga en el servidor.
- Mucho tráfico en la red.
- Mantenimiento costoso, en cada cliente.
- Posibilidad de clientes desfasados.

TESIS CON  
FALLA DE ORIGEN

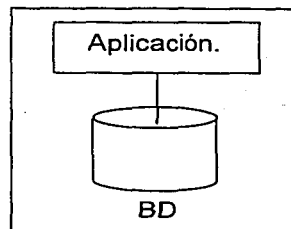


Figura 4.2 Esquemas de arquitecturas de 2 capas.

### **Diseño de la base de datos**

Para realizar el diseño de la base de datos se hace uso de la herramienta gráfica, el diagrama Entidad-Relación. En esta parte se define la relación entre cada uno de los elementos estructurales del sistema.

El modelo entidad-relación consiste en un conjunto de objetos llamados entidades y las relaciones entre ellas. Con el diagrama entidad/relación se presenta la estructura básica general de la base de datos gráficamente.

El diagrama relacional nos permite identificar entidades, describir entidades y establecer las relaciones entre entidades.

El diccionario de datos se encuentra formado por una serie de definiciones que se expresan en un lenguaje especial llamado, lenguaje de definición de datos, con el cual se definen o declaran los objetos de la base de datos.

El lenguaje de definición de datos permite a los usuarios manipular o procesar dichos objetos, por medio de un modelo apropiado de datos.

*A continuación se presenta en la figura 4.3 el diagrama entidad/relación con las entidades necesarias para la elaboración de la base de datos.*

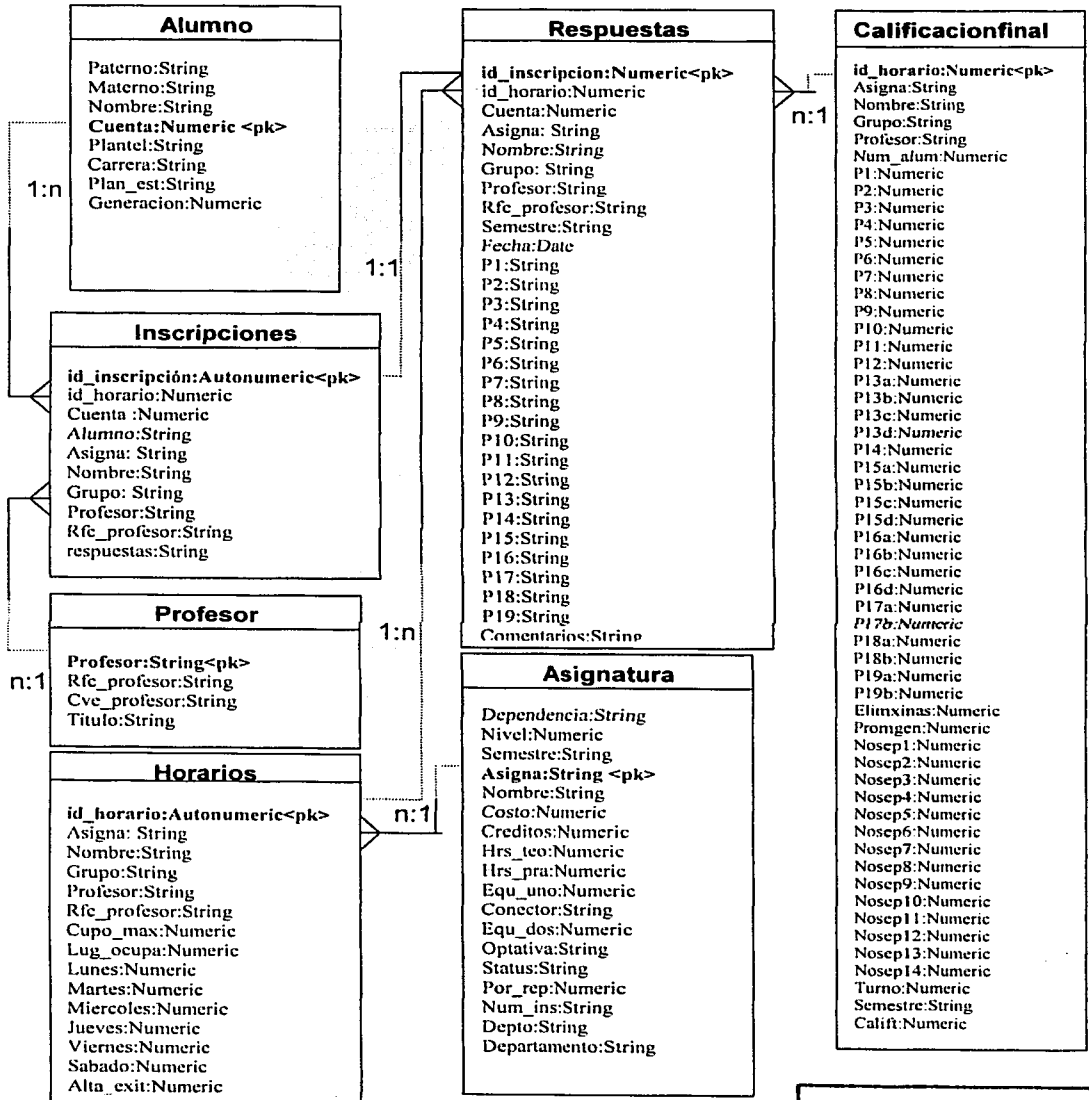


Figura 4.3 Diagrama Entidad/Relacion.

TESIS CON FALLA DE ORIGEN

A continuación en las tablas 1.a, 1.b, 1.c y 1.d se muestra el Diccionario de Datos para el Sistema de Evaluación de Profesores de la Facultad de Medicina Veterinaria y Zootecnia.

Entidad	Atributo	Tipo	Tamaño
Asignatura	Dependencia	String	50
	Nivel	Numeric	50
	Semestre	String	50
	Asigna	String <pk>	50
	Nombre	String	255
	Costo	Numeric	30
	Creditos	Numeric	30
	Hrs_teo	Numeric	30
	Hrs_pra	Numeric	30
	Equ_uno	Numeric	30
	Conector	String	30
	Equ_dos	Numeric	30
	Status	String	30
	Por_rep	Numeric	30
	Num_ins	String	50
Depto	String	50	
Departamento	String	50	

Entidad	Atributo	Tipo	Tamaño
Horario	id_horario	Autonumeric<pk>	10
	Asigna	String	50
	Nombre	String	50
	Grupo	String	50
	Profesor	String	255
	Rfc_profesor	String	50
	Semestre	String	50
	Cupo_max	Numeric	10
	Lug_ocupa	Numeric	10
	Lunes	Numeric	30
	Martes	Numeric	30
	Miércoles	Numeric	30
	Jueves	Numeric	30
	Viernes	Numeric	30
	Sabado	Numeric	30
Alta_exi	Numeric	30	

Tabla 1.a Diccionario de datos.

TESIS CON  
FALLA DE ORIGEN

<b>Alumno</b>	Paterno	String	50
	Materno	String	50
	Nombre	String	50
	<b>Cuenta</b>	<b>Numeric &lt;pk&gt;</b>	30
	Plantel	String	30
	Carrera	String	30
	Plan est	Numeric	30
	Generación	Numeric	30
<b>Inscripciones</b>	<b>id_inscripcion</b>	<b>Autonumeric&lt;pk&gt;</b>	10
	Id horario	Numeric	10
	Cuenta	Numeric	50
	Alumno	String	255
	Asigna	String	50
	Nombre	String	255
	Grupo	String	50
	Profesor	String	255
Rfc_profesor	String	50	
	respuesta	String	10
<b>Profesor</b>	<b>Profesor</b>	<b>String &lt;pk&gt;</b>	255
	Rfc_profesor	String	50
	Cve_profesor	String	50
	Titulo	String	255
<b>Respuestas</b>	<b>id_inscripcion</b>	<b>Numeric&lt;pk&gt;</b>	10
	Cuenta	Numeric	50
	id_horario	Numeric	10
	Asigna	String	50
	Nombre	String	255
	Grupo	String	50
	Profesor	String	255
	Rfc_profesor	String	50
	Semestre	String	50
	Fecha	Date	dd/mm/aa
	P1	Character	2
P2	Character	2	
P3	Character	2	

Tabla 1.b Diccionario de datos.

TESIS CON  
FALLA DE ORIGEN

<b>Respuestas</b>	P4	Character	2
	P5	Character	2
	P6	Character	2
	P7	Character	2
	P8	Character	2
	P9	Character	2
	P10	Character	2
	P11	Character	2
	P12	Character	2
	P13	Character	2
	P14	Character	2
	P15	Character	2
	P16	Character	2
	P17	Character	2
	P18	Character	2
	P19	Character	2
	Comentarios	String	255

<b>Calificacionfinal</b>	<b>id_horario</b>	<b>Numeric &lt;pk&gt;</b>	10
	Asigna	String	50
	Nombre	String	255
	Grupo	String	50
	Profesor	String	255
	Rfc_profesor	String	50
	Semestre	String	50
	Num_alum	Numeric	6
	P1	Numeric	10
	P2	Numeric	10
	P3	Numeric	10
	P4	Numeric	10
	P5	Numeric	10
	P6	Numeric	10
	P7	Numeric	10
	P8	Numeric	10
	P9	Numeric	10
	P10	Numeric	10
	P11	Numeric	10
	P12	Numeric	10

Tabla 1.c Diccionario de datos.

TESIS CON  
FALLA DE ORIGEN

<b>Calificacionfinal</b>	P13a	Numeric	10
	P13b	Numeric	10
	P13c	Numeric	10
	P13d	Numeric	10
	P14	Numeric	10
	P15a	Numeric	10
	P15b	Numeric	10
	P15c	Numeric	10
	P15d	Numeric	10
	P16a	Numeric	10
	P16b	Numeric	10
	P16c	Numeric	10
	P16d	Numeric	10
	P17a	Numeric	10
	P17b	Numeric	10
	P18a	Numeric	10
	P18b	Numeric	10
	P19a	Numeric	10
	P19b	Numeric	10
	Elimxinas	Numeric	10
	Promgen	Numeric	10
	Nosep1	Numeric	10
	Nosep2	Numeric	10
	Nosep3	Numeric	10
	Nosep4	Numeric	10
	Nosep5	Numeric	10
	Nosep6	Numeric	10
	Nosep7	Numeric	10
	Nosep8	Numeric	10
	Nosep9	Numeric	10
	Nosep10	Numeric	10
	Nosep11	Numeric	10
Nosep12	Numeric	10	
Nosep13	Numeric	10	
Nosep14	Numeric	10	
Turno	Numeric	10	
Calift	Numeric	10	

Tabla 1.d Diccionario de datos.

**TESIS CON  
FALLA DE ORIGEN**

### IV.III ESTRUCTURA DE PROCESOS.

Utilizando la herramienta UML(Lenguaje Unificado de Construcción de Modelos), en especial su diagrama de secuencia y de clase, se especificará la estructura de cada caso de uso que constituye el sistema propuesto.

#### DIAGRAMA DE CLASES.

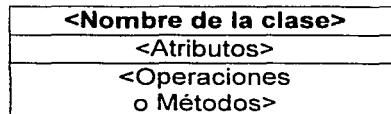
Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento.

Un diagrama de clases esta compuesto por los siguiente elementos:

#### CLASE (atributos, métodos y visibilidad):

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es un instancia de una clase).

En UML, una clase es representada por un rectángulo que posee tres divisiones:



TESIS CON FALLA DE ORIGEN
------------------------------


Donde:



- **Superior:** Contiene el nombre de la clase.
- **Intermedio:** Contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).
- **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: private, protected o public).

#### Métodos y Atributos:

##### Atributos.




Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

- Public (+,  ): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.

- **Private**(- ,  ): Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos los puede acceder).
- **Protected** (#,  ): Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven.

### Métodos.

Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:

- **Public** (+,  ): Indica que el atributo será visible tanto dentro como fuera de la clase, es decir, es accesible desde todos lados.
- **Private**(- ,  ): Indica que el atributo sólo será accesible desde dentro de la clase (sólo sus métodos los puede acceder).
- **Protected** (#,  ): Indica que el atributo no será accesible desde fuera de la clase, pero si podrá ser accesado por métodos de la clase además de las subclases que se deriven.

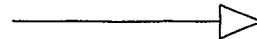
### RELACIONES ENTRE CLASES (Herencia, Composición, Agregación, Asociación y Uso).

En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser.

- Uno a muchos: 1...\*(1...n)
- 0 o muchos: 0...\*(0...n)
- número fijo: m(m denota el número).

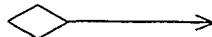
TESIS CON  
FALLA DE ORIGEN

### Herencia (Especialización/Generalización).



Indica que una subclase hereda los métodos y atributos especificados por un Super Clase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la Super Clase (public, protected).

### Agregación.



Cuando se requiere componer objetos que son instancias de clases definidas por el desarrollador de la aplicación, tenemos dos posibilidades.

Por valor. Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido esta condicionado por el tiempo de vida del que lo incluye. Este tipo de relación comúnmente llamada Composición (el Objeto base se construye a partir del objeto incluido).

Por referencia. Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada Agregación (el objeto base utiliza al incluido para su funcionamiento).

**Asociación:**                   —————>

La relación entre clases conocida como Asociación, permite asociar objetos que colaboran entre si. Cabe destacar que no es una relación fuerte, es decir, el tiempo de vida de un objeto no depende del otro.

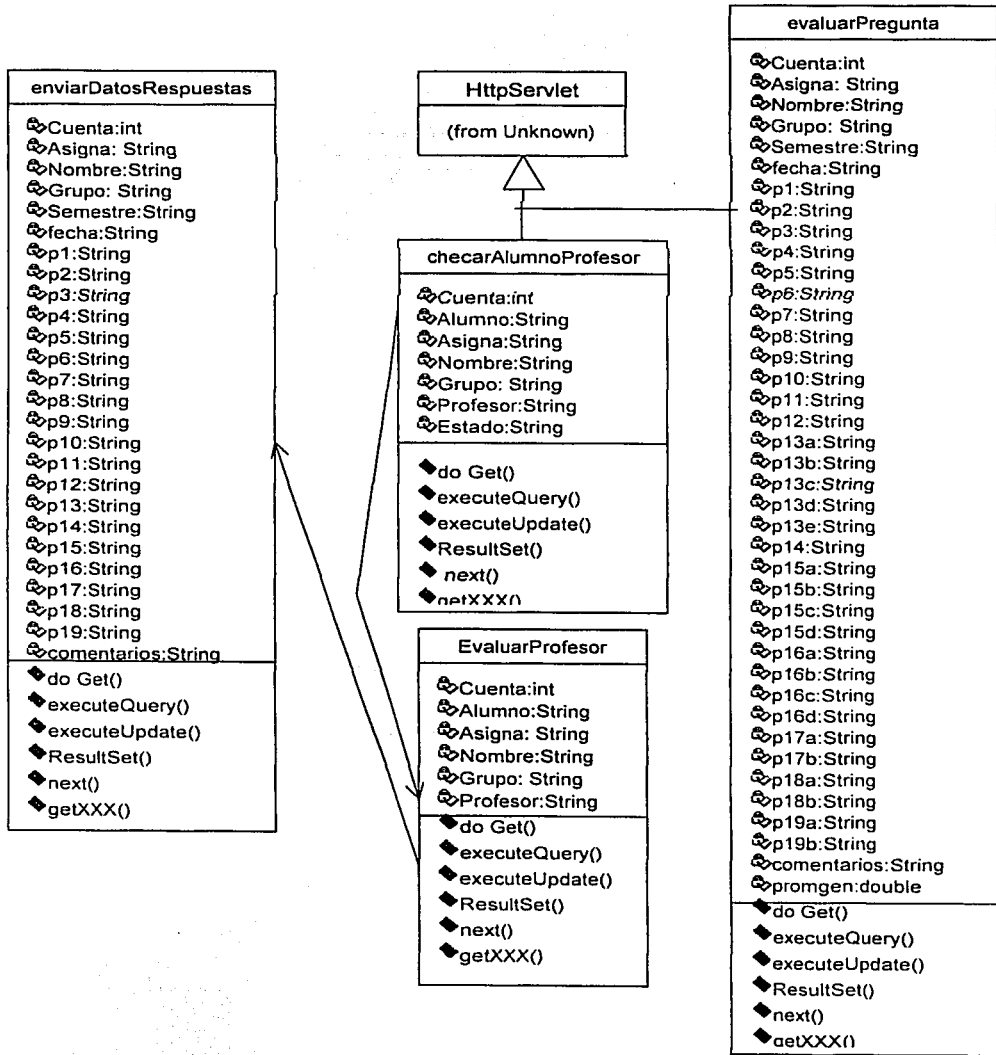
**Dependencia o Instanciación (uso).**                   —————>

Representa un tipo de relación muy particular , en la que una clase es instanciada (su intanciación es dependiente de otro objeto/clase). Se denota por una flecha punteada.

#### **DIAGRAMA DE CLASES DEL SISTEMA.**

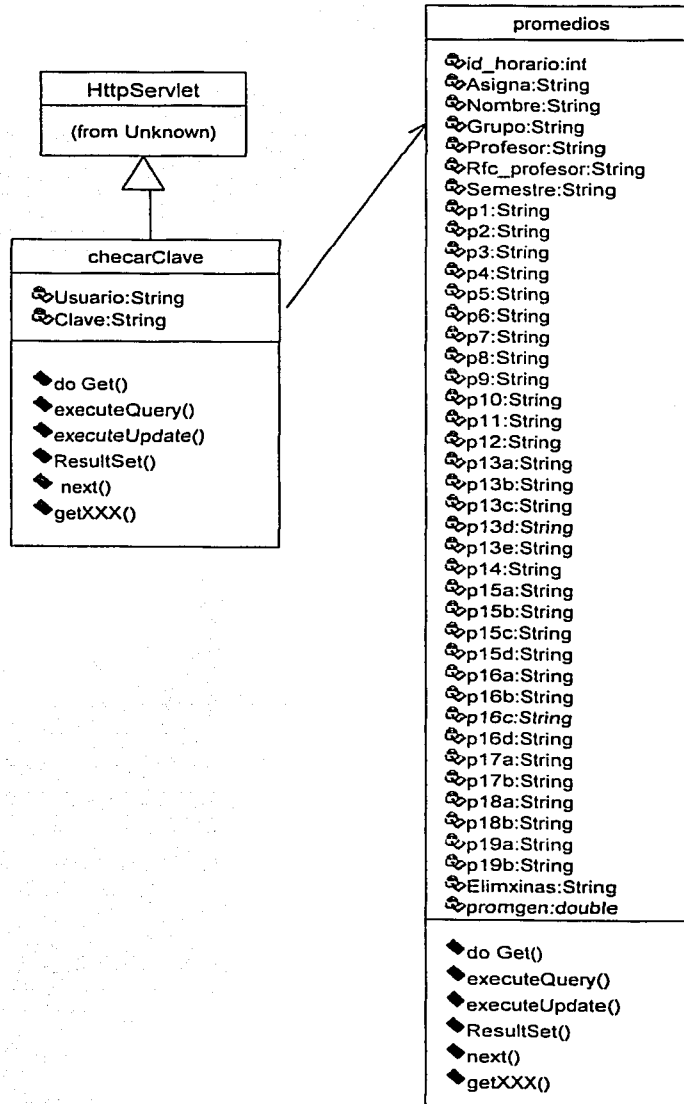
El diagrama de clases nos permite describir los objetos que componen nuestro sistema, así como las relaciones que existen entre ellos. En los siguientes diagramas de las figuras 4.4, 4.5 y 4.6 se muestran los objetos que constituyen los casos de usos utilizados en el sistema.

TESIS CON  
FALLA DE ORIGEN



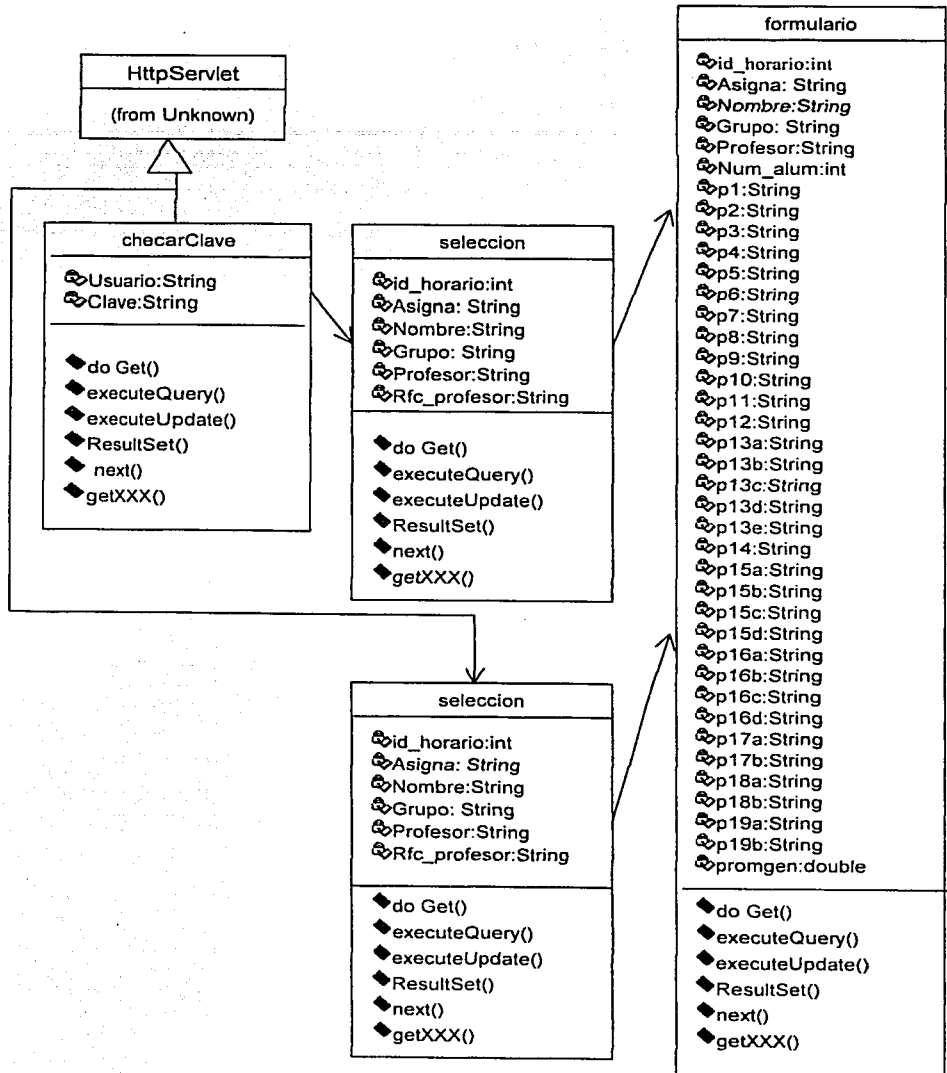
TESIS CON  
FALLA DE ORIGEN

Figura 4.4 Diagrama de clases del caso de uso evaluar.



TESIS CON  
 FALLA DE ORIGEN

Figura 4.5 Diagrama de clases del caso de uso promedios.



TESIS CON  
 FOLIO DE ORIGEN

Figura 4.6 Diagrama de clases del caso de uso reporte.

## DIAGRAMA DE SECUENCIAS.

Los diagramas de secuencia son modelos que describen la manera en que colaboran los grupos de objetos diseñados anteriormente.

A continuación se presentan los diagramas de secuencia que describen el funcionamiento del sistema entre los principales objetos que conforman los casos de uso mencionados se encuentran los siguientes.

### Diagrama de secuencias del caso de uso evaluar.

El diagrama de secuencias del **caso de uso evaluar** nos describe los pasos a seguir cuando un usuario (alumno) desea acceder al sistema de evaluación; será necesario que indique la URL del sistema por medio del navegador web, cuando esto ocurra se realizará un redireccionamiento hacia una página, que permitirá realizar el proceso de validación de número de cuenta, para ello el usuario (alumno) proporcionará su número de cuenta y pulsará el botón de Aceptar permitiendo que el **servlet checarAlumnoProfesor**, el cuál realizará una conexión a la base de datos y determinará si el usuario tiene una cuenta válida que le permita acceder al sistema.

Si la cuenta resulta afirmativa, el **servlet checarAlumnoProfesor** enviará una página con una tabla, la cual contendrá las asignaturas a evaluar, clave de la asignatura, nombre del profesor, grupo y estado de evaluación.

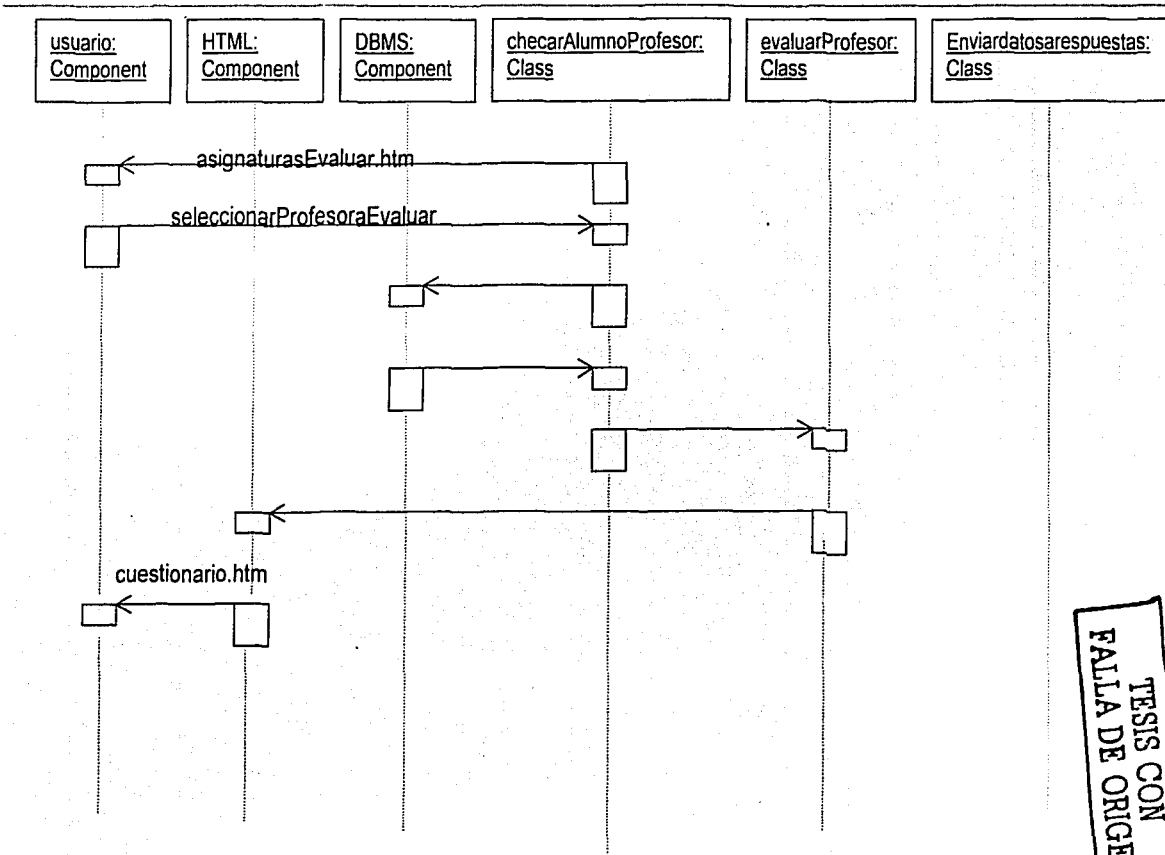
Dentro de esta tabla la liga para realizar alguna evaluación será el nombre del profesor, al realizar esta selección nos conectará con el **servlet evaluarProfesor**.

El **servlet evaluarProfesor** enviará una página con el contenido del cuestionario correspondiente al profesor seleccionado.

Una vez seleccionadas las opciones de respuesta del cuestionario, correspondiente al profesor en cuestión, serán almacenadas en una base de datos.

El **servlet enviardatosarespuestas** tendrá que establecer primero la conexión a la base de datos antes de introducir la información del cuestionario. Posteriormente deberá enviar una respuesta al usuario para confirmar que la información fue almacenada de manera satisfactoria, o en su defecto un mensaje de error indicando el problema ocurrido. A continuación se ejemplifica este proceso en las figuras 4.7.a, 4.7.b, 4.7.c.





TESIS CON  
FALLA DE ORIGEN

Figura 4.7.b Diagrama de Secuencias del Caso de uso evaluar.

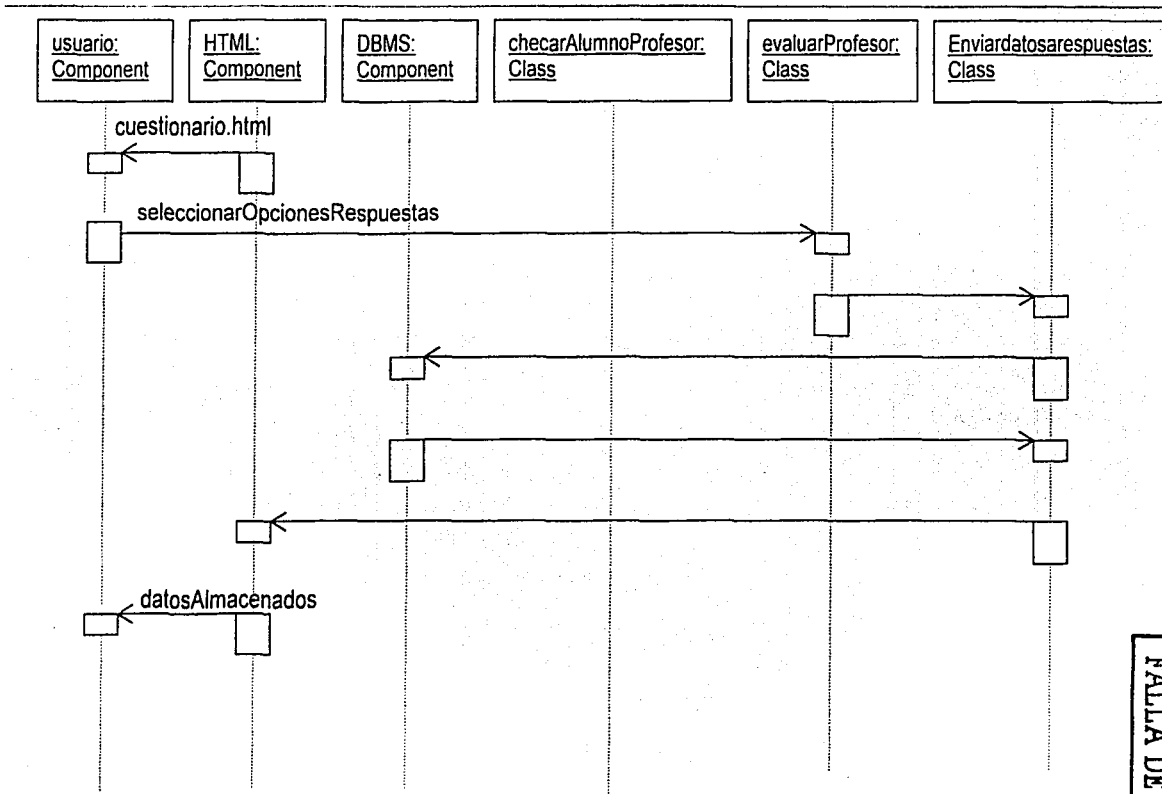


Figura 4.7.c. Diagrama de Secuencias del Caso de Uso evaluar.

TESIS CON  
 FALLA DE ORIGEN

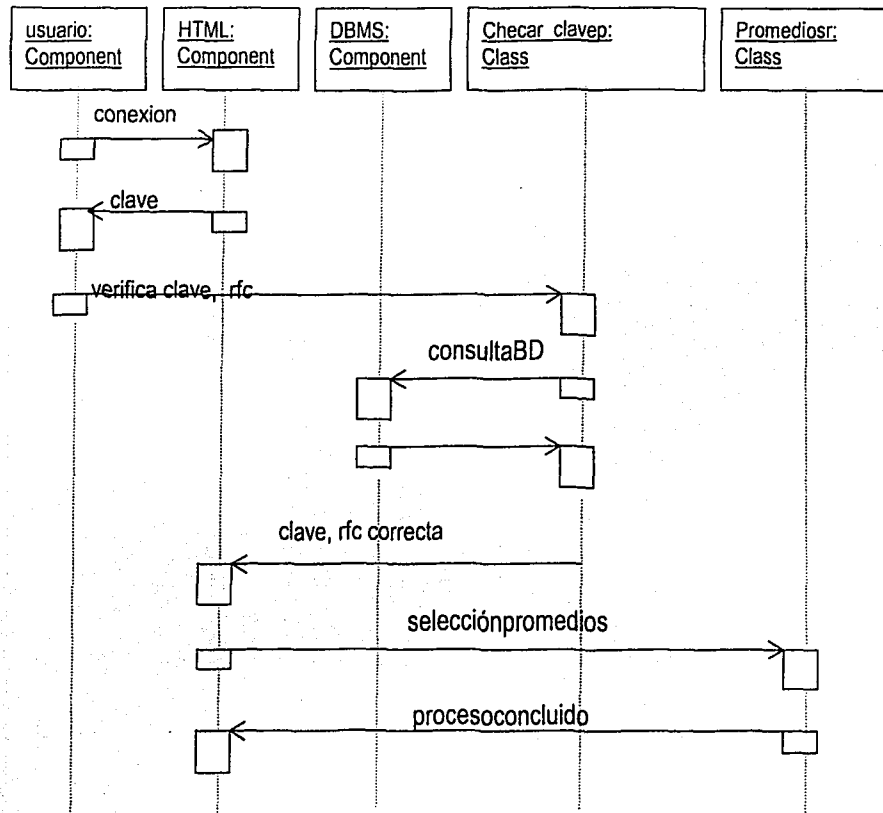
### **Diagrama de secuencias del caso de uso promedios.**

Una vez seleccionadas las opciones de respuesta del cuestionario, correspondiente al profesor en cuestión, serán almacenadas en una base de datos.

Esta información será procesada por el **servlet promedios**, entregando como resultado la calificación correspondiente a la asignatura grupo y profesor deseada.

Para poder efectuar este cálculo se tendrá que proporcionar al sistema una clave que sea válida, en caso contrario dicho proceso no podrá efectuarse.

A continuación se ejemplifica este proceso en la figura 4.8



TESIS CON  
 FALLA DE ORIGEN

Figura 4.8. Diagrama de Secuencias del Caso de uso Promedios.

**Diagrama de secuencias del caso de uso reportes.**

Los resultados obtenidos del proceso de evaluación serán resumidos a través de un reporte, el cuál podrá ser solicitado por el administrador proporcionando una clave de acceso, o bien por el profesor que de igual forma tendrá que introducir su registro federal de causantes (rfc) para tener acceso a la información solicitada.

A continuación se presenta en los diagrama 4.9.a y 4.9.b la secuencia a seguir por el usuario administrador y en los diagramas 4.10.a y 4.10.b la realizada por el profesor.

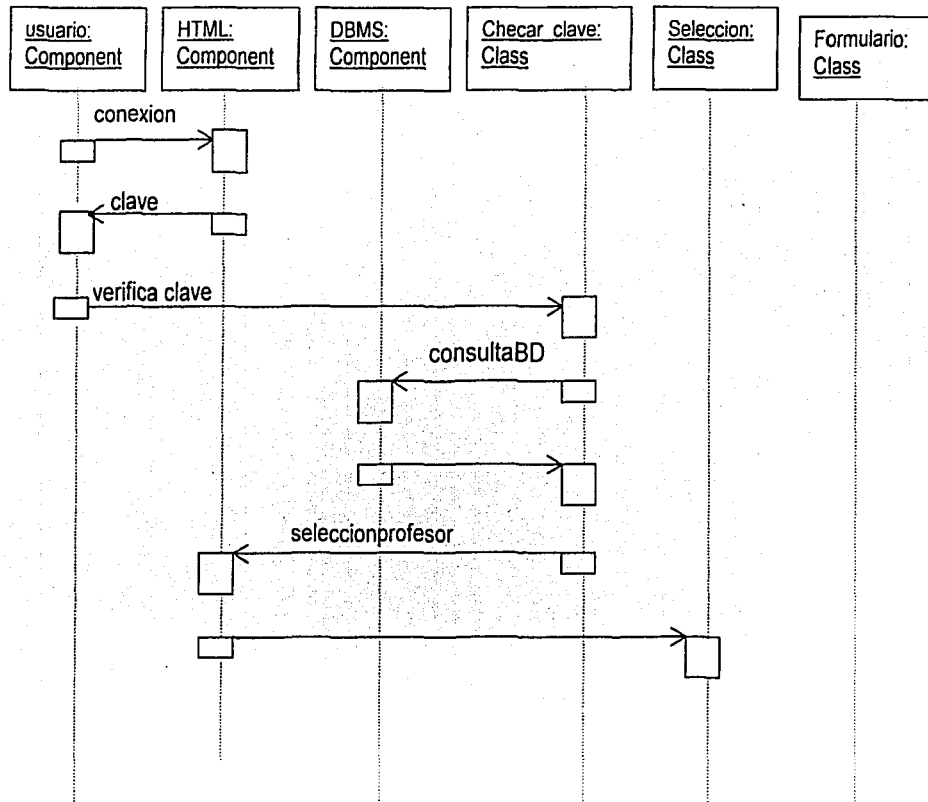
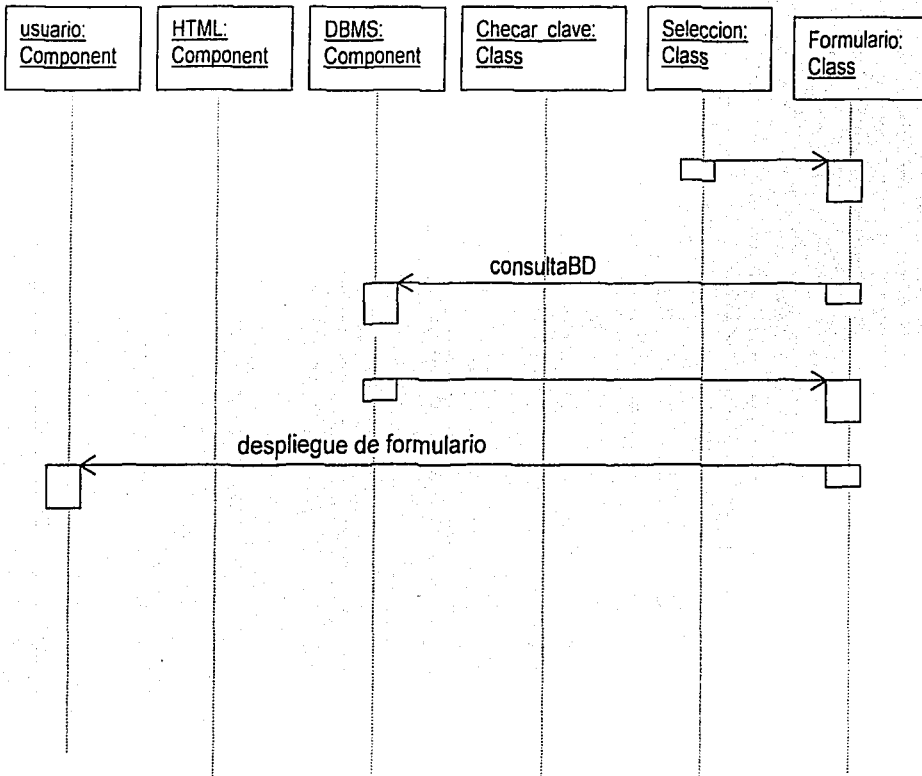


Figura 4.9.a Diagrama de Secuencias del Caso de uso reportes (clave).

TESIS CON  
FALLA DE ORIGEN



TESIS CON  
FALTA DE ORIGEN

Figura 4.9. b Diagrama de Secuencias del Caso de uso reportes (clave).

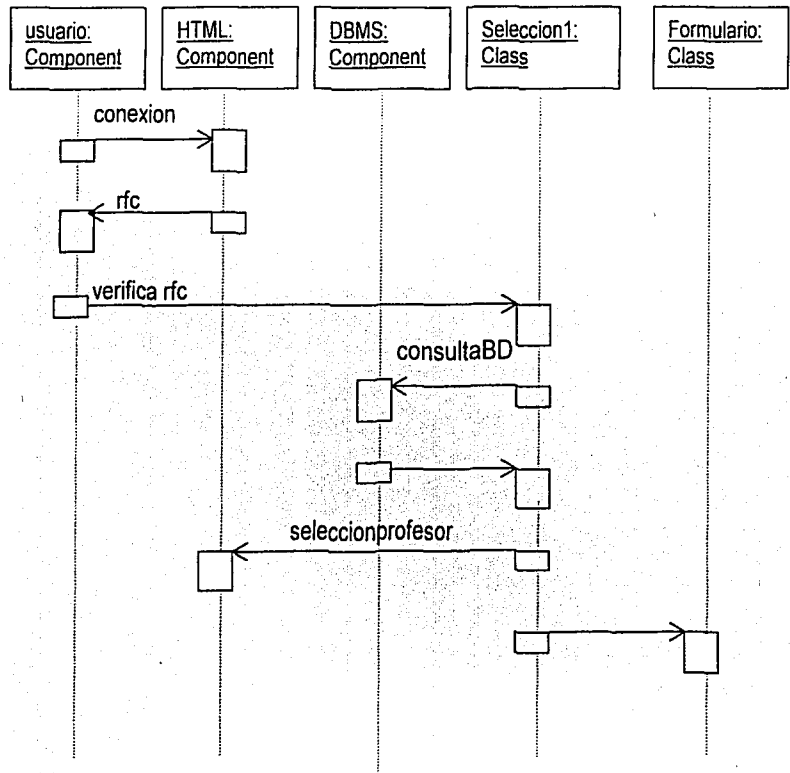


Figura 4.10.a Diagrama de Secuencias del Caso de uso reportes(rfc).

TESIS CON  
FALLA DE ORIGEN

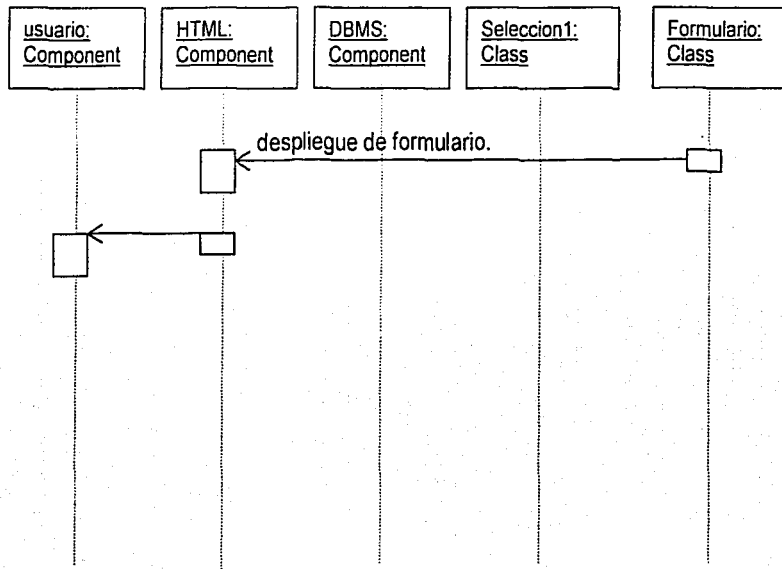


Figura 4.10.b Diagrama de Secuencias del Caso de uso reportes (rfc).

TESIS CON  
FOLLA DE ORIGEN

#### **IV.IV INTERFAZ CON EL USUARIO.**

La interfaz es una combinación de software y hardware que permite a un usuario comunicarse con la computadora.

El software se encargará de que la manipulación de objetos gráficos o el tecleo de comandos para la realización de tareas, resulte lo suficientemente entendible y sencillo de utilizar.

Por su parte el hardware se encargará de controlar la manipulación física de la computadora para el establecimiento de comunicación.

Tomando en cuenta lo anterior con el sistema a elaborar, se pretende que el alumno de la Facultad de Medicina Veterinaria y Zootecnia cuente con una herramienta fácil de entender y manipular.

Para lograr este objetivo el sistema se desarrollará utilizando ventanas y páginas web, donde se le indicará al usuario paso a paso la secuencia a seguir para dar inicio con la Evaluación de Profesores.

# **CAPÍTULO V CONSTRUCCIÓN.**

---

## **V.I. CONSIDERACIONES DEL AMBIENTE DE DESARROLLO.**

Para la implementación se utilizarán los recursos disponibles en el Departamento de Planeación de la Facultad de Medicina Veterinaria y Zootecnia.

El sistema se implementará en una máquina con las siguientes características en cuanto a hardware:

- PC con procesador Pentium IV.
- Disco duro de 40 GB.
- Memoria de 256 MB de RAM.
- Tarjeta de red.
- Impresora.

Con respecto al software se tiene lo siguiente:

- Windows 2000
- Servidor Tomcat.
- Lenguaje de Programación JAVA ( Servlets).
- Access 2000.

Para la instalación del servidor Tomcat se tomo en cuenta los siguientes requisitos:

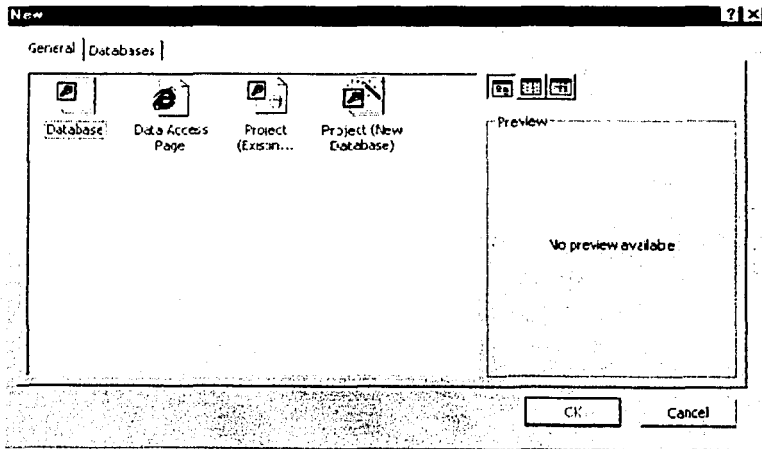
- Se ejecutan sin problemas sobre Windows 2000.
- El código binario funciona con procesadores de familia x86 y,
- Se requieren 12 MB durante la instalación y 3 MB para la instalación.

En cuanto a la Programación utilizada, en este caso JAVA, se requiere tener un claro conocimiento de los conceptos de la Programación Orientada a Objetos mencionados en el capítulo III.1 correspondiente al análisis de requerimientos.

## V.II CONSTRUCCIÓN DE LA BASE DE DATOS.

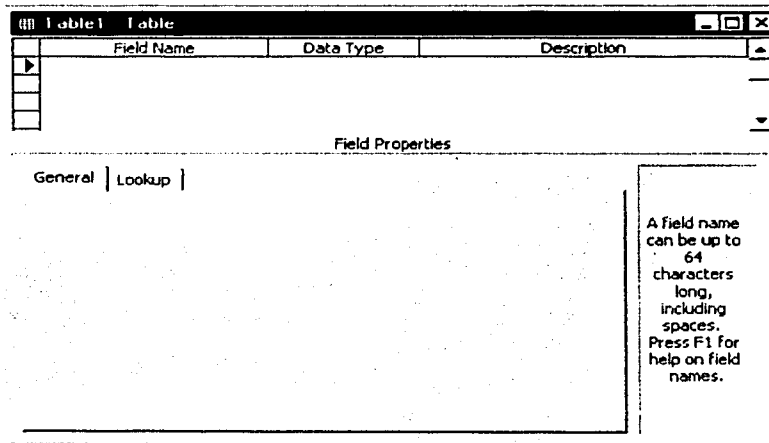
Access es una herramienta que se utilizo para la elaboración de la base de datos, se dieron de altas los siguientes tablas y atributos.

- Primero se da de alta una nueva base de datos.

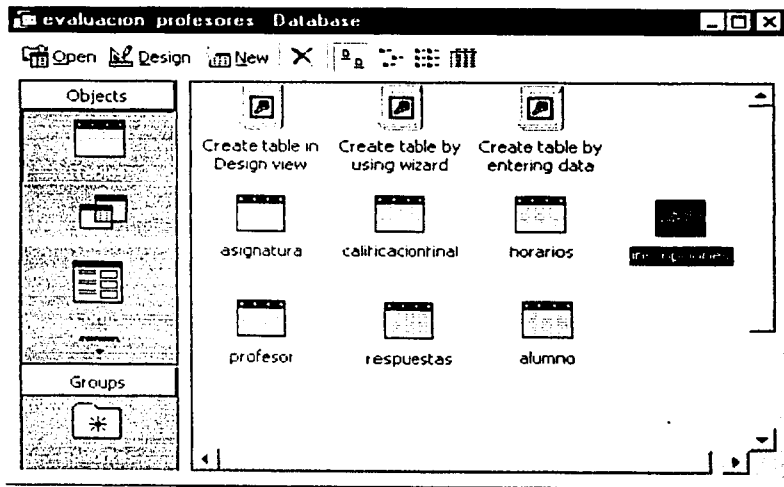


- Se guarda la nueva base de datos en la ruta que se desee direccionar.
- Posteriormente se procede a crear una tabla en vista de diseño, tomando en cuenta los conceptos de base de datos.

TESIS CON  
FALLA DE ORIGEN



- Se dan de alta las tablas que se requieren para el Sistema de Evaluación de Profesores, considerando que una tabla es un conjunto de registros que se pueden dividir en campos.



TESIS CON  
FALLA DE ORIGEN

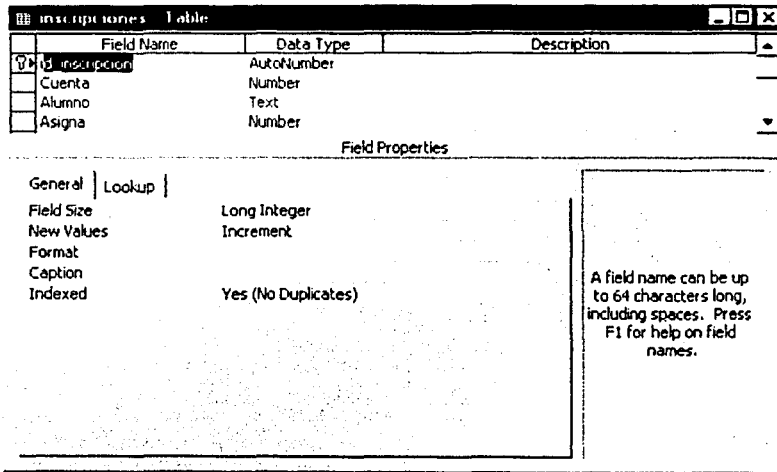
- En la vista hoja de datos se escriben los datos para cada registro, dentro de la tabla los datos se despliegan en un formato de filas y columnas, donde cada fila representa un solo registro de la tabla y cada columna representa un campo.

Id_ins	Cuenta	Alumno	Asigna	Nombre	Grupo	Profesor	respuesta
17	99507712	BAEZ RODRIGUEZ	1505	NUTRICION	1501	ANGELES C	no
12	99507774	ALVAREZ DE LA C	1605	PATOLOGIA	1603	BOUDA JAN	no
13	99507774	ALVAREZ DE LA C	1606	ALIMENTOS	1603	CERVANTE	no
15	99507712	BAEZ RODRIGUEZ	1507	FUNDAMEN	1509	HERNANDE	no
19	99349594	VICENTE MORENO	1106	HISTOLOG	1101	LAVIELLE R	ya
10	99508836	CASTELLANOS R/	1506	MEJORAMI	1509	LOPEZ ORC	no
8	99508836	CASTELLANOS R/	1608	EPIDEMIOL	1605	MARTINEZ I	no
14	99507774	ALVAREZ DE LA C	1608	EPIDEMIOL	1602	MONROY L	no
4	99509149	SOTO LOPEZ GRE	1710	MEDICINA	1705	NUÑEZ HER	no
21	99349594	VICENTE MORENO	1104	BIOLOGIA	1101	OCAMPO L	no
7	99508836	CASTELLANOS R/	1605	PATOLOGIA	1605	ORDOÑEZ E	no
1	99509149	SOTO LOPEZ GRE	1713	MANEJO DI	1705	RAMIREZ P	ya
6	99508836	CASTELLANOS R/	1606	ALIMENTOS	1605	RAMIREZ P	ya
16	99507712	BAEZ RODRIGUEZ	1406	BIOESTADI	1403	RINCON GA	no

Record: 14 of 21

TESIS CON  
FALLA DE ORIGEN

- En vista de diseño se definen las propiedades de toda la tabla y de cada campo individualmente, en las propiedades se incluyen el nombre de los campos, el tipo de datos utilizados para cada uno de ellos y los índices definidos para la tabla



TESIS CON  
FALLA DE ORIGEN

### V.III CODIFICACIÓN UTILIZANDO ACCESS Y JAVA.

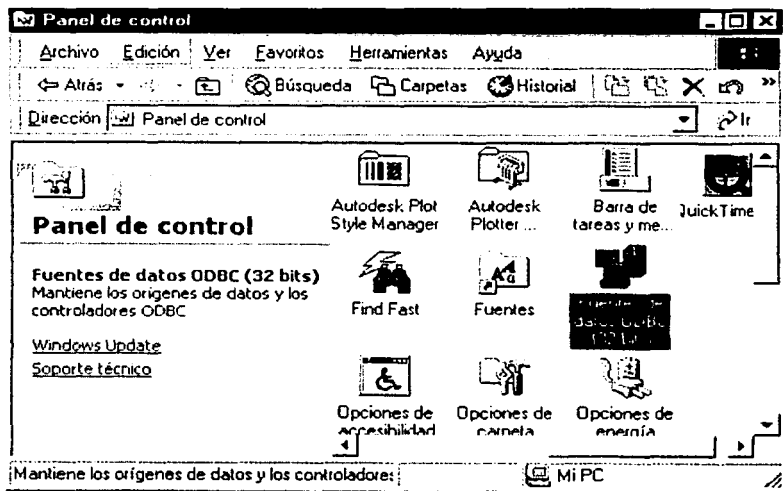
Una de las tareas más importantes y más frecuentemente realizadas por los servlets es la conexión a bases de datos mediante JDBC.

JDBC(Java Database Connectivity) es la parte de Java que se encarga de la conexión de base de datos, ya que provee las clases e interfaces necesarias para conectar los programas java a bases de datos relacionales de distintos proveedores.

Para establecer la conexión a la base de datos se utilizará el driver JDBC-ODBC, que se encargará de traducir las invocaciones JDBC a invocaciones ODBC a través de librerías ODBC del sistema operativo.

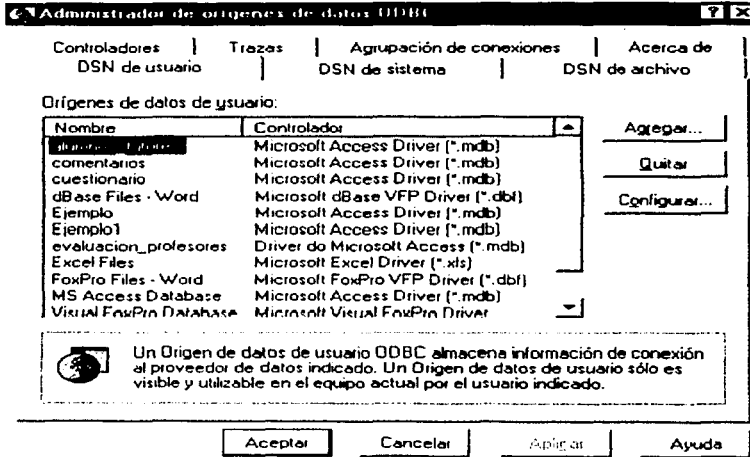
A continuación se describen los pasos para realizar este proceso.

1. En panel de control elegir Fuentes de datos.

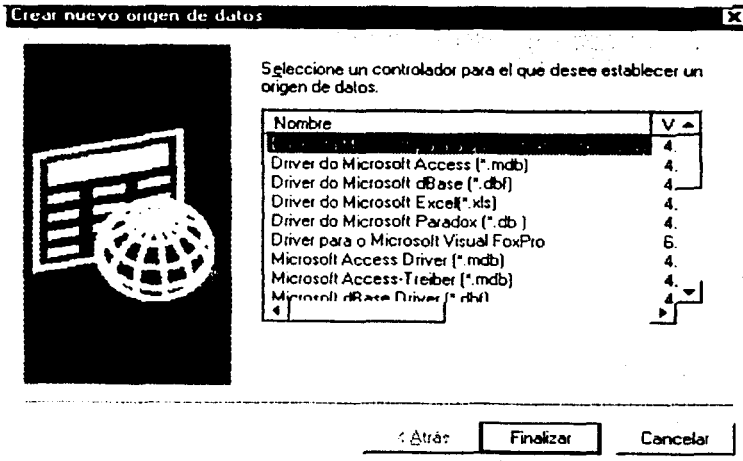


TESIS CON  
FALLA DE ORIGEN

2. En Administrador de orígenes de datos ODBC, presionamos el botón agregar.

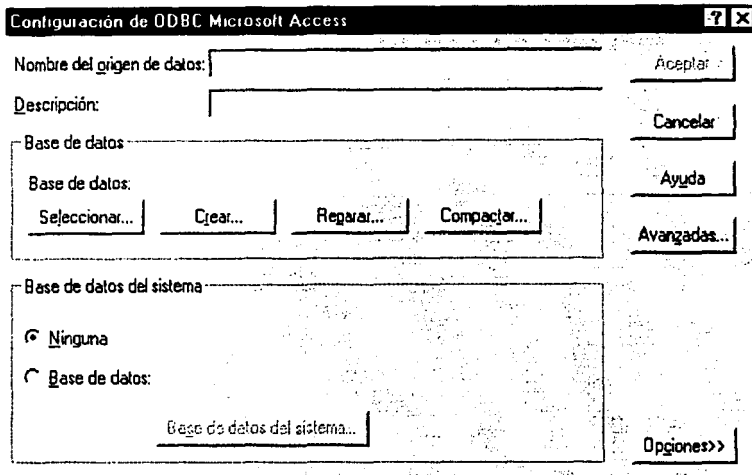


3. Seleccionamos Microsoft Access Driver (\*.mdb) y presionamos Finalizar.



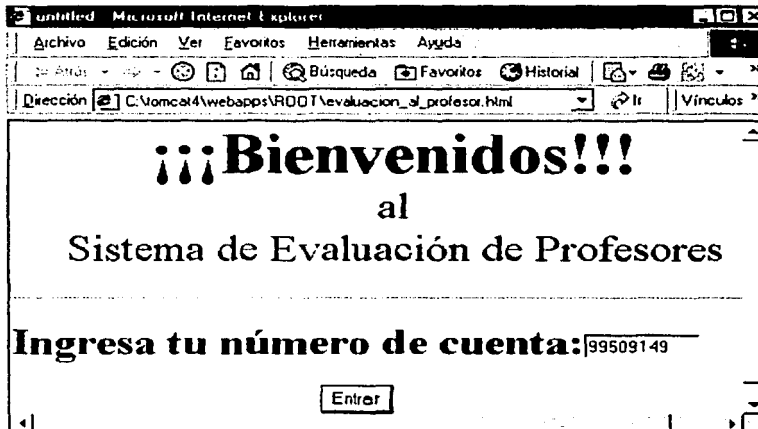
TESIS CON  
 FALLA DE ORIGEN

4. Presionamos el botón Seleccionar y elegimos nuestra base de datos.



Una vez establecida la conexión con la base de datos, se puede iniciar con la ejecución de la programación.

- Primeramente se genera solicitud del número de cuenta del alumno.



NO SE PUEDE REPRODUCIR SIN EL ORIGIN

- Para proporcionar una ejemplificación de la codificación del Sistema de Evaluación de Profesores, se muestra a continuación uno de los resultados generados por un Servlet, que consiste en proporcionar al alumno la lista de asignaturas cursadas durante el semestre, para su posterior evaluación.

The screenshot shows a Microsoft Internet Explorer browser window. The address bar contains the URL: `http://localhost:8080/examples/servlet/Checkal_alumno`. The main content of the page is as follows:

**Alumno: SOTO LOPEZ  
GRETA JESSICA**

**Estos son los profesores que debe evaluar:**

Asigna	Nombre	Grupo	Profesor	Estado
1713	MANEJO DE FORRAJES	1705	RAMIREZ PEREZ AURORA HILDA	Evaluado
	MEDICINA			

The browser's status bar at the bottom shows the current page URL: `http://localhost:8080/examples/servlet/Evaluar_prof` and the connection type: `Intranet local`.

TESIS CON FALLA DE ORIGEN

#### **V.IV SEGURIDAD.**

Las tres razones principales para utilizar la seguridad a nivel de usuario son las siguientes:

- Proteger la propiedad intelectual del código.
- Impedir que los usuarios cambien o inutilicen inadvertidamente una aplicación cambiando código de objetos de los que depende la aplicación.
- Proteger los datos sensibles de la base de datos.

##### **Seguridad de Access.**

Las versiones recientes de Microsoft Access ofrecen dos métodos para proteger una base de datos:

- Contraseña para abrir un archivo de base de datos.
- Seguridad a nivel de usuario o de cuentas.

##### **Contraseña de Archivo MDB.**

Los usuarios son obligados a identificarse y escribir una contraseña cuando inician Microsoft Access o una Aplicación Access. El método *Contraseña de Archivo MDB* es simple y consiste en habilitar una contraseña para abrir un archivo MDB específico. El método es el adecuado para una base de datos que esté compartida entre un pequeño grupo de usuarios o sobre un sólo equipo. No es aplicable si quiere replicar la BD o pretende implantar un sistema de seguridad en red.

##### **Seguridad a nivel de usuario o de cuentas.**

La seguridad se basa en *permisos*, los cuales son atributos para un grupo o usuario. Por ejemplo, a un grupo con el nombre *Usuarios\_Alpha* se les permitirá visualizar, introducir o modificar datos en una tabla *Cientes*, pero no se les permite cambiar el diseño de esa tabla, ni acceder otras tablas. Así un usuario que pertenezca a este grupo sólo tendrá estos atributos y, en particular, los que le quiera dar un Administrador.

##### **Seguridad en Java.**

El modelo de seguridad de Java se conoce como modelo de cuatro capas, este modelo se implementa mediante la construcción de cuatro barreras o líneas de defensa:

- Primera línea de defensa: Características del lenguaje/compilador.
- Segunda línea de defensa: Verificador de código de bytes.
- Tercera línea de defensa: Cargador de clases.
- Cuarta línea de defensa: Gestor de Seguridad.

### **Características del lenguaje/compilador.**

Java fue diseñado con las siguientes ideas en mente:

- Evitar errores de memoria.
- Imposibilitar acceso al SO.
- Evitar que caiga la máquina sobre la que corre.

Con el fin de llevar a la práctica estos objetivos, se implementaron las siguientes características:

- Ausencia de punteros.
- Protege frente a imitación de objetos, violación de encapsulación, acceso a áreas protegidas de memoria, ya que el programador no podrá referenciar posiciones de memoria específicas no reservadas, a diferencia de lo que se puede hacer en C y C++.
- Ya no se puede gestionar la memoria de forma tan directa como en C. En cambio, se instancian objetos, no se reserva memoria directamente, minimizando así la interacción del programador con la memoria y con el sistema operativo.
- El programador ya no libera la memoria manualmente mediante free (fuente muy común de errores en C y C++, que podía llegar a producir el agotamiento de la memoria del sistema). El recogedor de basura de Java se encarga de reclamar la memoria usada por un objeto una vez que éste ya no es accesible o desaparece. Así, al ceder parte de la gestión de memoria a Java en vez de al programador, se evitan las grietas de memoria (no reclamar espacio que ya no es usado más).
- En Java los arrays son objetos, lo cual les confiere ciertas funciones muy útiles, como la comprobación de límites. Para cada subíndice, Java comprueba si se encuentra en el rango definido según el número de elementos del array, previniendo así que se referencien elementos fuera de límite.
- Las variables y los métodos declarados privados sólo son accesibles por la clase o subclases herederas de ella y los declarados como protegidos, sólo por la clase.

### **Verificador de Códigos de Bytes.**

Sólo permite ejecutar código de bytes de programas Java válidos, buscando intentos de:

- Fabricar punteros,
- Ejecutar instrucciones en código nativo,
- Llamar a métodos con parámetros no válidos,
- Usar variables antes de inicializarlas,

El verificador efectúa cuatro pasadas sobre cada fichero de clase:

- En la primera, se valida el formato del fichero .
- En la segunda, se comprueba que no se instancien subclases de clases finales.
- En la tercera, se verifica el código de bytes: la pila, registros, argumentos de métodos.
- En la cuarta, se finaliza el proceso de verificación.

Si el verificador aprueba un fichero .class, se le supone que cumple ya con las siguientes condiciones:

- Acceso a registros y memoria válidos.
- No hay overflow o underflow de pila.
- Consistencia de tipo en parámetros y valores devueltos.

### **Cargador de Clases.**

A la hora de ejecutarse las applets en nuestra máquina, se consideran tres dominios con diferentes niveles de seguridad:

- La máquina local (el más seguro)
- La red local guardada por el firewall (seguro)
- La Internet (inseguro)

En este contexto, no se permite a una clase de un dominio de seguridad inferior sustituir a otra de un dominio superior, con el fin de evitar que una applet cargue una de sus clases para reemplazar una clase crítica del sistema.

Este tipo de ataque se imposibilita asignando un espacio de nombres distinto para clases locales y para clases cargadas de la Red. Siempre se accede antes a las clases del sistema, en vez de a clases del mismo nombre cargadas desde una applet.

### **Gestor de Seguridad.**

La gestión de seguridad la realiza la clase abstracta SecurityManager, que limita lo que las applets pueden o no hacer. Para prevenir que sea modificada por un applet maliciosa, no puede ser extendida por las applets.

Entre sus funciones de vigilancia, se encuentran el asegurar que las applets no acceden al sistema de ficheros, no abren conexiones a través de internet, no acceden al sistema, etc.

Para ello se introducen una serie de restricciones de seguridad o prohibiciones.

### **Restricciones de seguridad.**

En primer lugar, es importante hacer notar que las restricciones son distintas para cada navegador, ya que cada uno ofrece su propia política de seguridad. En cualquier caso, la mayoría de los navegadores presentarán las siguientes restricciones a la hora de ejecutar applets:

- No se puede trabajar con ficheros en la máquina del usuario a menos que éste lo permita.
- No se puede conectar a nada en la Red más que a la máquina que envió la applet.

# **CAPÍTULO VI**

## **ENFOQUE PARA LA PRUEBA DEL SOFTWARE.**

---

### **VI.1 VERIFICACIÓN Y VALIDACIÓN.**

La prueba del software a menudo se le conoce como Verificación y Validación. La verificación se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica. La validación se refiere a un conjunto de diferentes actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

La validación del software se consigue mediante una serie de pruebas que demuestran la conformidad con los requisitos. Un plan de prueba traza las clases de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

#### **Pruebas.**

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

#### **Objetivos de la prueba del software.**

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

#### **Tipos de pruebas.**

Las diversas pruebas a que debe ser sometido un sistema deben ser realizadas tanto por el equipo de desarrolladores, como por los usuarios, equipos de operación y mantenimiento en la implantación, aceptación y mantenimiento del sistema de información. Los tipos de pruebas que deben realizarse son:

- Pruebas Unitarias.
- Pruebas de Integración.
- Pruebas del Sistema. Pruebas de Implantación.

- Pruebas de Aceptación.
- Pruebas de Regresión.

### **Pruebas Unitarias.**

Las pruebas unitarias tienen como objetivo la funcionalidad y estructura de cada componente individualmente una vez que ha sido codificado.

La prueba unitaria se da por finalizada cuando se hayan realizado todas las verificaciones establecidas y no se encuentren defecto alguno.

### **Pruebas de integración.**

El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probadas unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces.

### **Pruebas del sistema.**

Las pruebas del Sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto del sistema de información con los que se comunica.

Son pruebas de integración del sistema de información completo y permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.

### **Pruebas de implantación.**

El objetivo de las pruebas de implantación es comprobar el funcionamiento correcto del sistema integrado de hardware y software en el entorno de operación y permitir al usuario que, desde el punto de vista de operación, realice la aceptación del sistema una vez instalado en su entorno real y en base al cumplimiento de los requisitos no funcionales especificados.

Una vez que hayan sido realizadas las pruebas del sistema en el entorno de desarrollo, se llevan a cabo las verificaciones necesarias para asegurar que el sistema funcionará correctamente en el entorno de operación. Debe comprobarse que responde satisfactoriamente a los requisitos de rendimiento, seguridad, operación y coexistencia con el resto de los sistemas de la instalación para conseguir la aceptación del usuario de operación.

Las pruebas de seguridad van dirigidas a verificar que los mecanismos de protección incorporados al sistema cumplan su objetivo, las de rendimiento a asegurar que el sistema responde satisfactoriamente en los márgenes establecidos en cuanto a tiempos de respuesta, de ejecución y de utilización de recursos, así como los volúmenes de espacio en disco y capacidad: por último con las pruebas de operación se comprueba que la planificación y control de trabajo del sistema se realiza de acuerdo con los procedimientos establecidos.

### **Pruebas de Aceptación.**

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Estas pruebas van dirigidas a comprobar que el sistema cumple los requerimientos de funcionalidad esperados, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información y conseguir así la aceptación final del sistema por parte del usuario.

### **Pruebas de Regresión.**

El objetivo de las pruebas de regresión es eliminar el efecto onda, es decir, comprobar que los cambios sobre un componente de un sistema de información no introducen un comportamiento no deseado o errores adicionales en otros componentes no modificados.

Las pruebas de regresión se deben llevar a cabo cada vez que se hace un cambio en el sistema, tanto para corregir un error como para realizar una mejora. No es suficiente probar que también es necesario controlar que las modificaciones no produzcan efectos negativos sobre él mismo u otros componentes.

**VI.II PRUEBA DEL SISTEMA.**

<b>OBJETIVO.</b>	<b>ACTIVIDADES.</b>
<p><b><u>Caso de uso evaluación.</u></b></p> <p>Autenticación de un usuario al sistema, considerando.</p> <ul style="list-style-type: none"> <li>• Un usuario con cuenta válida.</li> <li>• Un usuario con cuenta no válida.</li> <li>• Validación de un formulario que permita establecer el número de evaluaciones correspondientes.</li> <li>• Envío correcto de información a la base de datos.</li> </ul>	<p><b><u>Caso de uso evaluación.</u></b></p> <ul style="list-style-type: none"> <li>• Petición de la dirección de inicio al sistema a través del navegador web.</li> <li>• Introducir la cuenta del alumno.</li> <li>• Despliegue de las materias a evaluar, estableciendo un correcto enlace con las páginas o programas en cuestión.</li> <li>• En caso contrario efectuar el aviso correspondiente.</li> <li>• Verificar la autenticación de los datos (alumno-asignatura).</li> <li>• Efectuar las correspondientes ligas, comprobando su funcionalidad.</li> <li>• Ejecución correcta del programa que permita el despliegue del cuestionario, mediante un formulario.</li> <li>• Efectuando la comprobación de los datos, así como su correcto llenando, evitando duplicidad en las respuestas.</li> <li>• Ejecución del programa que realiza el envío de respuestas, generadas anteriormente con el formulario, comprobando que se realice de manera correcta la recepción de información por la base de datos.</li> </ul>

<b>OBJETIVO.</b>	<b>ACTIVIDADES.</b>
<p><b><u>Caso de uso promedios.</u></b></p> <p>Autenticación de un usuario al sistema, considerando.</p> <ul style="list-style-type: none"> <li>• Un usuario con cuenta válida.</li> <li>• Un usuario con cuenta no válida.</li> <li>• Evaluar respuestas.</li> <li>• Envío correcto de información a la base de datos.</li> </ul>	<p><b><u>Caso de uso promedios.</u></b></p> <ul style="list-style-type: none"> <li>• Efectuar las correspondientes ligas, comprobando su funcionalidad.</li> <li>• Ejecución del programa que efectúa el cálculo de promedios.</li> <li>• Comprobando la correcta conexión a la base de datos, para la introducción de los resultados obtenidos de este proceso.</li> <li>• Efectuando la comprobación de los datos, así como su correcto llenado, evitando duplicidad en los datos.</li> </ul>

<b>OBJETIVO.</b>	<b>ACTIVIDADES.</b>
<p><b><u>Caso de uso reportes.</u></b></p> <p>Autenticación de un usuario al sistema, considerando.</p> <ul style="list-style-type: none"> <li>• Un usuario con cuenta válida.</li> <li>• Un usuario con cuenta no válida.</li> <li>• Conexión a la base de datos de datos.</li> </ul>	<p><b><u>Caso de uso reportes.</u></b></p> <ul style="list-style-type: none"> <li>• Efectuar las correspondientes ligas, comprobando su funcionalidad.</li> <li>• Ejecución del programa que efectúa el cálculo de promedios y calificación final.</li> <li>• Comprobando la correcta conexión a la base de datos, para establecer una serie de consultas, que permitirán el llenado del reporte solicitado.</li> </ul>

### VI.III PRUEBA DE SEGURIDAD.

Para proporcionar una protección limitada a la base de datos del sistema, se agregó una contraseña para controlar qué usuarios podrán abrirla. Microsoft Access guarda la contraseña de la base de datos sin codificar.

El procedimiento realizado para activar la contraseña es el siguiente:

- En el menú **Archivo**, haga clic en **Abrir**.
- Haga clic en la flecha situada a la derecha del cuadro **Buscar en**, seleccione la unidad y la carpeta donde se encuentra la base de datos de Microsoft Access (.MDB) y luego haga clic en el icono de la base de datos.
- Haga clic en la flecha situada a la derecha del botón **Abrir** y, a continuación, haga clic en **Abrir en modo exclusivo**.
- En el menú **Herramientas**, señale **Seguridad** y después haga clic en **Establecer contraseña para la base de datos**.
- En el cuadro **Contraseña**, introduzca su contraseña.
- En el cuadro **Confirmar contraseña**, confirme su contraseña escribiendo de nuevo la contraseña y haciendo después clic en **Aceptar**.
- La contraseña está ahora habilitada. La próxima vez que se abra la base de datos, aparecerá un cuadro de diálogo pidiendo la contraseña.

# **CAPÍTULO VII**

## **MANTENIMIENTO Y DOCUMENTACIÓN.**

---

### **MANTENIMIENTO Y DOCUMENTACIÓN.**

#### **MANTENIMIENTO DEL SOFTWARE.**

La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software, y a cambios debidos a las mejoras por los requisitos cambiantes del cliente.

#### **TIPOS DE MANTENIMIENTO.**

##### **Mantenimiento Correctivo.**

- Involucra el diagnóstico y corrección de uno o más errores. Ocurren errores en el sistema que el usuario reporta al desarrollador.
- Una vez instalados los sistemas, la necesidad de corregir errores de emergencia es relativamente baja.

##### **Mantenimiento Adaptativo.**

- El mantenimiento adaptativo modifica el sistema para adaptarlo a los cambios del medio ambiente (entorno original).
- Este mantenimiento se debe a los cambios rápidos que ocurren en cualquier aspecto computacional. Por ejemplo, nuevas versiones del software de desarrollo, de sistemas operativos, de equipo periférico, etc.
- El mantenimiento de los sistemas también implica adaptaciones de versiones anteriores, adecuar cambios de reportes, archivos y procesos.

##### **Mantenimiento Perfectivo (Mejoras).**

- Conforme el sistema es usado, nuevas recomendaciones para satisfacer nuevos requerimientos, modificaciones a las funciones existentes, y mejoras en general son recibidas por parte de los usuarios.
- La mayor parte del tiempo de mantenimiento se convierte en perfectivo conforme pasa la vida del sistema.
- Este tipo de mantenimiento trata de cumplir las nuevas peticiones de los usuarios, mejorar la eficiencia (por ejemplo tiempo de respuesta).

### **Mantenimiento Preventivo (Reingeniería del Software).**

- El software se deteriora debido al cambio, por ello deben efectuarse cambios en el software a fin de que se pueda corregir, adaptar y mejorar más fácilmente (facilidad de mantenimiento). A este proceso se le llama reingeniería del software.
- Este tipo de mantenimiento usualmente es sugerido por el mismo departamento de desarrollo.

A continuación se muestran las siguientes pantallas que ejemplifican el mantenimiento a realizar con la base de datos, para el Sistema de Control de Evaluación de Docencia y Desempeño Académico por Internet.

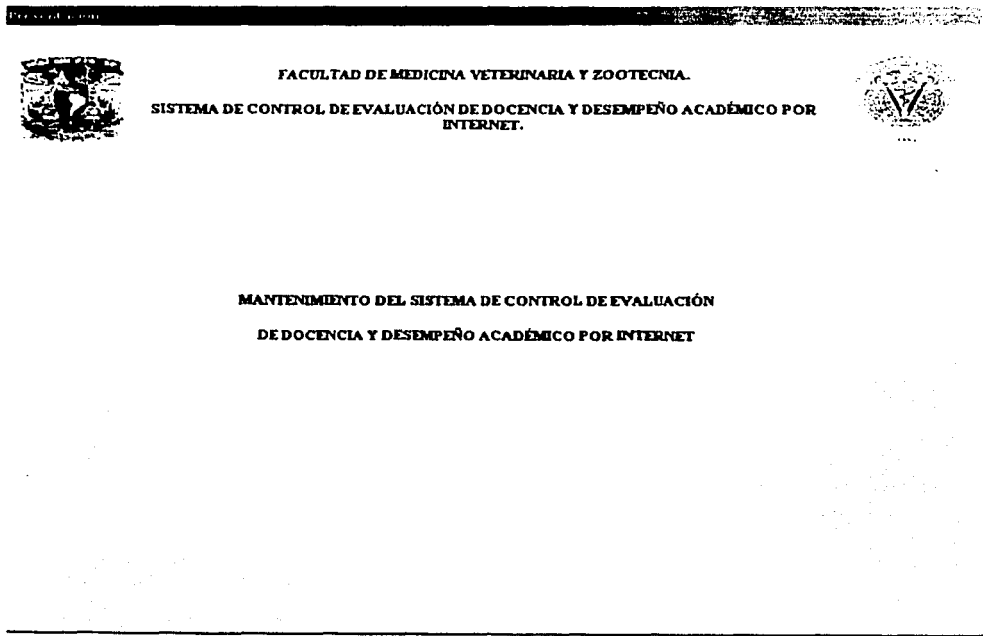


Figura 7.1 Pantalla de presentación.

TESIS CON  
FALLA DE ORIGEN

El usuario el permiso solicitado podrá elegir la base de datos a respaldar.

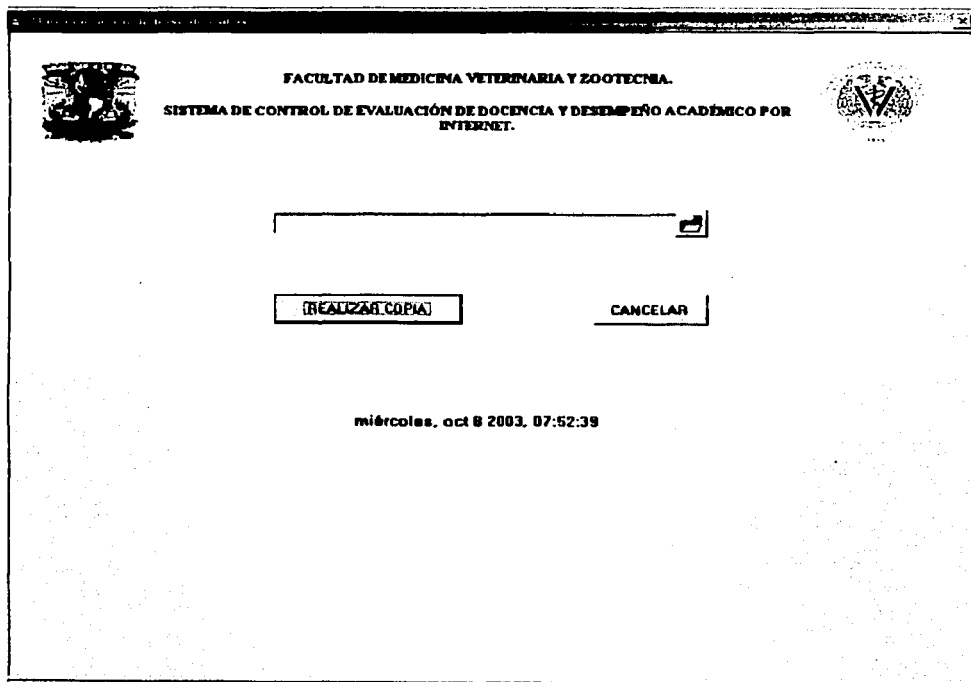


Figura 7.2 Pantalla de selección.

TESIS CON  
FALLA DE ORIGEN

Una vez encontrada la ubicación del archivo de la base de datos nos muestra la siguiente pantalla.

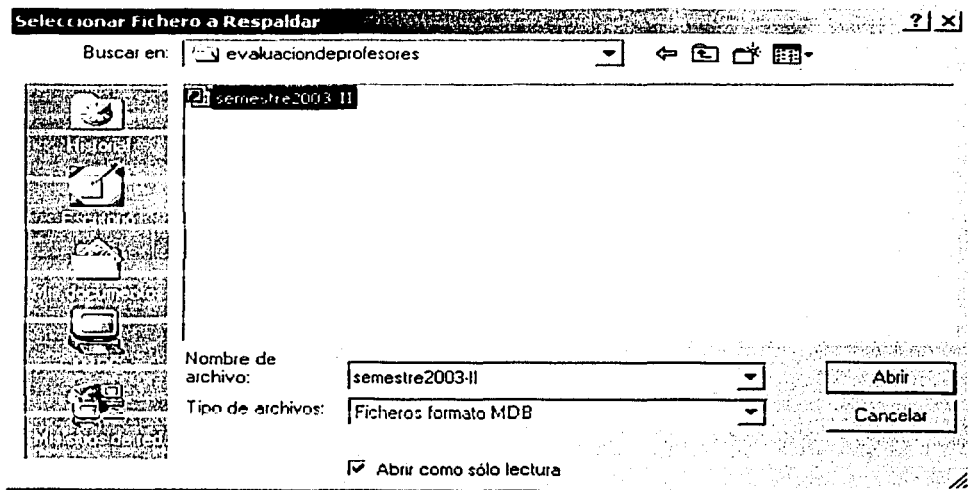


Figura 7.3 Pantalla de búsqueda.

TESIS CON  
FALLA DE ORIGEN

Cuando ya fue seleccionado el archivo aparece en la ventana como se muestra a continuación.

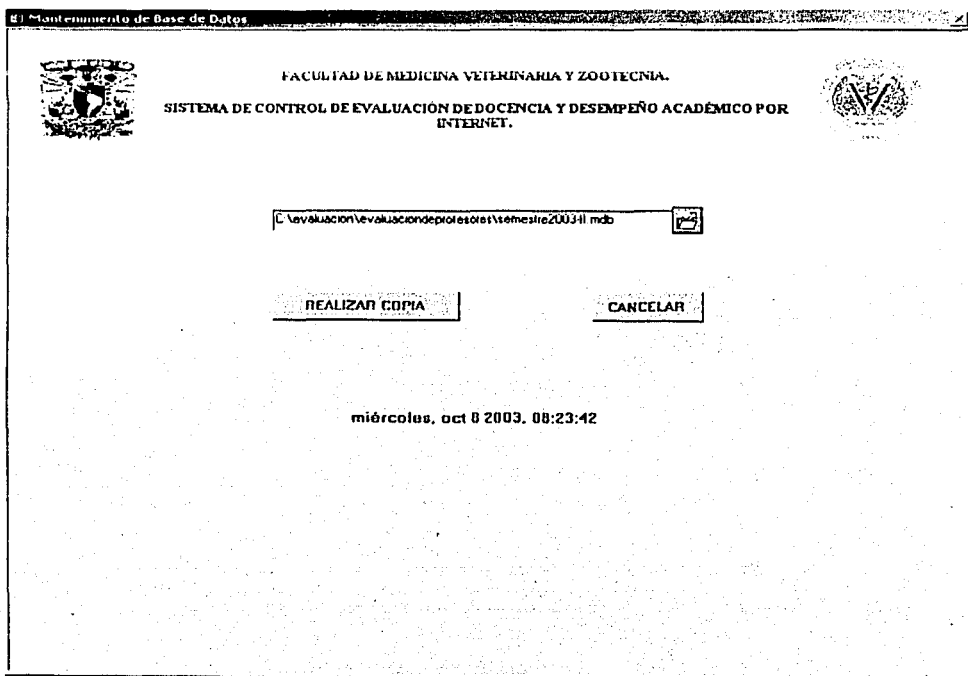


Figura 7.4 Pantalla de verificación.

TESIS CON  
FALLA DE ORIGEN

Una vez que el usuario ha verificado y seleccionado el archivo de base de datos a respaldar, se le muestra una mensajería para indicarle que el proceso ha concluido satisfactoriamente.

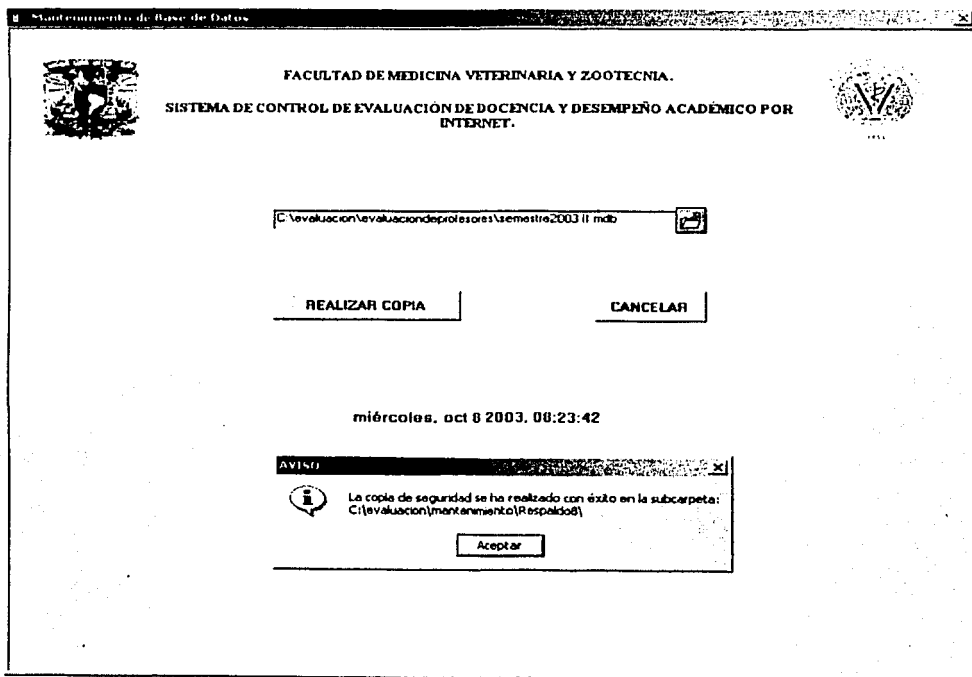


Figura 7.5 Pantalla de respaldo.

TESIS CON  
FALLA DE ORIGEN

Finalmente el usuario puede comprobar que el respaldo de su archivo efectivamente ha sido realizada.

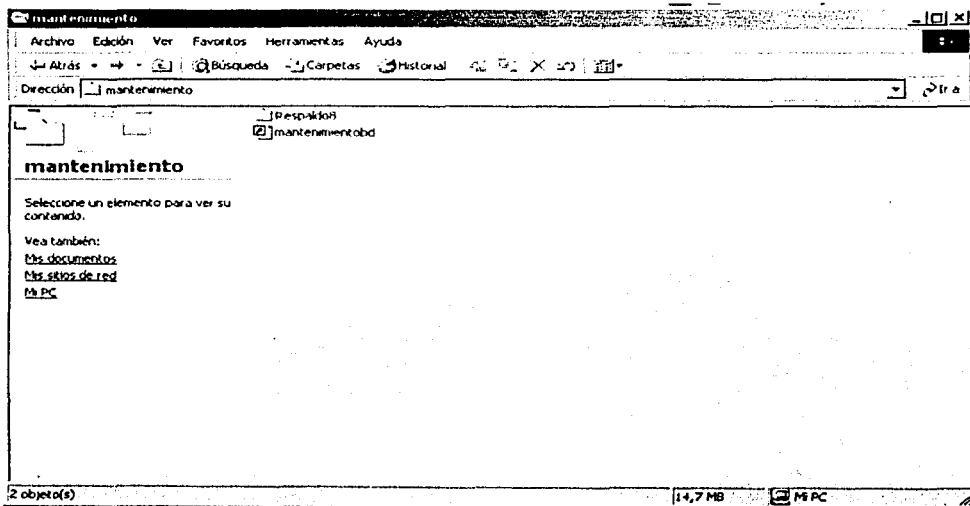


Figura 7.6 Pantalla mantenimiento.

TESIS CON  
FALLA DE ORIGEN

Como podemos observar el programa que permite realizar el respaldo de la base de datos se encuentra en la carpeta mantenimiento, cuando deseamos realizar una copia de nuestro archivo de base de datos activamos el programa y una vez seleccionado el archivo, nuestra copia quedara en una carpeta llamada Respaldo como se muestra en la siguiente pantalla.

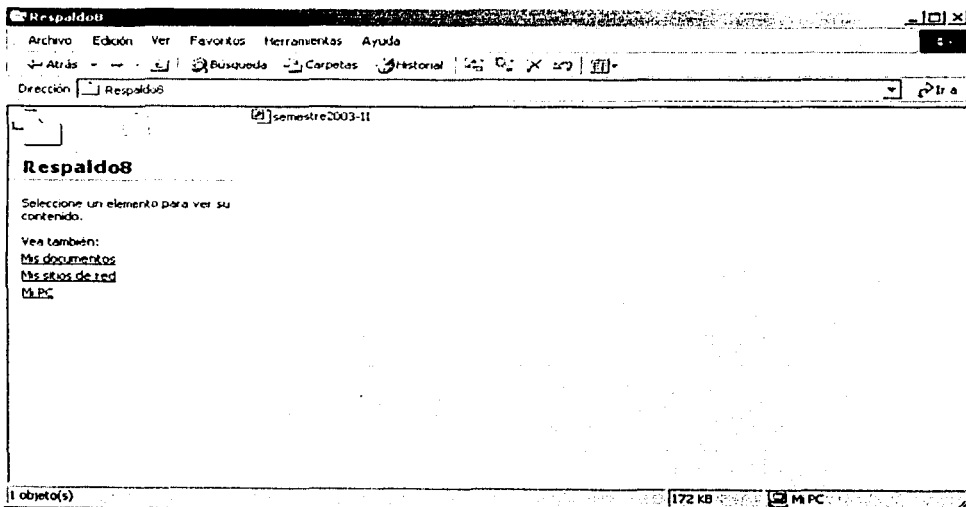


Figura 7.7 Pantalla respaldo.

## DOCUMENTACIÓN DEL SOFTWARE.

- La documentación contendrá como instalar el sistema.
- Una descripción funcional de lo que puede hacer el sistema.
- Un manual que explique de manera sencilla como opera el sistema y,
- Documentación del código fuente.

TESIS CON  
FALLA DE ORIGEN

## MANUAL DEL USUARIO.

A continuación se muestra en la figura 7.8, la interfaz gráfica que se le presenta al usuario cuando accede al sistema, para ello, el usuario deberá conectarse desde su navegador web a la dirección de la página de inicio del sistema, <http://profesores.ods.org/evaluacion> y posteriormente seleccionar el botón correspondiente, ya sea para comenzar con el proceso de evaluación, la creación de un reporte, el cálculo de promedios o bien consultar el manual del usuario.

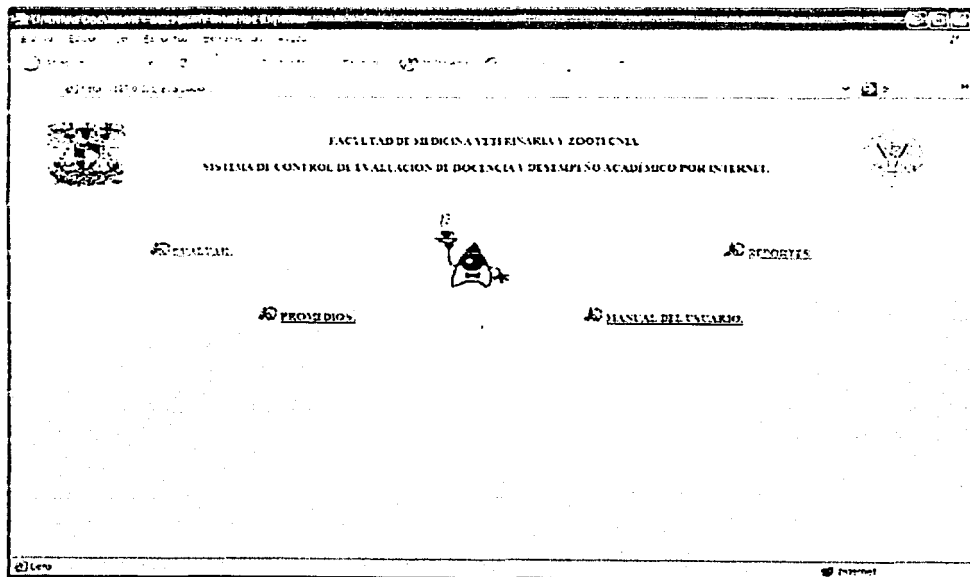


Figura 7.8 Página de inicio del Sistema de Control de Evaluación.

TESIS CON  
FALLA DE ORIGEN

La figura 7.9 muestra la interfaz gráfica del sistema después de haber seleccionado la opción Evaluar. El objetivo de esta interfaz gráfica, es solicitar la cuenta del usuario, para comenzar con el proceso de autenticación.

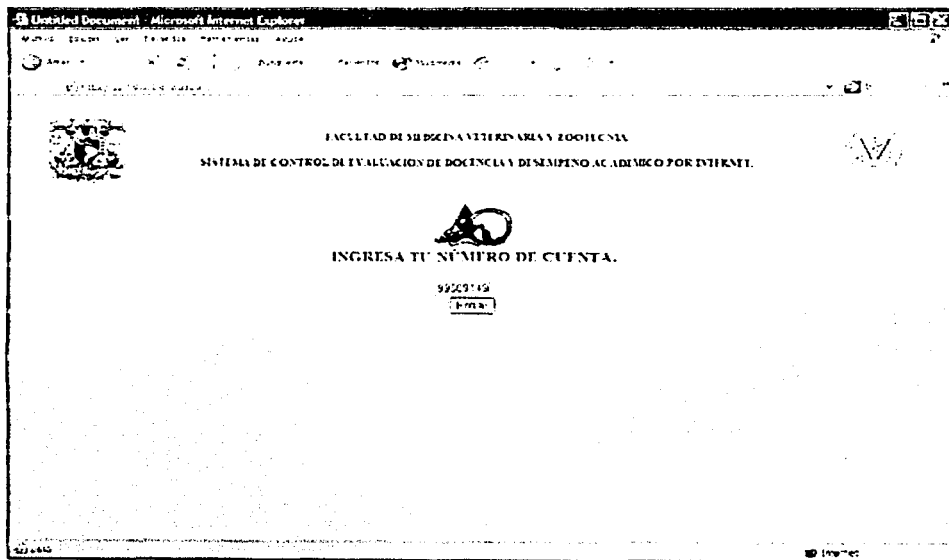


Figura 7.9 Página de autenticación del usuario.

TESIS CON  
FALLA DE ORIGEN

Si la cuenta del usuario resulta válida, comenzará con el proceso de evaluación seleccionando el id\_inscripción correspondiente al profesor, asignatura y grupo deseado, como se muestra en la figura 7.10.

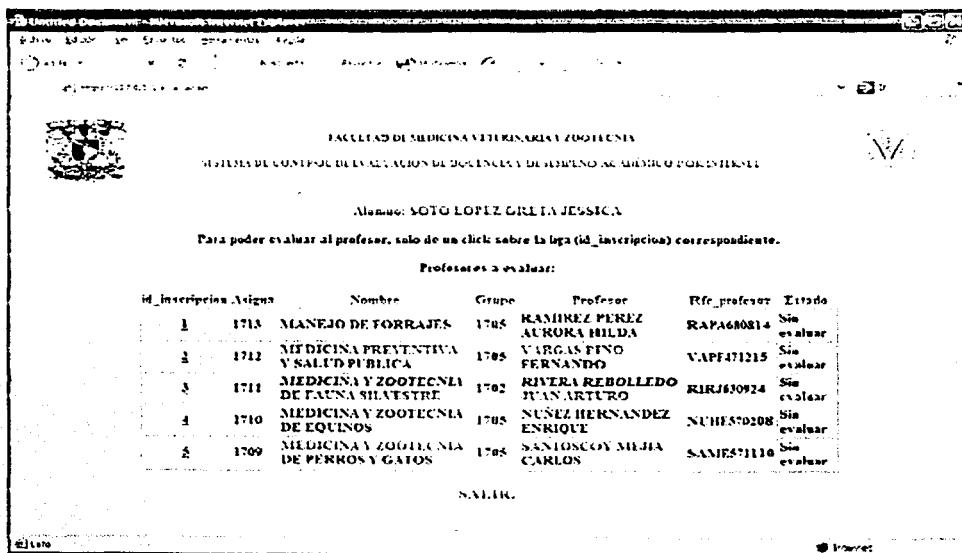


Figura 7.10 Página de evaluación del profesor.

TESIS CON  
FALLA DE ORIGEN

Una vez seleccionada la liga correspondiente se desplegará el cuestionario, como en la figura 7.11 para ser contestado por el usuario.

Unfilled Document Microsoft Internet Explorer  
Archivo Editar Ver Herramientas Ayuda  
Inicio Favoritos Historial Configuración de Internet  
http://127.0.0.1/...  
FACULTAD DE MEDICINA VETERINARIA Y ZOOTECNIA  
INSTITUTO DE CONTROL DE EVALUACION DE DOCENCIA Y DESEMPEÑO ACADÉMICO POR INTERNET  
Nombre: MANEJO DE FORAJEROS Profesor: RAMIREZ PIREZ AYRORA MELBA  
Asigna: 1713 Grupo: 1702  
CUESTIONARIO.  
1. Los conocimientos que tiene el profesor sobre su asignatura son:  
 Excelentes  Buenos  Regulares  Deficientes  No sé  
2. El profesor relaciona los temas de su curso con otros temas de la carrera.  
 Siempre  Frecuentemente  Pocas veces  Nunca  No sé  
3. El profesor compartió en clase experiencias de su vida profesional para dar a usted una visión más amplia de la carrera.  
Internet

Figura 7.11 Página de cuestionario.

TESIS CON  
FALLA DE ORIGEN

Cuando todas las preguntas se hayan contestado, el usuario enviará sus respuestas, para ser almacenadas en la base de datos.

Si el proceso se realizó correctamente aparecerá la siguiente pantalla de la figura 7.12.

En caso contrario el usuario tendrá que regresar con el navegador a revisar que haya contestado todas las preguntas.

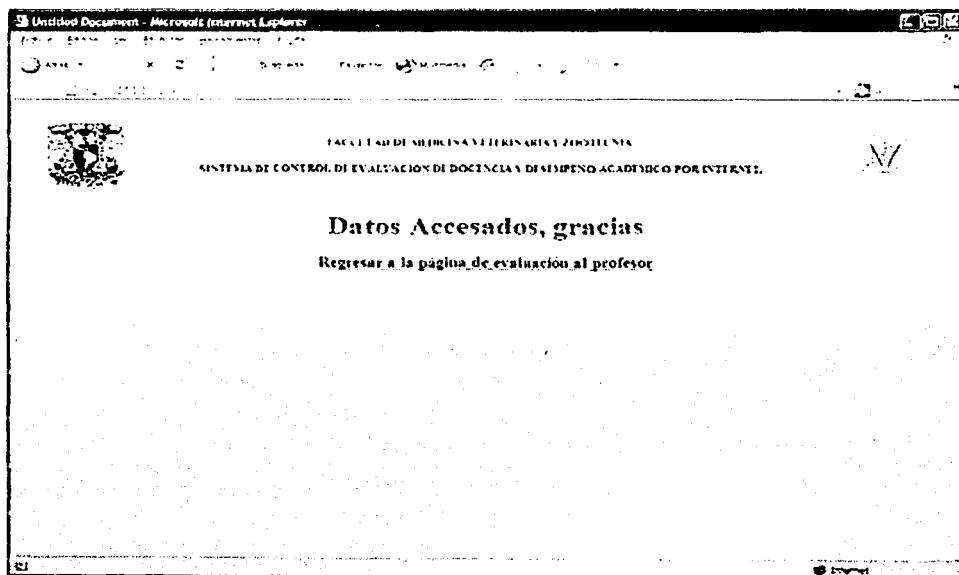


Figura 7.12 Página de acceso de datos.

TESIS CON  
FALLA DE ORIGEN

En la figura 7.12 mostrada anteriormente, donde nos muestra que los datos fueron accedidos correctamente, también nos permite regresar a la página de evaluación, con la finalidad de poder tener la opción de evaluar a otro profesor si así se desea.

En esta figura 7.13 observamos que la id\_inscripción que ya fue evaluada es desactivada y nos muestra su estado.

Facultad de Medicina Veterinaria y Zootecnia  
 Sistema de Control de Evaluación de Docencia y Desempeño Académico por Internet

Alumno: SOTO LOPEZ GRETA JESSICA

Para poder evaluar al profesor, sólo de un click sobre la liga (id\_inscripción) correspondiente.

Profesores a evaluar:

id_inscripción	Asigna	Nombre	Grupo	Profesor	Rfc_profesor	Estado
1	1713	MANEJO DE FORRAJES	1705	RAMIREZ PEREZ ACROBA HILDA	RAP1630214	Evaluado
2	1712	MEDICINA PREVENTIVA Y SALUD PUBLICA	1704	VARGAS PINO FERNANDO	VAPF471215	Sin evaluar
3	1711	MEDICINA Y ZOOTECNIA DE FAUNA SILVESTRE	1702	RIVERA REBOLLEDO JUAN ARTURO	RIRJ639924	Sin evaluar
4	1710	MEDICINA Y ZOOTECNIA DE EQUINOS	1705	MUNIZ HERNANDEZ ENRIQUE	MUHES*0208	Sin evaluar
5	1709	MEDICINA Y ZOOTECNIA DE PERROS Y GATOS	1705	SANTOS COYMEJIA CARLOS	SAMTS71110	Sin evaluar

SALIR.

Figura 7.13 Página de evaluación.

TESIS CON  
FALLA DE ORIGEN

Una segunda opción a elegir es la realización de cálculo de promedios, para ello el usuario tendrá que proporcionar su clave, como se muestra en la figura 7.14.

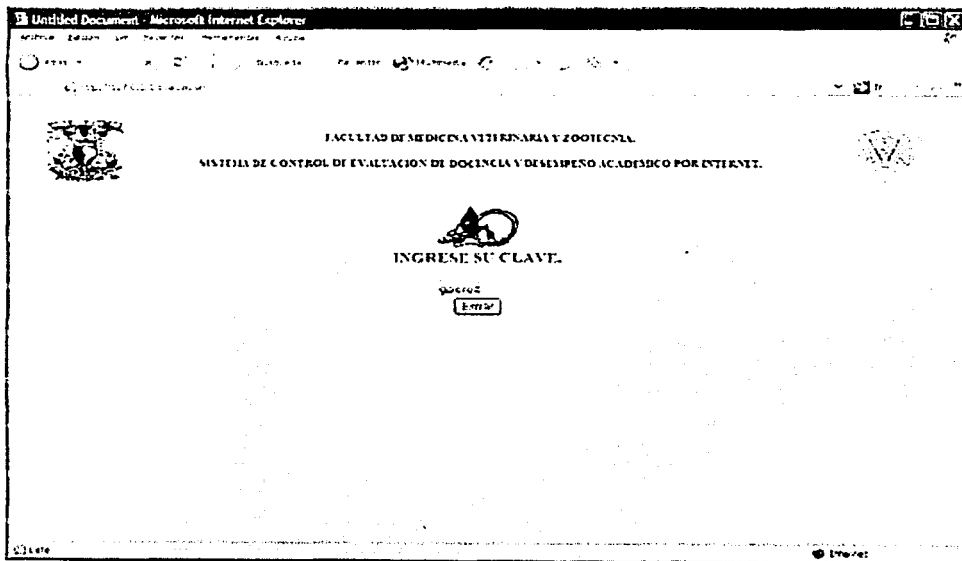


Figura 7.14 Validación del usuario.

TESIS CON  
FALLA DE ORIGEN

Posteriormente aparece una pantalla como en las figuras 7.15a y 7.15b para confirmar la realización del cálculo.

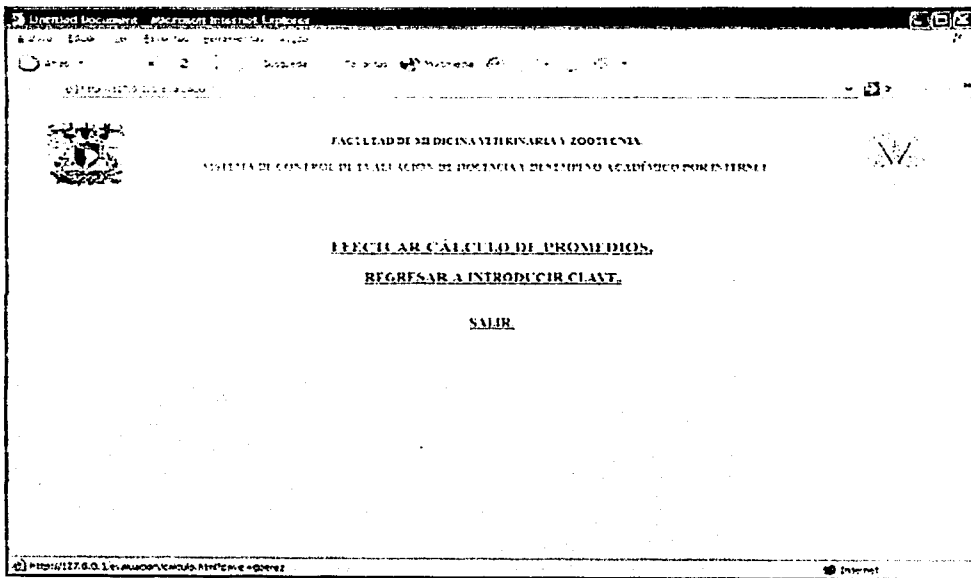


Figura 7.15a Página de cálculo.

TESIS CON  
FALLA DE ORIGEN

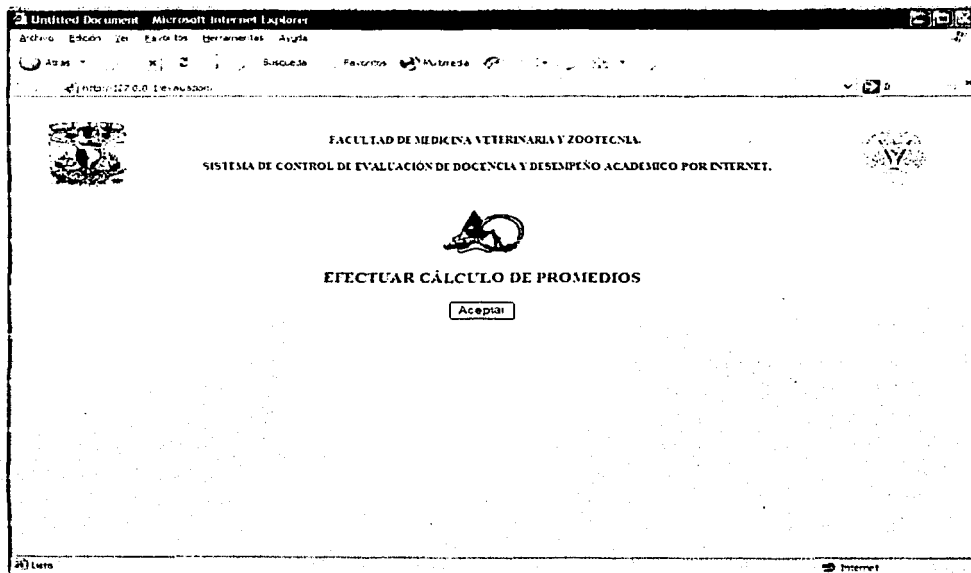


Figura 7.15b Página de cálculo.

TESIS CON  
FALLA DE ORIGEN

**FALTA**  
**PAGINA**  
**130**

Una tercera opción es la realización de reportes orientado a dos tipos de usuarios, al profesor o bien al administrador.

En la figura 7.17 aparecen dos tipos de ligas uno de reporte general utilizado por el administrador, y el reporte individual por el profesor.

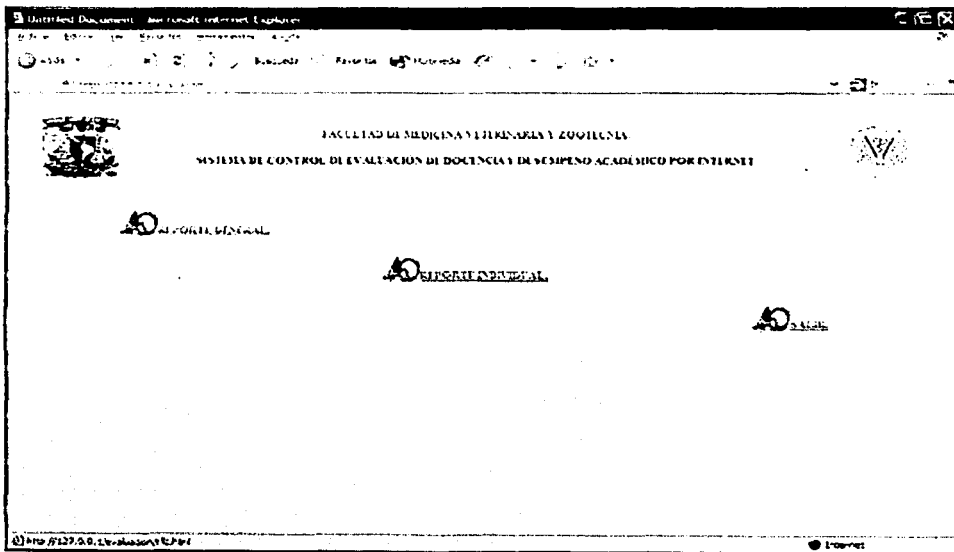


Figura 7.17 Selección de reportes.

TEXTO CON  
FALLA DE ORIGEN

Si el usuario selecciona la opción reporte general tendrá que proporcionar su clave de acceso, figura 7.18.

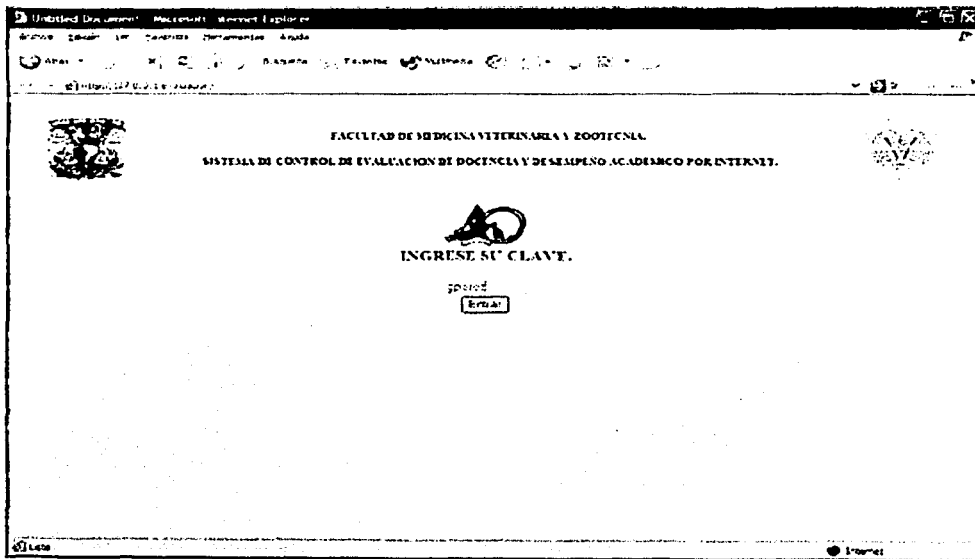


Figura 7.18 Acceso a reporte.

TESIS CON  
FALLA DE ORIGEN

La elaboración del reporte puede realizarse en formato HTML y en documento de Word, como en la figura 7.19.

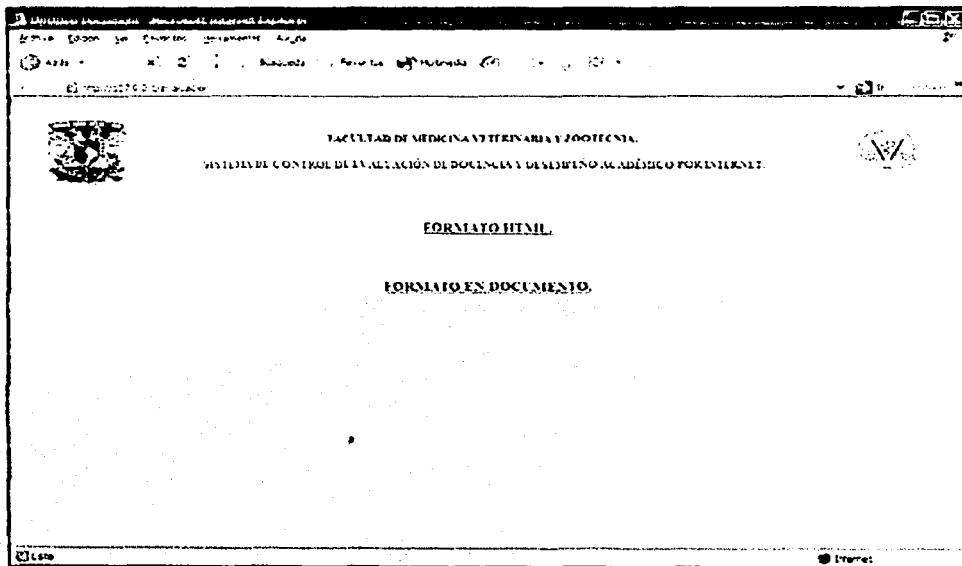


Figura 7.19 Formato.

TESIS CON  
FALLA DE ORIGEN

Si seleccionamos formato HTML aparece la pantalla de la figura 7.20, en ella se muestra una lista de profesores a elegir.

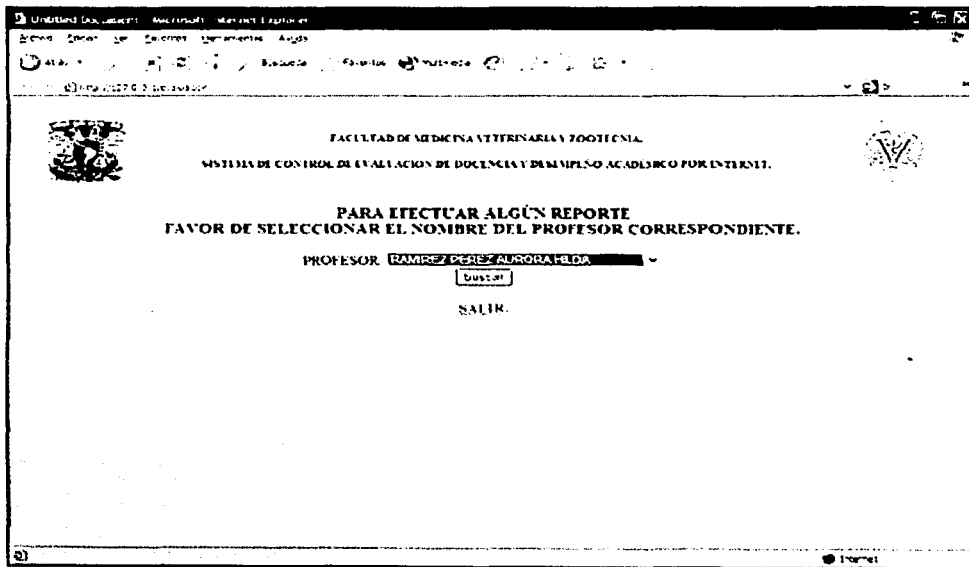


Figura 7.20 Página de selección de profesor.

TESIS CON  
FALLA DE ORIGEN

Al seleccionar al profesor nos mostrará la siguiente pantalla figura 7.21, para elegir el identificador correspondiente a la materia grupo y profesor deseado.

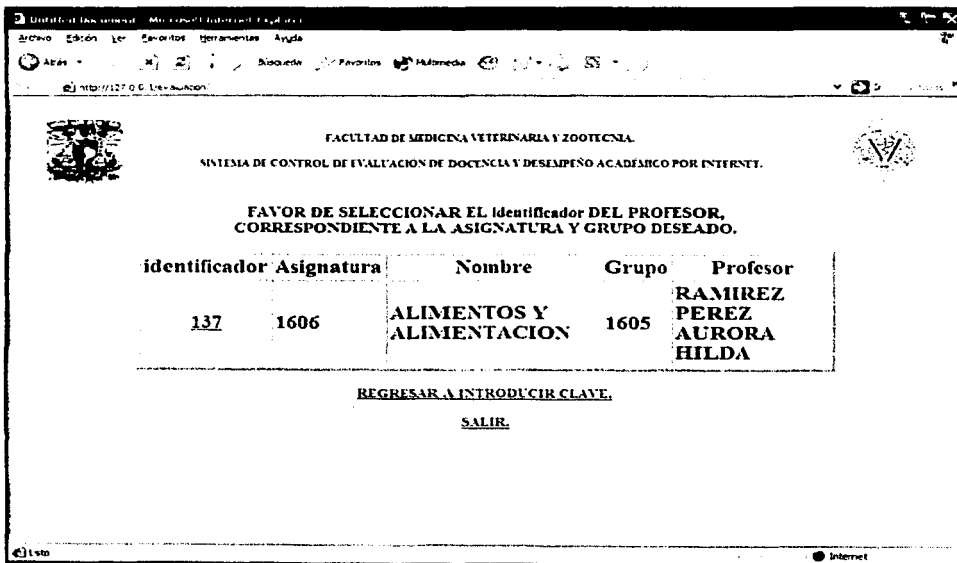


Figura 7.21 Selección identificador.

TESIS CON  
FALLA DE ORIGEN

Finalmente al elegir el identificador nos muestra el reporte en formato HTML como en la siguiente figura 7.22

Unidad de Planeación - Med. y Zootecnia - Reporte

Activos Datos de Evaluación Informes de Asesores

AYUDANTE DE PROFESORES

Facultad de Medicina Veterinaria y Zootecnia

SISTEMA DE CONTROL DE EVALUACIÓN DE DOCENCIA Y DESEMPEÑO ACADÉMICO POR INTERNET

UNIDAD DE PLANEACIÓN

EVALUACIÓN DE PROFESORES REALIZADA POR LOS ALUMNOS EN TIEMPO LIBRE

NOMBRE DEL PROFESOR: RAMÍREZ PEREZ AURORA WILDA

ASIGNATURA: GRUPO: 10406

CONOCIMIENTO: Es sobre los conocimientos que el profesor tiene de los temas que enseña el curso.

COMUNICACIÓN: Es un concepto sobre la capacidad del maestro para exponer con claridad su materia, si la presentación de su clase es organizada y lógica, si menciona la atención a los intereses del alumno, si sus clases no son monótonas, si invita al alumno a pensar y si expone claramente sus objetivos.

EXPERIENCIA PROFESIONAL: Es la capacidad del profesor para permitir la relación del conocimiento de la materia con la vida diaria, ejemplos, experiencias o sereno de la actualidad.

IMPARTICIÓN: De acuerdo con el espacio en los cuatrimestres, los alumnos evalúan positivamente al profesor docente que trata de orientar a sus alumnos. Esto es lo posible que el alumno cree por su cuenta.

SERIEDAD: Evalúa la capacidad del profesor de establecer en forma personal la orientación, comunicación y asesoría a los alumnos.

JUSTICIA E IMPARCIALIDAD: Es un imparcial, observa cómo evalúan los alumnos su relación personal con el grupo. El profesor tiene un criterio de calificación el cual respeta.

ARTAS	CALIFICACIÓN OBTENIDA	PROMEDIO DE LA FACULTAD	MAXIMO QUT PUEDE OBTENER
COMUNICACION	22.3	111.3	36.6
CONOCIMIENTOS	10.56	66.99	20.61
EXPERIENCIA PROFESIONAL	9.46	48.4	11.36

TESIS CON FALLA DE ORIGEN

Figura 7.22 Reporte formato html.

La opción de reporte individual está enfocada para el profesor, el cual deberá introducir su rfc como se muestra en la figura 7.23 para poder tener acceso a sus reportes.

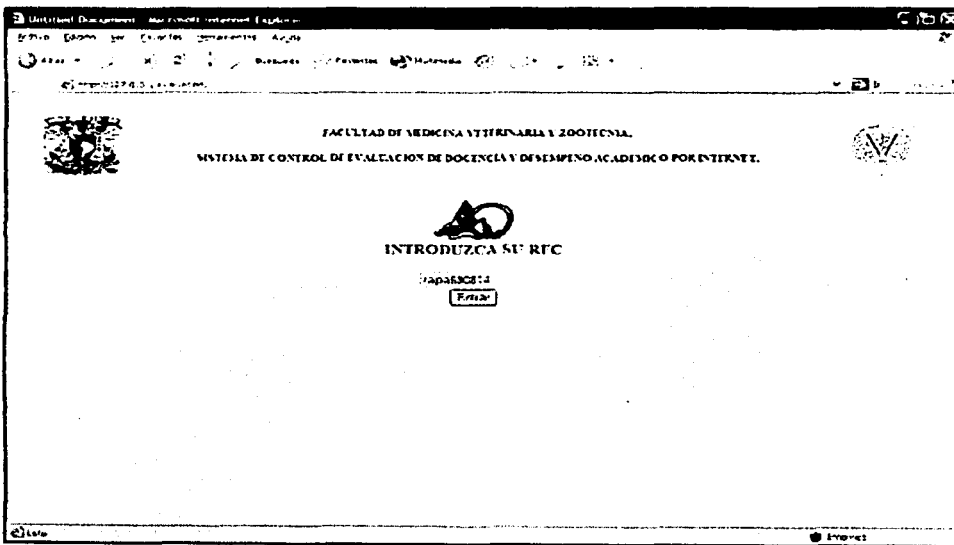


Figura 7.23 Pantalla de acceso rfc.

TESIS CON  
FALLA DE ORIGEN

Una vez que el sistema valida el rfc del profesor, le mostraran una lista de todas las materias impartidas, para seleccionar la deseada como en la figura 7.24.

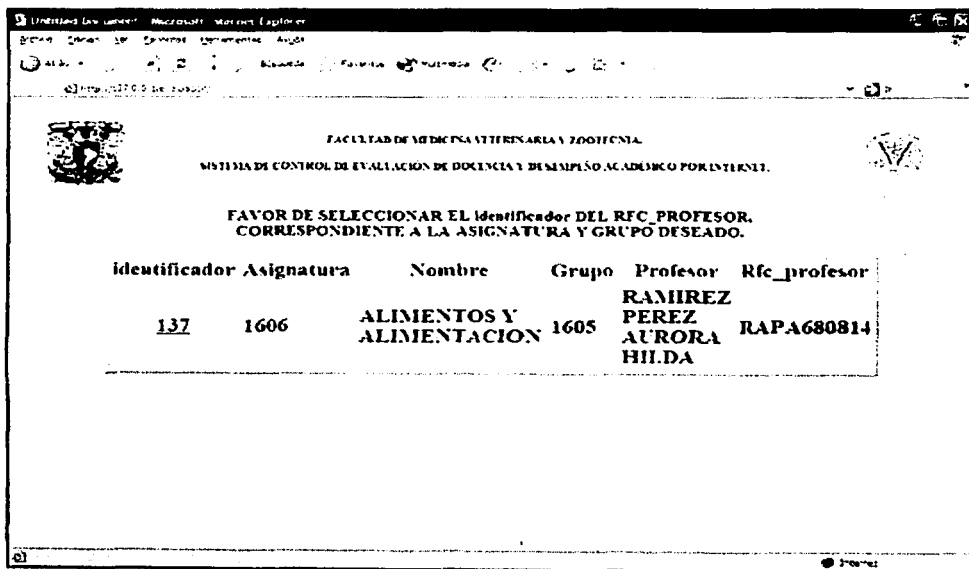


Figura 7.24 Pantalla de selección de identificador.

TESIS CON  
FALLA DE ORIGEN

Finalmente se muestra el reporte deseado como en la figura 7.25

UNIDAD DE PLANEACIÓN.  
EVALUACIÓN DE PROFESORES REALIZADA POR LOS ALUMNOS EN EL SEMESTRE

NOBRE DEL PROFESOR: RAMÍREZ PÉREZ AURORA HILDA.  
ASIGNATURA: ALIMENTOS Y ALIMENTACIÓN GRUPO: 1605

**CONOCIMIENTOS.** Evalúa los conocimientos que el profesor tiene de los temas que consta el curso.  
**COMUNICACIÓN.** Evalúa la capacidad del docente para exponer con claridad su materia, si la presentación de su clase es organizada y lógica, si mantiene la atención a través del alumno, si sus temas no son monótonos, si ayuda al alumno a estudiar y si explica claramente sus asignaciones.  
**EXPERIENCIA PROFESIONAL.** Evalúa la capacidad del profesor para permitir la relación del conocimiento de la materia con la vida diaria, ejemplos, experiencias u otro tipo de actividades.  
**MOTIVACIÓN.** De acuerdo con lo expuesto en los cuestionarios, los alumnos evalúan positivamente al profesor durante que trata de orientarles a sus alumnos. Trata en lo posible que el alumno estudie por sí mismo.  
**ACCESIBILIDAD.** Evalúa la capacidad del profesor de establecer en forma personal la orientación, comunicación e atención a los alumnos.  
**JUSTICIA E IMPARCIALIDAD.** Trata imparcial, sin tener como evaluación como relación personal con el grupo. El profesor debe un criterio de calificación el cual respeta.

AREA	CALIFICACION OBTENIDA	PROMEDIO DE LA FACULTAD	MAXIMO QUE PUEDE OBTENER
COMUNICACION	22.3	111.0	28.6
CONOCIMIENTOS	10.56	59.29	28.44
EXPERIENCIA PROFESIONAL	0.46	46.4	11.54

Figura 7.25 Pantalla de selección de identificador.

TRABAJOS CON  
FALLA DE ORIGEN

## CONCLUSIONES.

---

El problema que se planteo al inicio del presente trabajo, ha sido resuelto con el desarrollo de un sistema de software. Proporcionando así una solución óptima a la labor tediosa, de gran pérdida de tiempo y demasiada lentitud en su proceso.

La construcción de esta aplicación satisface las necesidades del usuario en cuanto a utilidad, eliminando la problemática ocasionada al utilizar un sistema que ofrece las siguientes desventajas.

- Acceso local.
- Disponibilidad.
- Distribución de información.
- Capacitación.

Ya no será necesario recolectar todas las bases de datos de los diferentes departamentos de la Facultad de Medicina Veterinaria y Zootecnia, así como efectuar la compresión del programa que realiza todo este proceso, para posteriormente cargarlo en cada una de las máquinas utilizadas en las evaluaciones, y finalmente volver a recolectar ahora los resultados obtenidos de las evaluaciones, para realizar el análisis que nos llevará a la obtención de promedios, y la generación de reportes.

La forma como se resolvió este problema, fue realizando un estudio de las necesidades del usuario, así como la tecnología requerida.

En la etapa de diseño se utilizó la notación gráfica del Lenguaje de Modelado Unificado (UML), para describir el funcionamiento del sistema a través de sus diagramas de casos de uso, clases y secuencia.

Para la programación se utilizó Java Servlets encargados de establecer la conexión con la base de datos o bien con otro servlet para efectuar operaciones de inserción, selección y eliminación de información, en el caso de la base de datos. La comunicación con otro servlet permite enviar las asignaturas a evaluar, para posteriormente enviar el cuestionario, o bien efectuar los cálculos para la obtención de promedios.

Mediante la herramienta HTML construimos la presentación de la aplicación, así como la utilización de formularios para la recepción de información. El motor utilizado para la ejecución de los Servlets fue el servidor tomcat.

En cuanto a la configuración de la aplicación se utilizó el archivo de configuración web.xml, instalado dentro del subdirectorio Web-inf del contexto configurado en Tomcat, con el fin de establecer los parámetros necesarios para su funcionamiento.

En la etapa de pruebas se diseñó un conjunto de pruebas para comprobar la confiabilidad del sistema y garantizar la calidad del software, durante estas pruebas se pudieron detectar varios errores que fueron corregidos sistemáticamente.

## **BIBLIOGRAFÍA.**

---

Análisis y diseño orientado a objetos con Aplicaciones.

Grady Booch.

2ª. Edición Addison-Wesley/Diaz de Santos.

Análisis y diseño orientado a objetos.

James Martín, James J. Odell.

Prentice Hall.

El lenguaje unificado de modelado:Manual de referencia.

James Rumbaugh, Ivar Jacobson, Grady Booch

UML y patrones: Introducción y diseño orientado a objetos.

Graig Larman.

Prentice Hall 2ª. Edición.

Ingeniería del software.

Un enfoque práctico.

Pressman Roger S.

Tercera Edición.

Mc Graw-Hill.

Fundamentos y modelos de Bases de Datos.

De Miguel, A. y Piattini, M.

Conceptos y diseño de bases de datos relacionales del Modelo E/R al modelo relacional.

De Miguel, A. y Piattini, M.

Redes de computadores, protocolos, normas e interfaces

Uyless Black.

BIBLIOGRAFÍA.

---

**Cómo programar en Java.**  
**Deitel y Deitel**  
**PrenticeHall**

**Java 2 Manual de usuario y tutorial.**  
**Agustín Froufe.**  
**Alfaomega.**  
**2da. Edición.**