



**BENEMÉRITA UNIVERSIDAD  
AUTÓNOMA DE PUEBLA**  
Facultad de Ciencias de la Computación



FACULTAD de CIENCIAS  
de la COMPUTACIÓN

**“Sistema de Información del  
Área de Software de Base”**

*Tesis*  
*que para obtener el Título de Licenciado*  
*en Ciencias de la Computación*

Presenta:  
José Alejandro Díaz Martínez

Asesor:  
M.C. Pedro Bello López

Abril 2013

# AGRADECIMIENTOS

## **A MI FAMILIA**

### **MIS PADRES EMMA Y ALEJANDRO, MI HERMANA GABY**

Por su incondicional e interminable apoyo.

Por su amor, tiempo y comprensión.

Este logro también es de ustedes.

## **A MI ASESOR**

### **M.C. PEDRO BELLO LÓPEZ**

Por su invaluable colaboración y consejos, por ser guía en este proyecto.

Gracias por todo su tiempo.

## **A MIS SINODALES Y DEMÁS MAESTROS DE LA UNIVERSIDAD**

Por brindarme todas las herramientas y medios para obtener el mejor de los tesoros: el conocimiento.

Gracias por compartir su sabiduría.

## **A MI UNIVERSIDAD**

# ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I. MARCO DE REFERENCIA.....	2
<b>1.1 Ingeniería de Software</b> .....	2
<b>1.2 Base de Datos</b> .....	6
<b>1.3 Herramientas de Representación</b> .....	8
<b>1.4 Herramientas de Desarrollo</b> .....	10
CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA.....	14
<b>2.1 Planteamiento del Problema</b> .....	14
<b>2.2 Modelo de Casos de Uso</b> .....	16
<b>2.2.1 Diagrama de Casos de Uso</b> .....	16
<b>2.2.2 Documentación y Flujo de Eventos</b> .....	17
<b>2.3 Glosario de Términos</b> .....	33
<b>2.4 Modelo Conceptual</b> .....	34
<b>2.4.1 Definición de Entidades</b> .....	34
<b>2.4.2 Identificación de Relaciones</b> .....	34
<b>2.4.2 Diagrama de Clases Simple</b> .....	35
<b>2.5 Modelo de Análisis</b> .....	36
<b>2.5.1 Diagramas de Secuencia</b> .....	36
<b>2.5.2 Diagramas de Actividad</b> .....	38
<b>2.6 Análisis y Diseño de la Base de Datos</b> .....	41
<b>2.6.1 Diagrama Entidad Relación</b> .....	41
<b>2.6.2 Esquema Relacional</b> .....	42
CAPÍTULO III. IMPLEMENTACIÓN Y PRUEBAS.....	43
<b>3.1 Alumno</b> .....	43
<b>3.1.1 Bienvenida y Menús</b> .....	43
<b>3.1.2 Consultas</b> .....	44
<b>3.1.3 Comentarios</b> .....	46

3.1.4 Validación e Ingreso al Sistema .....	47
3.2 Profesor.....	47
3.2.1 Avisos .....	48
3.2.2 Recursos Didácticos (Material Docente).....	50
3.2.3 Materias .....	53
3.3 Coordinador .....	55
3.3.1 Actas de Reuniones .....	55
3.3.2 Profesores .....	56
CONCLUSIÓN Y TRABAJO A FUTURO .....	60
BIBLIOGRAFÍA .....	61

# INTRODUCCIÓN

Para la Facultad de Ciencias de la Computación, es importante contar con un sistema que permita a los alumnos y profesores conocer el trabajo que se realiza en las diversas áreas de conocimiento dentro de la Facultad de Ciencias de la Computación y en particular en el área de software de base, por tanto, se plantea un sistema de información que muestre las asignaturas que corresponden al área para el mejor aprovechamiento de los alumnos.

Comencemos por definir el concepto de sistema de información y su objetivo.



*Un sistema de información reúne, almacena, procesa y distribuye conjuntos de información entre los diferentes elementos que configuran una organización, y entre la organización misma y su entorno[1].*

La principal finalidad es la de mantener vinculada a la comunidad estudiantil y al personal docente de esta unidad académica, de tal forma que se tenga un medio de comunicación y de compartición de información que permita un mejor organización y distribución de la misma, esto para un mejor aprovechamiento de los alumnos. De igual manera, los profesores podrán estar enterados de las actividades y las reuniones de esta área.

En el Capítulo I se especifica el marco de referencia donde se habla de los conocimientos y herramientas básicos en los cuales fue apoyado el desarrollo de la presente aplicación. En el Capítulo II se realiza el análisis y diseño del sistema para ir detallando cada parte de la aplicación utilizando diferentes tipos de diagramas y representación de la información.

La solución presentada es a través de una página web en la cual los alumnos y personas en general podrán expresar sus opiniones, comentarios, dudas o sugerencias hacia los usuarios del sistema, así como también realizar consultas del contenido que los profesores hagan disponible. Los profesores asignados al área de software de base podrán notificar avisos, subir recursos didácticos y notificar las materias que están impartiendo. Por último se determina un usuario administrador llamado coordinador (coordinador del área) cuyas opciones, adicionales a las de un profesor, serán las de dar de alta a los profesores, y subir documentos de las reuniones que se llevan a cabo en el área de software de base.

En el Capítulo III se detallan las particularidades correspondientes a la parte de implementación y pruebas, por último, las conclusiones obtenidas en el desarrollo del sistema.

# CAPÍTULO I. MARCO DE REFERENCIA

## 1.1 Ingeniería de Software

La ingeniería de software es el proceso de construir aplicaciones de tamaño o alcance prácticos, en las que predomina el esfuerzo del software y que satisfacen los requerimientos de funcionalidad y desempeño.

Desde la concepción del desarrollo de software han existido varias corrientes de mejoramiento de esta disciplina. La programación estructurada, que ahora muchos programadores toman como un hecho, usan sentencias, ramificaciones e iteraciones. Para que las partes de este tipo de programas tengan un tamaño aceptable, se introducen funciones. Estos principios de la programación estructurada son tan útiles hoy como antes, y el desarrollo de software no hubiera avanzado sin ellos. Sin embargo, se han encontrado nuevas perspectivas y paradigmas útiles que se fundamentan en la programación estructurada. Uno de los problemas más grandes con programación estructurada es que no está diseñada para hacer corresponder las entidades del programa con las entidades del mundo real, lo que dificulta manejarla y adaptarla cuando los requerimientos cambian.

En tiempos más recientes, la orientación a objetos (OO), el uso de objetos con datos y funcionalidad, ha demostrado ser una forma de pensamiento muy efectiva. La OO es un medio de expresión e implementación en lugar de una arquitectura o un diseño. La OO es efectiva porque los objetos pueden representar las partes del mundo real y las componentes del software[2].

Para darnos una idea más general acerca de los diferentes modelos de la ingeniería de software, a continuación se muestra un resumen de éstos con sus principales características[3]:

- *El modelo lineal secuencial.* Llamado algunas veces 'ciclo de vida básico' o 'modelo en cascada', sugiere un enfoque sistemático, secuencial, para el desarrollo de software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.
  - Ventajas.
    - Uno de los modelos más antiguos y más extensamente utilizado en la ingeniería de software.
    - Sirve de fundamento para procedimientos más complejos.
    - Por su simplicidad es fácil de entender y de llevar a la práctica.
  - Desventajas.
    - Los proyectos reales raramente siguen el concepto secuencial que propone el modelo.
    - A menudo es difícil que el cliente exponga explícitamente todos los requerimientos. El modelo secuencial lo requiere y tiene dificultades a la hora de acomodar la incertidumbre natural al comienzo de muchos proyectos.

- Una versión de trabajo (*'Beta'*) no estará disponible hasta que el proyecto esté muy avanzado. Esto puede causar que un error sea desastroso al detectarse hasta la revisión del programa.
- *Construcción de prototipos*. El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un "diseño rápido". El diseño rápido se centra en esos aspectos del software que se verán visibles para el usuario/cliente (por ejemplo: enfoques de entrada y formatos de salida). El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.
  - Ventajas
    - Permite definir de manera más precisa los requerimientos del sistema. Esto llevará a una solución más apegada a las necesidades del cliente y elevar el porcentaje de funcionalidad.
    - Cuando se llega a una participación intensa entre cliente y desarrollador, los procesos iterativos suelen reducir, los primeros prototipos se acercan demasiado a lo requerido en el sistema y esto permitiría hacer una entrega del sistema en un menor tiempo.
    - Ayuda a la corrección oportuna de errores, replanteamiento de funciones específicas y la reutilización de módulos ya funcionales.
  - Desventajas
    - En algunos de los casos el prototipo al no cumplir completamente con lo planteado en sus funciones, se tiene que volver a hacer completo, es decir, empezar desde cero, habiendo que inclusive desecharlo.
    - Con lo rápido que suelen ser las liberaciones de prototipos, a veces no se tiene en cuenta la calidad del software global o la facilidad de mantenimiento a largo plazo.
    - De igual manera, el desarrollador, al tener que cumplir con tiempos cortos de entrega de prototipos, puede utilizar un sistema operativo o lenguaje de programación inadecuado simplemente porque está disponible y porque es conocido; un algoritmo eficiente se puede implementar simplemente para demostrar la capacidad. Después de algún tiempo, el desarrollador debe familiarizarse con estas selecciones, y olvidarse de las razones por las que son inadecuadas. La selección menos ideal ahora es parte integral del sistema.
- *Modelo DRA*. El Desarrollo Rápido de Aplicaciones (DRA) es un modelo de proceso del desarrollo de software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una adaptación a alta velocidad del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una

construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo

- Ventajas
  - Por la definición de sus fases, suele tener un mejor funcionamiento en sistemas de información.
  - El proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario)
  - Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.
  - Si una aplicación puede modularse de tal forma que permita completarse cada una de las funciones principales en menos de tres meses, es un candidato DRA. Cada una de las funciones pueden ser afrontadas por un equipo DRA separado y ser entregadas en un solo conjunto.
- Desventajas
  - Para proyectos grandes aunque por escalas, el DRA requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.
  - DRA requiere cliente y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay compromiso por ninguna de las partes constituyentes, los proyectos DRA fracasarán.
  - DRA no es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el software nuevo requiere un alto grado de interoperatividad con programas de computadora ya existentes.
- *Modelos evolutivos de proceso de software.* Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software.
  - *El modelo incremental.* Combina elementos del modelo lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos. El modelo incremental aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software. El desarrollo incremental es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en la fecha límite que se haya establecido para el proyecto. Los primeros incrementos se pueden implementar con menos personas.
  - *El modelo espiral.* Reconoce la necesidad de pasar por la secuencia de análisis de requisitos, diseño, implementación y pruebas más de una vez. Conjuga la naturaleza iterativa de construcción de prototipos con los

aspectos controlados y sistemáticos del modelo lineal secuencial. Proporciona el potencial para el desarrollo rápido de versiones incrementales de software. Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

- *Modelo espiral WINWIN (Victoria&Victoria)*. Define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral. Más que una simple actividad de comunicación con el cliente, se definen las siguientes actividades:
  - Identificación del sistema o subsistemas clave de los directivos.
  - Determinación de las condiciones de victoria.
  - Negociación de las condiciones de victoria de los directivos para reunirlos en un conjunto de condiciones victoria-victoria para todos los afectados (incluyendo el equipo de proyecto de software).

Además del énfasis realizado en la negociación inicial, el modelo en espiral WINWIN introduce tres hitos en el proceso, llamados puntos de fijación, que ayudan a establecer la completitud de un ciclo alrededor de la espiral y proporcionan hitos de decisión antes de continuar el proyecto de software.

- *Modelo de desarrollo concurrente*. Define una serie de acontecimientos que dispararán transiciones de estado a estado para cada una de las actividades de la ingeniería de software. Por ejemplo, durante las primeras etapas de diseño, no se contempla una inconsistencia del modelo de análisis. Esto genera la corrección del modelo de análisis de sucesos, que dispara la actividad de análisis del estado **hecho** al estado cambios en **espera**.

El modelo de proceso concurrente se utiliza a menudo con el paradigma de desarrollo de aplicaciones cliente/servidor. Un sistema cliente/servidor se compone de un conjunto de componentes funcionales. Cuando se aplica cliente/servidor, el modelo de proceso concurrente define actividades en dos dimensiones: una dimensión de sistemas y una división de componentes. Los aspectos de nivel de sistema se afrontan mediante tres actividades: diseño, ensamblaje y uso.

- *Proceso unificado de desarrollo de software (PUDS)*. Este proceso se deriva de metodologías anteriores desarrolladas por Jacobson, Booch y Rumbaugh[5]. Debido a que los enfoques iterativos repiten todas las partes del proceso en cascada, puede ser complicado describirlos. El PUDS intenta resolver este problema con una clasificación de iteraciones: clasificación de concepción que comprende principalmente análisis de requerimientos; clasificación de elaboración que establece la meta técnica clave de seleccionar y confirmar una arquitectura; la iteración de construcción establece el producto básico, que todavía requiere

trabajo para liberarlo; y la iteración de transición es preparar la aplicación para liberarla al cliente.

Debido a las características del enfoque orientado a objetos y las ventajas que ofrece al brindar un conjunto organizado de iteraciones, la metodología que se seguirá en el presente proyecto será la de PUDS. Con ésta se hará la especificación de los requerimientos ya que las etapas iterativas e incrementales que plantea son fácilmente de seguir e implementar, así como de detectar errores y corregirlos tempranamente a bajo costo.

En base al PUDS se contemplan los siguientes pasos:

- 1.- Planeación del Proyecto de Desarrollo de SW.
- 2.- Análisis de Requerimientos.
- 3.- Diseño Arquitectónico o Estructural.
- 4.- Diseño Funcional o de Comportamiento.
- 5.- Implementación o Codificación.
- 6.- Pruebas.
- 7.- Mantenimiento.

## 1.2 Base de Datos

Una base de datos es una colección de información que existe durante un periodo largo, a menudo, de muchos años[7]. En el lenguaje coloquial, con la expresión base de datos se designa una colección de datos que es administrada por un sistema de administración de bases de datos, que se abrevia DBMS (Data Base Management System). Se espera que este sistema:

1. Permita a los usuarios crear otras bases de datos y especificar su esquema (estructura lógica de datos) por medio de un lenguaje especializado denominado lenguaje de definición de datos.
2. Ofrezca a los usuarios la capacidad de consultar los datos (una “consulta” es un tecnicismo de base de datos que formula una pregunta sobre los datos) y modificarlos, usando para ello un lenguaje apropiado, llamado a menudo lenguaje de consulta o lenguaje de manipulación de datos.
3. Soporte el almacenamiento de cantidades muy voluminosas de datos durante un largo periodo, protegiéndolos contra accidentes o utilización no autorizada y permitiendo el acceso eficiente para hacer consultas y modificar la base de datos.
4. Controle el acceso simultáneo a los datos por parte de muchos usuarios, sin permitir que las acciones de uno de ellos afecte a los otros ni que los accesos simultáneos corrompan los datos accidentalmente.

Bajo la estructura de la base de datos se encuentra el *modelo de bases de datos*: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Para ilustrar el concepto de un modelo de datos, describiremos dos modelos de datos: el modelo entidad-relación y el modelo relacional. Los

diferentes modelos de datos que se han propuesto se clasifican en tres grupos diferentes: modelos lógicos basados en objetos, modelos lógicos basados en registros y modelos físicos. [8]

*Modelo entidad-relación.* EL modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados **entidades**, y de **relaciones** entre estos objetos. Una entidad es una cosa u objeto en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades.

Las **entidades** se describen en una base de datos mediante un conjunto de **atributos**. Por ejemplo, los atributos *número-cuenta* y *saldo* describen la cuenta particular de un banco y pueden ser atributos del conjunto de entidades *cuenta*.

Una **relación** es una asociación entre varias entidades. Por ejemplo, una relación *impositor* asocia un cliente con cada cuenta que tiene. El conjunto de todas las entidades del mismo tipo, y el conjunto de todas las relaciones del mismo tipo, se denominan respectivamente **conjunto de entidades** y **conjunto de relaciones**.

La estructura general de una base de datos se puede expresar gráficamente mediante un *diagrama* E-R, que consta de los siguientes componentes:

- Rectángulos, que representa conjuntos de entidades.
- Elipses, que representan atributos.
- Rombo, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones.

Cada componente se etiqueta con la entidad o relación que representa.

Además de entidades y relaciones, el modelo E-R representa ciertas restricciones que los contenidos de la base de datos deben cumplir. Una restricción importante es la *correspondencia de cardinalidades*, que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones. Por ejemplo, si cada cuenta puede pertenecer solo a un cliente, el modelo puede expresar esa restricción.

*Modelo relacional.* En el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único.

El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la base de datos se estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro.

No es difícil ver cómo se pueden almacenar las tablas en archivos. Por ejemplo, un carácter especial (como una coma) se puede usar para delimitar los diferentes atributos de un registro, y otro carácter especial (como un carácter de nueva línea) se puede usar para

delimitar registros. El modelo relacional oculta tales detalles de implementación de bajo nivel a los desarrolladores de bases de datos y usuarios.

El modelo de datos relacional es el modelo de datos más ampliamente usado, y una amplia mayoría de sistemas de bases de datos actuales se basan en el modelo relacional.

El modelo relacional se encuentra a un nivel de abstracción inferior al modelo de datos E-R. Los diseños de bases de datos a menudo se realizan en el modelo E-R, y después se traducen al modelo relacional. Por ejemplo, es fácil ver que las tablas *cliente* y *cuenta* corresponden a los conjuntos de entidades del mismo nombre, mientras que la tabla *impositor* corresponde al conjunto de relaciones *impositor*.

*Modelo de datos orientado a objetos.* Es otro modelo de datos que está recibiendo una atención creciente. El modelo orientado a objetos se puede observar como una extensión del modelo E-R con las nociones de encapsulación, métodos (funciones) e identidad de objeto.

*Modelo de datos relacional orientado a objetos.* Combina las características del modelo de datos orientado a objetos y el modelo de datos relacional.

*Modelos de datos semiestructurados.* Permiten la especificación de datos donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos. El **lenguaje de marcas extensible (XML, eXtensible Markup Language)** se usa ampliamente para representar datos semiestructurados.

Históricamente, otros dos modelos de datos, el *modelo de datos de red* y el *modelo de datos jerárquico*, precedieron al modelo de datos relacional. Estos modelos estuvieron ligados fuertemente a la implementación subyacente y complicaban la tarea del modelado de datos. Como resultado se usan muy poco actualmente, excepto en el código de bases de datos antiguo que aún está en servicio en algunos lugares.

El modelo utilizado es el Relacional[6], ya que para los datos que se manejan el orden en que se manejan no es de relevancia, además de que al plantearse el problema, se observa que los datos y las entidades tienen una dependencia clara viéndose de manera general como colección de relaciones.

### 1.3 Herramientas de Representación

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés)[4] es un estándar a nivel internacional para el diseño de software.

UML es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software,

para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.[5]

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces.

Los aspectos definatorios del Proceso Unificado se resumen en tres frases clave: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

Citaremos a continuación los principales diagramas de UML:

- *Clase*: muestra las entidades en un sistema o dominio y la forma en que tales entidades se relacionan entre sí.
- *Objeto*: muestra las instancias de las clases y sus relaciones entre sí. Cada objeto se representa como un rectángulo con nombre.
- *Caso de uso*: muestra la utilización del sistema. Cada caso de uso aparece como una elipse, y cada actor como una figura representativa.
- *Estado*: captura el estado de un objeto durante cierto periodo. Un estado se representa como un rectángulo de vértices redondos, y una transición entre estado como una línea que los conecta.
- *Secuencia*: representa la forma en que interaccionan los objetos entre sí al paso del tiempo. Los objetos se colocan en la parte superior, y el tiempo procede desde la parte superior hacia la parte inferior del diagrama. Las flechas denotan mensajes que van de objeto a objeto.
- *Actividad*: muestra los pasos y puntos de decisión que suceden dentro del comportamiento de un objeto, o dentro de un proceso de negocios. Cada paso es un rectángulo de vértices redondeados (de una forma más ovalada que la representación de un estado) y cada punto de decisión es un rombo.
- *Colaboración*: es otra forma de representar la manera en que los objetos trabajan en conjunto con el tiempo. Los objetos podrían estar en cualquier parte del diagrama. Los mensajes de un objeto a otro aparecen como líneas que conectan a los objetos. Cada línea está numerada de acuerdo con su ubicación en la secuencia de mensajes y muestra información que pertenece a la naturaleza del mensaje.
- *Componente*: modela los componentes de software de un sistema. Cada componente aparece como un rectángulo con dos pequeños rectángulos sobrepuestos en su borde izquierdo.
- *Distribución*: representa la arquitectura física de un sistema de cómputo. Puede mostrar cada equipo de cómputo y dispositivo en el sistema y los componentes que en ellos residen. El equipo de cómputo o nodo se representa como un cubo con los componentes individuales dentro de él.

Facilita herramientas con el fin de definir, detallar, documentar y construir un sistema. En el desarrollo del presente sistema se utilizan diagramas, entre lo que principalmente se muestran:

- De Casos de Uso
- De Clases
- De Secuencia
- De Actividad

#### 1.4 Herramientas de Desarrollo

Para el desarrollo del sistema se utilizan diferentes herramientas dependiendo de cada etapa de acuerdo a nuestra metodología, las cuales se enlistan a continuación:

- Visual Paradigm for UML 10.0[13]. Para ingeniería de software
- WAMPSEVER 2.2[14]. Para ejecución de aplicaciones PHP[15], bases de datos en MySQL[16] y servidor web (Apache).
- Dreamweaver 8[12]. Para diseño y programación de aplicaciones web.

PHP (acrónimo de PHP: Hyper Pre-Processor) es un lenguaje de programación concebido principalmente como herramienta para el desarrollo de aplicaciones WEB. PHP permite diseñar páginas dinámicas de servidor, es decir, generar páginas bajo petición capaces de responder de manera inteligente a las demandas del cliente y que permitan la automatización de gran cantidad de tareas. Si tuviéramos que definir PHP en una sola línea, podríamos decir que es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.[9]

Aunque existe una multitud de lenguajes y entornos de desarrollo para Internet, PHP se ha convertido en uno de los lenguajes, del lado del servidor, más ampliamente utilizados para el desarrollo de páginas dinámicas junto con ASP, JSP, ColdFusion y Perl. En los últimos años, el número de servidores que utilizan PHP se ha disparado.

En PHP se combinan muchas características que contribuyen notablemente a su masiva utilización; entre otras, está el hecho de ser un software de libre distribución y multiplataforma (existen versiones de PHP para Unix, Win32, Mac OS X, etc.) que sigue la filosofía *Open Source*. También ha contribuido a su éxito el hecho de haberse convertido en el complemento ideal para el popular tándem Linux-Apache en el desarrollo de sitios Web. Pero lo más destacable del lenguaje y una de las características que más han influido en su popularización es la sencillez de uso que presenta a los programadores principiantes (se puede desarrollar aplicaciones sencillas en corto tiempo) combinada con las posibilidades avanzadas que proporciona al programador profesional (comunicación con bases de datos, comunicación vía sockets, generación de gráficos, etc.).

Como hemos mencionado, el otro lenguaje utilizado para el diseño de páginas dinámicas de servidor y ampliamente extendido es ASP (Active Server Pages). Es un lenguaje derivado de Visual Basic (aunque se puede programar con otros lenguajes como VBScript y JScript) desarrollado por Microsoft. Evidentemente, se emplea principalmente sobre plataformas que funcionan bajo sistemas operativos Windows (aunque desde hace poco tiempo existe un software de SUN, Sun ONE Active Server Pages, anteriormente conocido como Chili Soft ASP, que permite trabajar con ASP en plataformas Unix/Linux).

PHP es en las plataformas Unix lo que ASP es en las plataformas Windows. De hecho, ambos lenguajes se insertan en documentos HTML haciendo uso de emulación de etiquetas (otros lenguajes como Perl deben generar toda la página HTML de respuesta). Pero lo cierto es que, a pesar de sus semejanzas, las diferencias entre ambos lenguajes son muchas. Diferencias que hacen que la balanza se vaya inclinando hacia PHP como una mejor solución para implementar aplicaciones Web.

La principal diferencia es que ASP es una tecnología propietaria de Microsoft, mientras que PHP sigue la filosofía *Open Source*. Esto hace que ASP esté pensado para funcionar principalmente sobre plataformas Microsoft, a pesar de que existan soluciones -con un costo económico elevado- como Sun ONE ASP que permiten su utilización sobre Unix/Linux. Sin embargo, PHP permite que sin ningún problema podamos migrar nuestras aplicaciones Web entre todos los sistemas operativos y servidores en los que funciona.

La filosofía de producto comercial de ASP influye además en que gran cantidad de software adicional necesario para complementar una aplicación Web supone un costo económico adicional, por ejemplo, ASPencrypt (necesario para la encriptación), ServerObject's Qmail (gestor de correo electrónico) o Arinsans SAFileUp (necesario para la gestión de descargas de archivos). Sin embargo, en PHP todas estas operaciones están incluidas de forma gratuita.

Dentro de otras ventajas sobre ASP, PHP tiene mayor rapidez de ejecución, gestión de memoria más acertada, comunicación de errores y solución más rápida debido a la gran cantidad de usuarios que el *open source* facilita, lo que hace que las revisiones y actualizaciones surjan de manera más rápida.

MySQL se ha convertido en una solución viable de misión crítica para la administración de datos. Antes, MySQL se consideraba como la opción ideal para sitios Web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL supera desde hace tiempo a muchas soluciones comerciales en velocidad y dispone de un sistema de permisos elegante y potente, y ahora, además, a partir de la versión 4 incluye el motor de almacenamiento InnoDB compatible con ACID.[10]

Dispone de funciones de volcado online e incorpora una gran cantidad de funciones nuevas. Son pocas las razones para desechar MySQL como solución de base de datos. MySQL AB, la compañía responsable del desarrollo de MySQL, dispone de un sistema de asistencia eficiente y a un precio razonable, y, como ocurre con la mayor parte de las comunidades de código abierto, encontrara una gran cantidad de ayuda en la Web.

Algunas de las razones para escoger MySQL como solución de misión crítica para la administración de datos:

- **Coste:** El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Asistencia:** MySQL ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- **Velocidad:** MySQL es mucho más rápido que la mayor parte de sus rivales.

- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación. Así mismo, se desarrolla y actualiza de forma mucho más rápida que muchos de sus rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo.
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** MySQL resulta fácil de utilizar y de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

Apache. Una de las características que presenta Apache es que funciona en plataformas virtuales muy utilizadas. Al principio, Apache se utilizaba para ser el primer servidor Web basado en Unix, pero esto ya no es verdad. Apache no sólo funciona en la mayoría (prácticamente en todas) las versiones de Unix sino que, además, funciona en Windows 2000/NT/9x y en muchos otros sistemas operativos de escritorio y de tipo servidor como son Amiga OS. [11]

Apache presenta muchas otras características, entre ellas un elaborado índice de directorios; un directorio de alias; negociación de contenidos; informe de errores HTTP configurable; ejecución SetUID de programas CGI; gestión de recursos para procesos hijos; integración de imágenes del lado del servidor; reescritura de las URL; comprobación de la ortografía de las URL; y manuales online.

El resto de características importantes de Apache son:

- Soporte del último HTTP 1.1: Apache es uno de los primeros servidores Web en integrar el protocolo HTTP 1.1. Es totalmente compatible con el nuevo estándar HTTP 1.1 y al mismo tiempo sigue siendo compatible con HTTP 1.0. Apache está preparado para todas las novedades del nuevo protocolo.
- Sencillo, con la configuración basada en un poderoso archivo: el servidor de Apache no posee una interfaz de usuario gráfica para su administración. Se trata de un sencillo archivo de configuración llamado httpd.conf que se puede utilizar para configurar Apache.
- Soporte para CGI (Common Gateway Interface): Apache soporta CGI utilizando los módulos mod\_cgi y mod\_cgid. Es compatible con CGI y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores Web.
- Soporte de FastCGI: no todo el mundo escribe sus CGI en Perl. ¿cómo pueden hacer sus aplicaciones CGI más rápidas? Apache también tiene una solución para esto. Utilice el módulo mod\_fcgi para implementar un entorno FastCGI dentro de Apache y haga sus aplicaciones FastCGI arranquen rápidamente.

- Soporte de host virtuales: Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.
- Soporte de autenticación HTTP: Apache soporta autenticación básica basada en la Web. Está también preparado para autenticación basada en la digestión de mensajes, que es algo que los navegadores Web populares ya han implementado. Apache puede implementar autenticación básica utilizando tanto archivos estándar de contraseña como los DBM, llamadas a SQL o llamadas a programas externos de autenticación.
- Perl integrado: Perl se ha convertido en el estándar para la programación de scripts CGI. Apache es seguramente uno de los factores que hacen de Perl un lenguaje de programación CGI tan popular. Apache se encuentra más cerca de Perl que nunca. Puede bajar un script CGI basado en Perl a la memoria utilizando su módulo mod\_perl, y reutilizarlo tantas veces como necesite. Este proceso elimina las desventajas del arranque que se encuentran asociadas a menudo con los lenguajes de interpretación como Perl.
- Soporte de scripts PHP: este lenguaje de script ha comenzado a ser muy utilizado y Apache ofrece un amplio soporte de PHP utilizando el módulo mod\_php.
- Soporte de servlets de Java: los servlets de Java y las Java Server Pages (JSP) se están convirtiendo en algo muy común en los sitios Web dinámicos. Puede ejecutar servlets de Java utilizando el premiado entorno Tomcat con Apache.
- Servidor proxy integrado: puede convertir Apache en un servidor proxy caché. Sin embargo, la implementación actual del módulo opcional de proxy no soporta HTTP proxy o el último protocolo HTTP 1.1. Se está planeando actualizar este módulo muy pronto.
- Estado del servidor y adaptación de registros: Apache da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web. Además, puede adaptar sus archivos de registro a su gusto.
- Soporte de Server Side Includes (SSI): Apache ofrece un conjunto de Server Side Includes que añaden una gran cantidad de flexibilidad para el desarrollador del sitio Web.
- Soporte de Secured Socket Layer (SSL); puede crear fácilmente un sitio Web SSL utilizando OpenSSL y el módulo mod\_ssl de apache.

## CAPÍTULO II. ANÁLISIS Y DISEÑO DEL SISTEMA

### 2.1 Planteamiento del Problema

En el área de software de base de la Facultad de Ciencias de la Computación en la BUAP, se plantea la necesidad de un sistema de información que ayude a la colaboración e intercambio de medios o recursos para efectos de un mejor desempeño de los alumnos en su desarrollo académico y de una mejor organización de la información por parte de los profesores.

Para tal efecto, se plantea lo siguiente:

- El área de Software de Base está formada por diferentes materias (Ensamblador, Sis. Oper., Programación de Sistemas, etc.).
- Cada materia tiene como atributos una clave (NRC) y pertenece a una carrera (ING, LIC, IT).
- El área de Software de Base también está formada por Profesores adscritos a dicha área.
- Cada profesor además de sus datos generales (Nombre, grado), tendrá un identificador único (ID).
- En base a la documentación de la Secretaría Académica de esta facultad, existe una clasificación de ésta área (y las demás), dividiéndose en PRIMARIAS, SECUNDARIAS O TERCARIAS; cada profesor está asignado una y sólo una vez a alguna de estas divisiones, de ésta misma forma están asignadas las Materias.
- Cada profesor tendrá la posibilidad de subir sus Recursos (Apuntes, Proyectos, Artículos, Software); hacer Avisos (Fecha de Inicio, Fecha Fin, Descripción)
- Los Profesores podrán enrolarse a una Materia (para saber qué materia imparten).
- Cada Materia tendrá disponibles o asignados ciertos Recursos de sus respectivos Profesores.
- El sistema tendrá un Usuario Coordinador es uno de los profesores de ésta área, además de las funciones que tiene como profesor, podrá subir al servidor el Documento de alguna Reunión; dar de alta o bajas, y editar la información de los Profesores.
- Los alumnos así como los usuarios generales del sistema web, podrán realizar consultas acerca de los recursos que ofrece ésta plataforma, así como también

tener un espacio para hacer comentarios, preguntas, sugerencias, etcétera, sin que de forma obligatoria se pida saber su identidad personal.

- Los comentarios deberán tener como restricción el uso de un lenguaje adecuado, excluyendo palabras ofensivas o groserías, para lo cual se pide que exista un método de filtrado de contenido el cual no permita la impresión de este tipo de contenido.

## 2.2 Modelo de Casos de Uso

Con ayuda del planteamiento del problema, podemos distinguir entre tres tipos de usuario: alumno, profesor y coordinador.

### 2.2.1 Diagrama de Casos de Uso

Los casos de uso son las diferentes operaciones que los usuarios requieren llevar a cabo, por ejemplo: “*el alumno puede hacer comentarios o sugerencias*”. La *Figura 2.1* muestra gráficamente éste concepto:

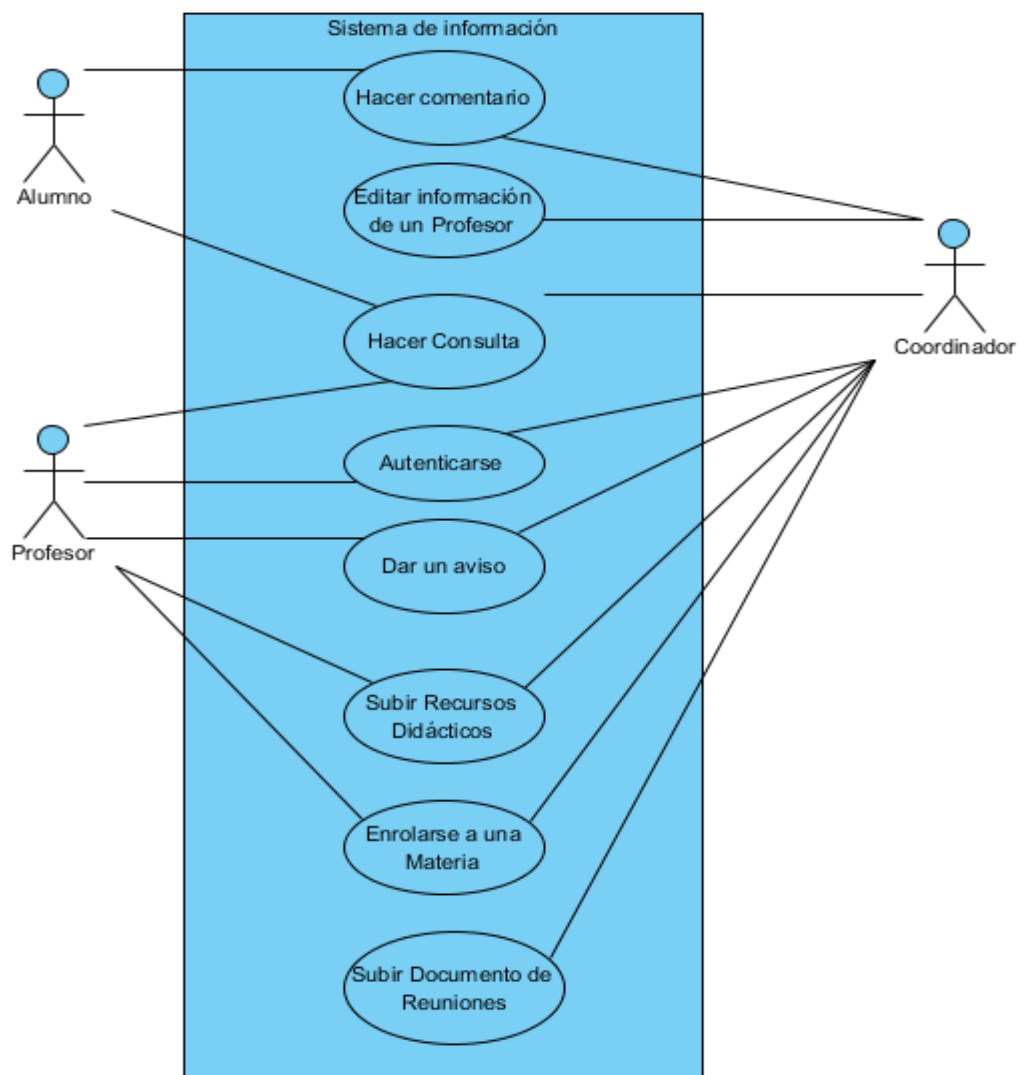


Figura 2.1. Diagrama de Casos de Uso

## 2.2.2 Documentación y Flujo de Eventos

A continuación se detalla cada uno de los casos de uso incluyendo el flujo de eventos y los posibles escenarios que se presentan en cada uno de ellos.

### c.1.Hacer Comentarios:

#### **Actores:**

Alumno, Profesor, Coordinador.

#### **Descripción:**

Permite a los diferentes usuarios del sistema realizar un comentario, duda, sugerencia o algún otro contenido, siempre y cuando se haga uso de un lenguaje adecuado.

#### **Flujo de eventos:**

##### *Flujo principal*

Se le ofrece al usuario la opción para realizar comentarios.

En pantalla se muestran los campos necesarios para que el usuario pueda completar su comentario.

Se implementará una función *Captcha* para validar el envío del formulario.

El usuario selecciona la opción "Enviar".

Una vez ingresado su comentario, se analizará el contenido.

Analizando contenido.

El contenido del comentario se analizará con el objeto de que no sea obsceno o quebrante el sano funcionamiento moral del sistema.

Comentario aprobado.

Si el comentario es aceptado bajo las políticas anteriores, se publicará en ese instante.

Se otorga un mensaje de confirmación al Alumno.

Se otorga la opción al Alumno de regresar a la página principal.

##### *Flujo de excepción*

Los campos marcados como obligatorios no están llenos.

Se emitirá un mensaje para invitar al usuario a completarlos, la información no podrá ser enviada hasta que el campo contenga la información requerida.

Analizando el contenido.

Si el contenido contuviera palabras obscenas, se le emitirá un mensaje al usuario sugiriéndole quitarlas ya que no se publicará su comentario. Se le mostrará la opción para que lo intente de nuevo.

## **c.2.Hacer Consultas:**

### **Actores:**

Alumno, Profesor, Coordinador.

### **Descripción:**

Los usuarios tendrán la posibilidad de realizar consultas de todo el material disponible en la plataforma.

### **Flujo de eventos:**

#### *Flujo principal*

En la plataforma se tendrá disponible una barra de búsqueda.

En ella el usuario ingresará alguna palabra o frase clave que permita iniciar la búsqueda.

El usuario selecciona el botón "Buscar".

Se inicia la búsqueda.

Realizando búsqueda.

De acuerdo a los criterios de búsqueda, se desplegarán los resultados correspondientes.

Los resultados se ofrecerán como ligas para ingresar al sitio o descargar el archivo o la página web, en caso de que aplique.

Finaliza la búsqueda.

#### *Flujo de excepción*

No se encontraron resultados.

En el caso de que las palabras que ingrese el usuario no se encuentren en la base de datos del sistema, se enviará un mensaje notificando la situación y se le brindará la opción de intentarlo de nuevo o de regresar a la página principal.

### **c.3.Autenticarse:**

#### **Actores:**

Profesor, Coordinador.

#### **Descripción:**

Tanto a los Profesores como a los Coordinadores se les otorgan un nombre de usuario y una contraseña con la cual se valida el acceso al sistema.

#### **Flujo de eventos:**

##### *Flujo principal*

En la página principal existe un espacio para ingresar los datos de autenticación.

El usuario ingresa su nombre de usuario.

El usuario ingresa su contraseña.

El usuario selecciona la opción "enviar".

Validando cuenta.

Se valida la existencia del registro en la base de datos.

Si el usuario es correcto, se otorga la interfaz correspondiente: Profesor o Coordinador.

Otorgar interfaz.

Se otorga la interfaz. Esto le da al usuario que ingresó al sistema las herramientas para trabajar bajo su perfil.

Una vez que el usuario haya realizado las operaciones que necesita, se le da la opción para salir del sistema ("salir").

Salir del sistema.

El usuario termina su sesión.

##### *Flujo de excepción*

Error de autenticación.

Se emite un mensaje de error de fallo de autenticación.

Se borran los datos ingresados y se piden nuevamente.

#### **c.4.Realizar un Aviso:**

##### **Actores:**

Profesor, Coordinador.

##### **Descripción:**

El Profesor podrá hacer un aviso dirigido al área de software de base o a los demás usuarios del sistema

##### **Flujo de eventos:**

###### *Flujo principal*

Agregando un aviso.

La información que se requiere para dar un aviso es la siguiente:

- Se solicita el nombre del aviso “Asunto”, de carácter obligatorio.
- Se pide una “Descripción”. Se otorga un campo de 250 caracteres. Campo no obligatorio.
- “Fecha y hora” de inicio. Campo obligatorio.
- “Fecha” de fin. Campo no obligatorio.
- Además se mostrará el nombre del Profesor quien hace el aviso.

El usuario puede cancelar (al omitir) o confirmar la emisión del aviso. Selecciona la opción correspondiente.

Editando un aviso.

Se muestra al usuario los avisos que ha emitido, de los cuales se ofrece la opción de “Editar”.

La información que podrá editar el usuario es la siguiente:

- “Asunto”. Nombre del aviso. Campo obligatorio
- “Descripción”. Se otorga un campo de 250 caracteres. Campo no obligatorio.
- “Fecha y hora” de inicio. Campo obligatorio.
- “Fecha” de fin. Campo no obligatorio.

El usuario puede cancelar o confirmar la emisión del aviso. Selecciona la opción correspondiente.

Eliminando un aviso.

Se muestra al usuario la opción “Eliminar aviso” en el panel principal.

El usuario selecciona la opción.

Se muestran los avisos que el usuario ha hecho en forma de lista exponiendo la información de cada uno (Asunto, Fecha, Descripción).

Se ofrece el botón “Eliminar” para eliminar el aviso seleccionado

Si selecciona Eliminar, se emite un mensaje de “Alerta” y se pide confirmación: “Eliminar”.

Publicando un aviso.

Una vez que el usuario emite el aviso, se mostrará en la zona de “Avisos” en la plataforma.

#### *Flujo de excepción*

El usuario selecciona la opción Avisos pero no es la opción que requiere.

En cualquier momento puede seleccionar alguna otra opción.

Agregando un aviso, campos obligatorios no llenados.

Se verifican los campos y se le hace invitación a completarlos, el formulario no podrá ser enviado hasta que cumpla con este requerimiento.

El usuario tendrá la oportunidad de completar los campos, o de acceder a alguna otra sección del sistema.

Editando un aviso, campos obligatorios no llenados.

Se verifican los campos y se le hace invitación a completarlos, el formulario no podrá ser enviado hasta que cumpla con este requerimiento.

El usuario tendrá la oportunidad de completar los campos, o de acceder a alguna otra sección del sistema.

### **c.5.Subir un Recurso Didáctico:**

#### **Actores:**

Profesor, Coordinador.

#### **Descripción:**

El usuario podrá subir los apuntes o notas de su curso para que el contenido pueda ser consultado por los alumnos. En esta sección se consideraran como recursos los siguientes: proyectos, artículos, ejercicios, notas, apuntes, software.

#### **Flujo de eventos:**

##### *Flujo principal*

El Profesor selecciona la opción “Recursos” del panel principal.

Aquí el Profesor tendrá la opción de dar un “agregar o editar recursos” y “eliminar recursos”.

El usuario selecciona la operación.

Agregando un recurso.

Se muestran las siguientes opciones:

- “Seleccionar archivo”. Aquí se seleccionara el archivo desde la ubicación del profesor, hacia la base de datos del sistema. Campo obligatorio.
- “Nombre”. Se ingresa el nombre que el Profesor desea que se muestre para los alumnos. Se otorgara un espacio de 100 caracteres. Campo obligatorio.
- “Materia”. El profesor asignará el material a alguna materia del área. Campo obligatorio.
- “Tipo”. Se puede asignar un tipo de contenido, como se mencionó en la descripción. Campo no obligatorio.

Una vez que los campos están completos se le ofrecen las opciones de “Subir recurso”.

Editando un recurso.

Se muestran las siguientes opciones:

- “Cambiar archivo”. Aquí se seleccionara el archivo que el usuario desea cambiar, si no lo modifica, se queda el anterior. Campo obligatorio.
- “Nombre”. El usuario podrá cambiar el nombre del archivo. Se otorgara un espacio de 100 caracteres. Campo obligatorio.
- “Materia”. El profesor asignará el material a alguna materia del área. Campo obligatorio.
- “Tipo”. Se puede asignar un tipo de contenido, como se mencionó en la descripción. Campo no obligatorio.

Una vez que los campos están completos se le ofrecen las opciones de “Guardar cambios”.

Eliminando un recurso.

Se muestra al usuario la opción “Eliminar recurso” en el panel principal.

El usuario selecciona la opción.

Se muestran los recursos que el usuario ha subido en forma de lista exponiendo la información de cada uno (Nombre, Materia).

Se ofrecen los botones “Eliminar” para eliminar el recurso seleccionado.

El usuario selecciona la opción.

Al seleccionar Eliminar, se emite un mensaje de “Alerta” y se pide confirmación: “Eliminar”.

Publicando un recurso.

Una vez que el Profesor ha hecho las modificaciones necesarias. Se publicara su recurso en la sección “Recursos” de la plataforma.

*Flujo de excepción*

El usuario selecciona la opción Recursos pero no es la opción que requiere.

En cualquier momento puede seleccionar alguna otra opción.

Agregando recurso.

Se validan los campos obligatorios. Si faltase alguno, se notificará y se le indicara al usuario ingresar un valor válido.

Editando un recurso.

Se validan los campos obligatorios. Si faltase alguno, se notificará y se le indicara al usuario ingresar un valor válido.

## **c.6.Inscribirse a una Materia:**

### **Actores:**

Profesor, Coordinador.

### **Descripción:**

Los usuarios podrán inscribirse a una materia para manifestar la o las materias que estén impartiendo en ese momento.

### **Flujo de eventos:**

#### *Flujo principal*

Se le muestra una opción al usuario “Mis materias” en el panel principal.

El usuario selecciona esta opción para verificar las materias a las que está inscrito o para agregar alguna nueva.

Enrolándose a una materia.

Se ofrece la opción “Inscribirme a una materia” para agregar una materia.

Aparece el listado de materias disponibles.

El usuario selecciona la que le interesa.

Se ofrece un mensaje de confirmación en donde se muestra la información de la materia en la que está a punto de darse de alta.

Al confirmar la operación (mediante la opción “inscribir”), se emitirá un mensaje que confirma la operación.

La materia inscrita se visualizará en la sección “Mis materias”.

Eliminando una materia.

Se ofrece la opción “Dar de baja de una materia” para quitar alguna materia de la cual ya no requiera su información.

Se muestra la lista de materias a las que está de la cual seleccionará la materia que desee dar de baja.

Una vez hecha la selección, se ofrecen las opciones “Continuar” para realizar la operación.

Si selecciona la opción, se emitirá un mensaje de “Alerta” para confirmar, y el botón “Dar de baja”.

El usuario seleccionara la operación en caso de requerirla.

#### *Flujo de excepción*

El usuario selecciona la opción Mis materias pero no es la opción que requiere.

El usuario podrá cambiar de sección en cualquier momento.

Enrolándose a una materia, materia ya inscrita.

Se validará en la base de datos si es que el Profesor ya se ha inscrito a esa materia con anterioridad, de ser así, se mostrará un mensaje con esta observación.

Visualizando materias, ninguna materia inscrita.

En el caso de que el usuario no se haya inscrito a ninguna materia, se mostrará un mensaje notificando este estado.

El usuario puede cambiar de sección en cualquier momento.

Eliminando materia, ninguna materia inscrita.

En el caso de que el usuario no se haya inscrito a ninguna materia, se mostrará un mensaje haciendo esta notificación.

El usuario puede cambiar de sección en cualquier momento.

### **c.7.Subir Documento de Reunión:**

#### **Actores:**

Coordinador.

#### **Descripción:**

En esta sección el Profesor tendrá la posibilidad de subir o actualizar las actas de las reuniones hechas en el año.

#### **Flujo de eventos:**

##### *Flujo principal*

Se le muestra una opción al usuario "Reuniones" en el panel principal.

El usuario selecciona esta opción para verificar las actas de las reuniones hechas en el año.

Subiendo un acta de una reunión.

Se muestra al Profesor una opción llamada "Agregar acta", para agregar un acta a la base de datos del sistema.

La información a solicitar es la siguiente:

- "Seleccionar archivo": aquí el usuario puede subir el documento del acta. Campo obligatorio.
- "Nombre": generalmente es el relativo al número de acta y la fecha. Campo obligatorio.

Se ofrece la opción "Agregar" para agregar el documento o "Cancelar" para cancelar la operación y regresar a la página anterior.

Editando una reunión.

Se muestra la opción "Editar" para modificar los datos de una reunión.

La información que se puede editar es la siguiente:

- "Modificar archivo": aquí el usuario puede subir el documento del acta. Campo obligatorio.
- "Nombre": generalmente es el relativo al número de acta y la fecha. Campo obligatorio.

Se ofrece la opción "Aceptar" para hacer la modificación de la reunión o "Cancelar" para cancelar la operación y regresar a la página anterior.

Publicando una reunión.

El documento se mostrará en la sección "Reuniones" de la plataforma para su consulta.

Eliminando una reunión.

Se muestra la opción "Eliminar" para eliminar un acta del sistema.

Se muestran las actas que se han subido en forma de lista exponiendo la información de cada una (Nombre).

Se ofrecen los botones “Eliminar” para eliminar el acta seleccionada.  
El usuario selecciona la opción preferente.  
Si es Eliminar, se emite un mensaje de “Alerta” y se pide confirmación:  
“Eliminar”.

#### *Flujo de excepción*

El usuario selecciona la opción Mis materias pero no es la opción que requiere.

En cualquier momento el usuario puede pasar a cualquier sección.

Subiendo un acta de una reunión.

Se validan los campos obligatorios, en caso de faltar alguno el usuario será notificado para que ingrese un dato válido o la opción “Cancelar” para cancelar el procedimiento.

Editando una reunión.

Se validan los campos obligatorios, en caso de faltar alguno el usuario será notificado para que ingrese un dato válido o la opción “Cancelar” para cancelar el procedimiento.

### **c.8.Dar de Alta a un Profesor:**

#### **Actores:**

Coordinador.

#### **Descripción:**

Esta operación le otorgará al Administrador (el coordinador del área) del sistema, dar de alta a un profesor, definiendo así el perfil correspondiente.

#### **Flujo de eventos:**

##### *Flujo principal*

Se ofrece al administrador la opción "Alta profesor":

Se otorga el formulario correspondiente para dar de alta a un profesor.  
Se muestra información sobre campos obligatorios.

Validando campos:

Se verifica que los campos marcados como obligatorios estén definidos en la información que haya ingresado el usuario.

Si el sistema detecta que no es así, mostrará un mensaje en el cual se le invitara a llenarlo.

Si los campos obligatorios no están ingresados, el formulario no podrá enviarse.

Confirmando registro:

Una vez que los campos ingresados sean correctos se confirmará el registro.

El nuevo Profesor ya está dado de alta.

##### *Flujo de excepción*

Cancelar registro.

En cualquier momento el Administrador podrá cancelar la operación de dar de alta a un Profesor.

Campos ingresados no válidos.

El sistema notificará algún error al momento de ingresar alguno de los datos necesarios.

### **c.9.Editar Información de un Profesor:**

#### **Actores:**

Coordinador.

#### **Descripción:**

El usuario tiene la posibilidad de editar o cambiar la información de los profesores dados de alta.

#### **Flujo de eventos:**

##### *Flujo principal*

Se le muestra una opción al usuario “Editar información de profesor” en el panel principal.

Al seleccionar esta opción se mostrará en pantalla la información de los profesores dados de alta.

El usuario deberá seleccionar el número de profesor del cual desea editar su información y seleccionar el botón “Continuar”.

Editando información de un profesor.

Una vez seleccionado el profesor, se habilitaran los campos de la información del profesor.

Los campos estarán con los valores que ya existen de ese profesor, de tal forma que si no se afectara nada, los valores se conserven en la base de datos.

El usuario puede cambiar los campos que considere necesarios.

Validando campos.

Se verifica que los campos marcados como obligatorios estén definidos en la información que haya ingresado el usuario.

Si el sistema detecta que no es así, mostrará un mensaje en el cual se le invitara a llenarlo.

Si los campos obligatorios no están ingresados, el formulario no podrá enviarse.

Confirmando registro.

Una vez que los campos ingresados sean correctos se confirmará el registro.

El nuevo Profesor ya está dado de alta.

##### *Flujo de excepción*

Cancelar registro.

En cualquier momento el Administrador podrá cancelar la operación de dar de alta a un Profesor.

Campos ingresados no válidos.

El sistema notificará algún error al momento de ingresar alguno de los datos necesarios.

### **c.10.Eliminar a un Profesor:**

#### **Actores:**

Coordinador.

#### **Descripción:**

Operación que permite dar de baja del sistema a un profesor o coordinador.

#### **Flujo de eventos:**

##### *Flujo principal*

Se le muestra una opción al usuario “Eliminar a profesor” en el panel principal.

Al seleccionar esta opción se mostrará en pantalla la información de los profesores dados de alta.

El usuario deberá seleccionar el número de profesor que desea eliminar y seleccionar el botón “Continuar”.

Confirmación de operación.

Una vez seleccionado el profesor, se muestra un mensaje de “Alerta” pidiendo al usuario confirmar la operación

Se muestra la información del profesor que está por ser modificado.

Eliminando profesor.

Si el usuario confirma la operación, el sistema dará de baja al profesor.

Se muestra un mensaje de confirmación de operación.

##### *Flujo de excepción*

Cancelar operación.

En cualquier momento el Administrador podrá cancelar la operación de dar de baja a un Profesor.

## 2.3 Glosario de Términos

### **Recurso didáctico:**

Todo aquel material que se ha elaborado con la intención de facilitar al docente su función y a su vez la del alumno en con contexto educativo.

### **Documento de reunión:**

Documento en el cual se sustenta la información tratada en determinada reunión de academia.

### **Aviso:**

Información importante que se quiera transmitir.

### **Comentario:**

Apreciación sobre cualquier cosa puesta en análisis. En este sistema, se incluyen aquí las posibilidades, de incluir dudas o sugerencias o algún otro tipo, siempre y cuando se utilice un lenguaje adecuado.

## 2.4 Modelo Conceptual

Para este sistema se tomo en cuenta la teoría acerca del Modelo Entidad Relación [6]. Éste ayudara a diseñar la perspectiva acerca de los datos que se van a manejar.

### 2.4.1 Definición de Entidades.

Se listan a continuación los objetos sobre los que se recoge información.

Materias: Clave, Nombre, Carrera

Profesores: Id, Nombre, Rol

División: Tipo

Recursos: Tipo, Archivo, Descripción

Avisos: Fecha de inicio, Fecha de fin, Descripción

Reunión: Fecha, Documento

### 2.4.2 Identificación de Relaciones

A continuación, en la *Figura 2.2* se muestran las asociaciones entre las entidades anteriores.

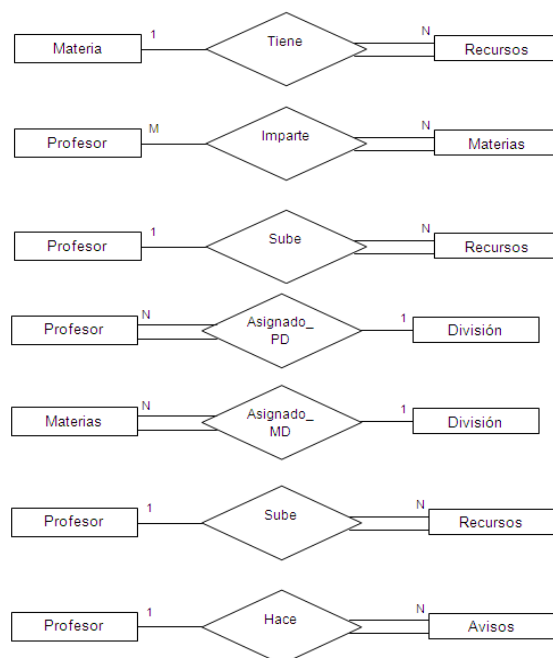


Figura 2.2 Relaciones entre Entidades

### 2.4.2 Diagrama de Clases Simple

En base al esquema del modelo conceptual descrito anteriormente, podemos definir un modelo de clases simple (Figura 2.3) que ayuda de mejor manera describir las relaciones entre estas clases.

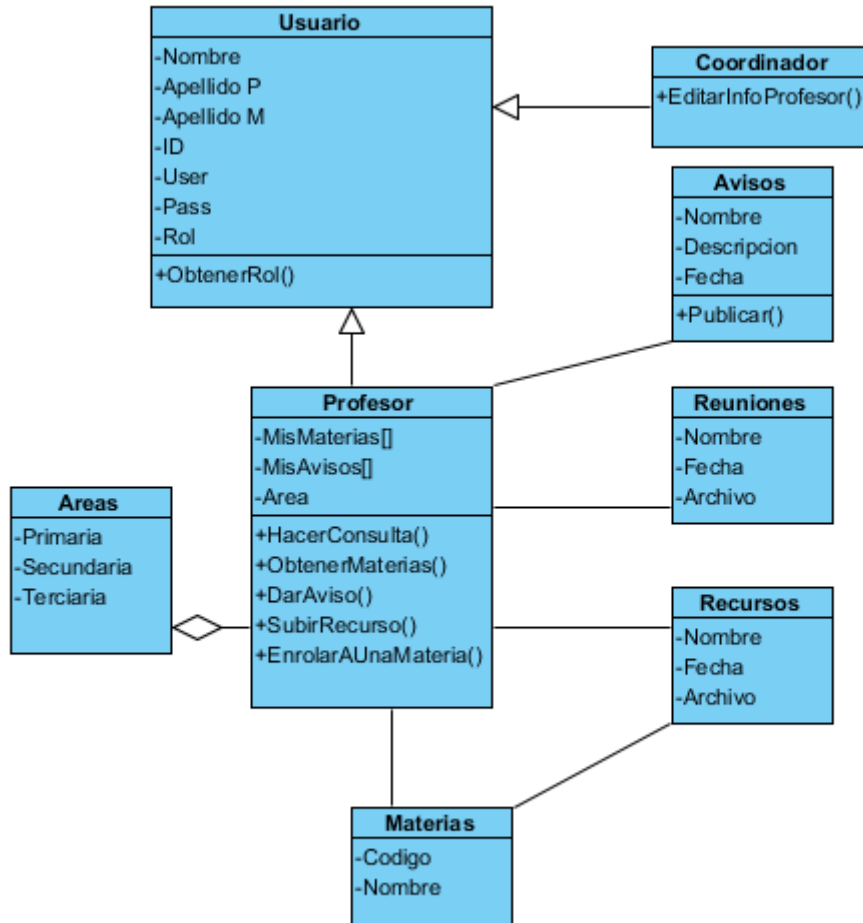


Figura 2.3 Diagrama de Clases Simple

## 2.5 Modelo de Análisis

Debido a la importancia de especificar la comunicación entre objetos, además de los diagramas de clases, nos ayudaremos de otro tipo de diagramas. Por una parte, los diagramas de secuencia, en donde resulta más claro definir los pasos de mensajes entre objetos definidos en el sistema. En otro caso, utilizaremos los diagramas de actividad que describen el comportamiento de cada una de las clases.

### 2.5.1 Diagramas de Secuencia

En la Figura 2.4 analizaremos los procedimientos de un alumno o usuario en general a los cuales se puede acceder directamente en el sistema.

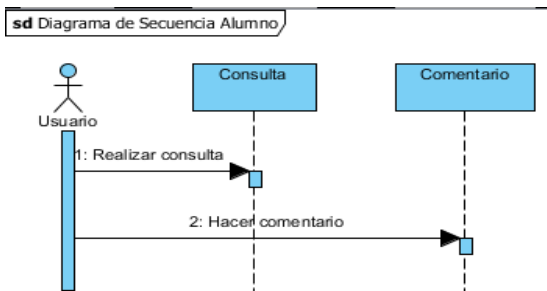


Figura 2.4 Diagrama de Secuencia Alumno

Como podemos ver, desde la vista principal de nuestro sistema se tendrán disponibles las funciones para realizar Consultas y Comentarios; de esta manera, lo único que tiene que hacer el usuario es ingresar a estas funciones ingresando la información necesaria para obtener los resultados requeridos.

A continuación, en la Figura 2.5 se muestra el caso del usuario con perfil de profesor. Entra la parte de autenticación o *logueo* para tener acceso a las diferentes funciones de este usuario.

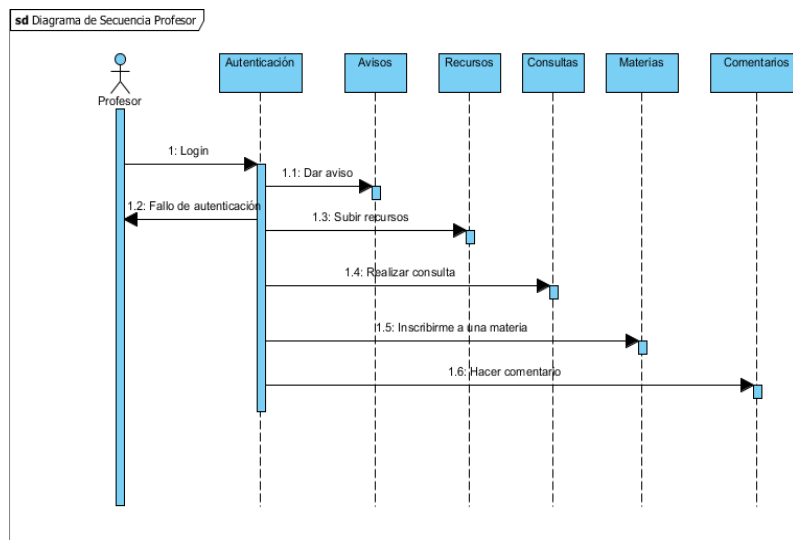


Figura 2.5 Diagrama de Secuencia Profesor

Podemos observar que las opciones del profesor están limitadas al ingreso correcto al sistema mediante un nombre de usuario y contraseña que se otorgará a los usuarios correspondientes.

Es importante también mencionar que en el apartado de comentarios, no se tomará información de quién publique este tipo de contenido inclusive para los casos en que el usuario haya sido identificado por el sistema (como profesores o coordinador), a menos que el usuario así lo decida ingresando de forma manual información de su identidad (nombre por ejemplo).

En seguida el caso del usuario perfil de coordinador.

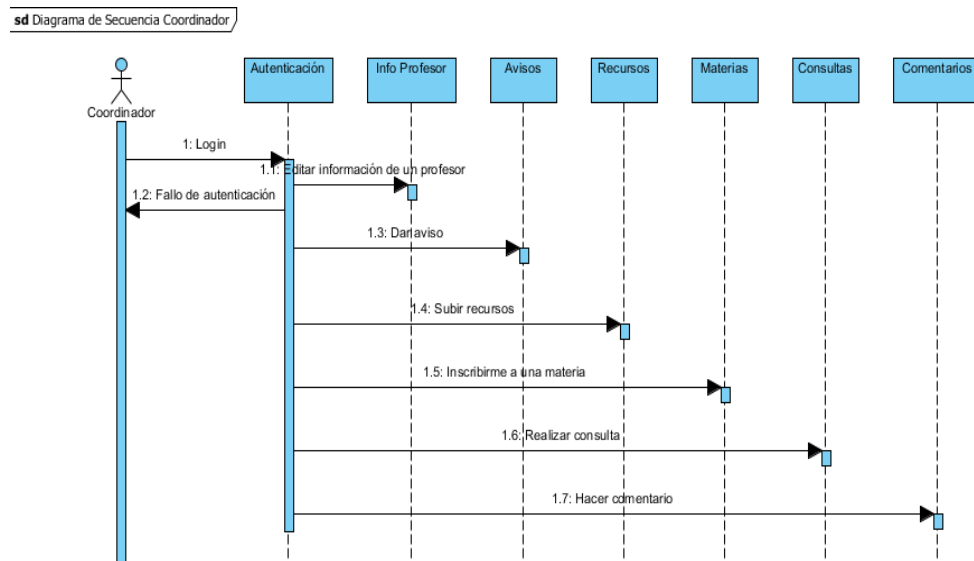


Figura 2.6 Diagrama de Secuencia Coordinador

En la *Figura 2.6* observamos los procedimientos relacionados con el usuario con perfil coordinador, añadiéndose las opciones relacionadas con el alta, baja, y modificación de información de profesores.

## 2.5.2 Diagramas de Actividad

En esta sección pasaremos a observar el comportamiento de los usuarios dentro del sistema y su operatividad con ayuda de los siguientes diagramas de actividad.

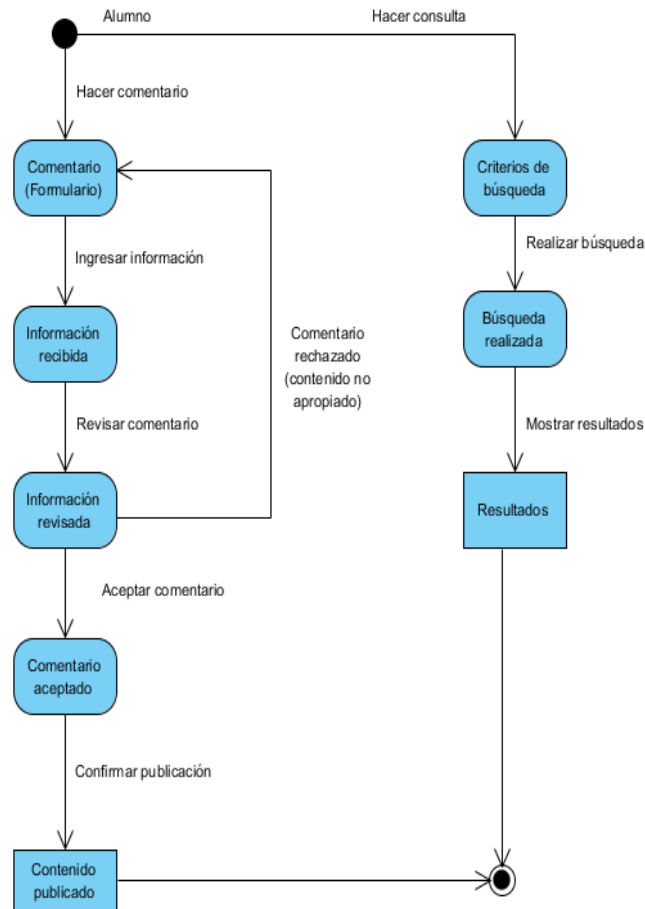


Figura 2.7 Diagrama de Actividad Alumno

En la *Figura 2.7* podemos observar el flujo que debe seguir la aplicación el momento de hacer un comentario o una consulta desde un usuario normal o con el perfil de alumno.

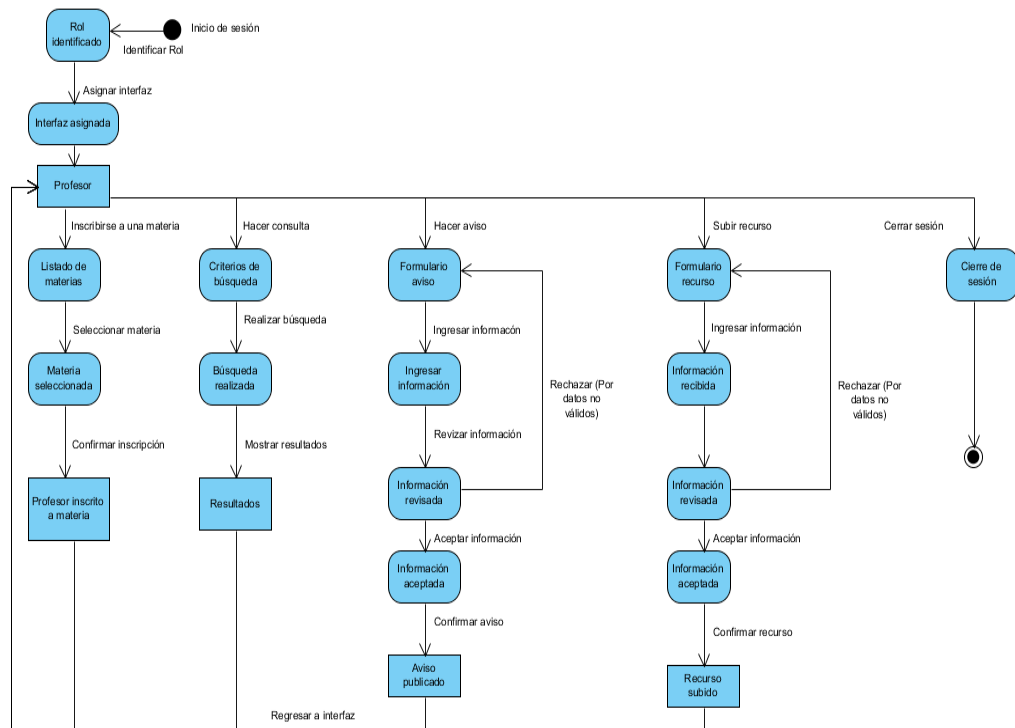


Figura 2.8 Diagrama de Actividad Profesor

Ilustrado en la *Figura 2.8* vemos el caso del usuario profesor y sus diferentes operaciones, así como sus respectivas excepciones.

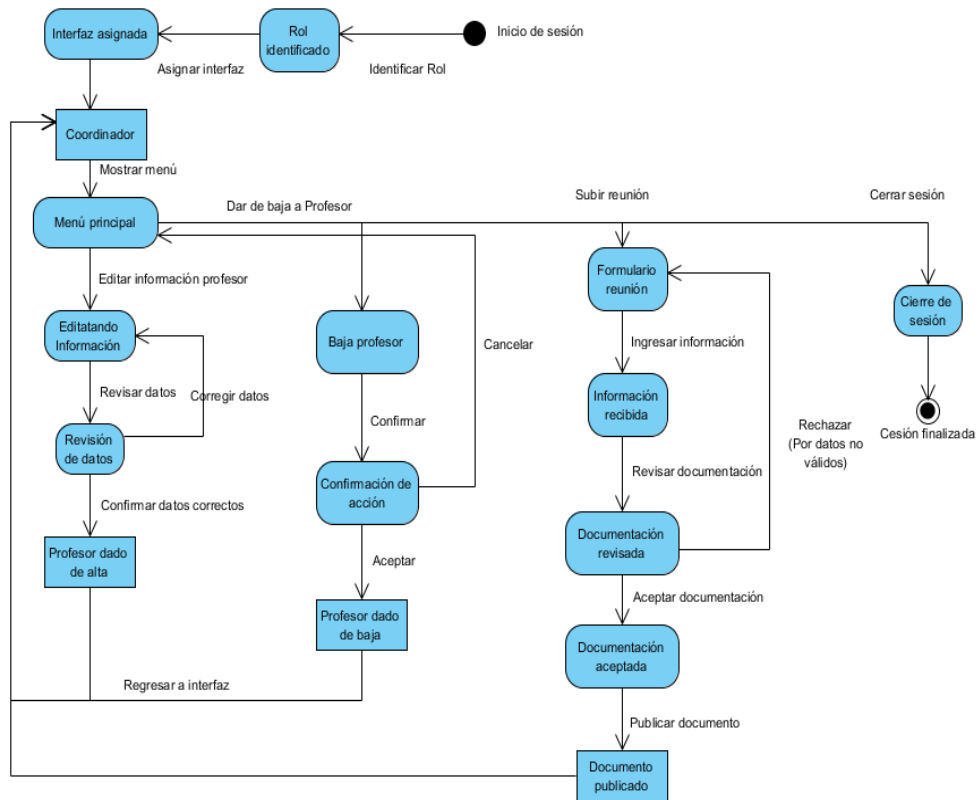


Figura 2.9 Diagrama de Actividad Coordinador

Por último, en la *Figura 2.9* encontramos el diagrama de actividad del usuario coordinador, quien nuevamente muestra la actividad adicional relacionada con la edición de información de profesores.

## 2.6 Análisis y Diseño de la Base de Datos.

En la *Figura 2.10* tenemos el diagrama entidad relación que refleja la información obtenida del análisis de requerimientos derivado del modelo conceptual del sistema.

### 2.6.1 Diagrama Entidad Relación

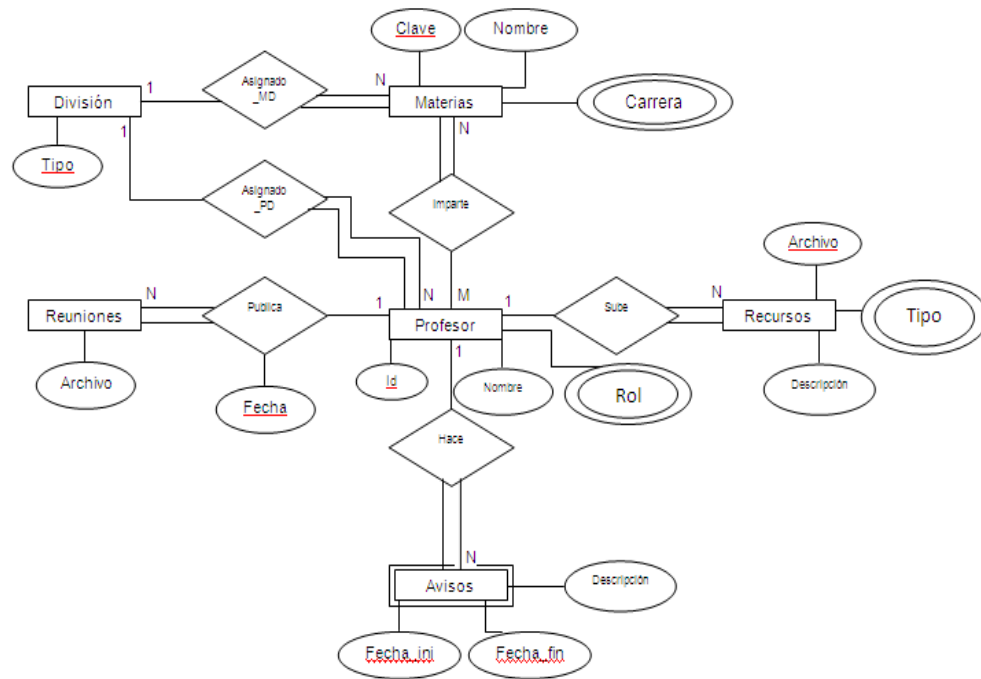


Figura 2.10 Diagrama Entidad Relación

Del diagrama anterior y siguiendo el procedimiento correspondiente comúnmente denominado como mapeo [6], podemos distinguir las diferentes entidades, sus atributos, y las relaciones entre ellas, con lo cual, obtenemos es esquema relacional que a continuación se muestra en la *Figura 2.11*.

## 2.6.2 Esquema Relacional

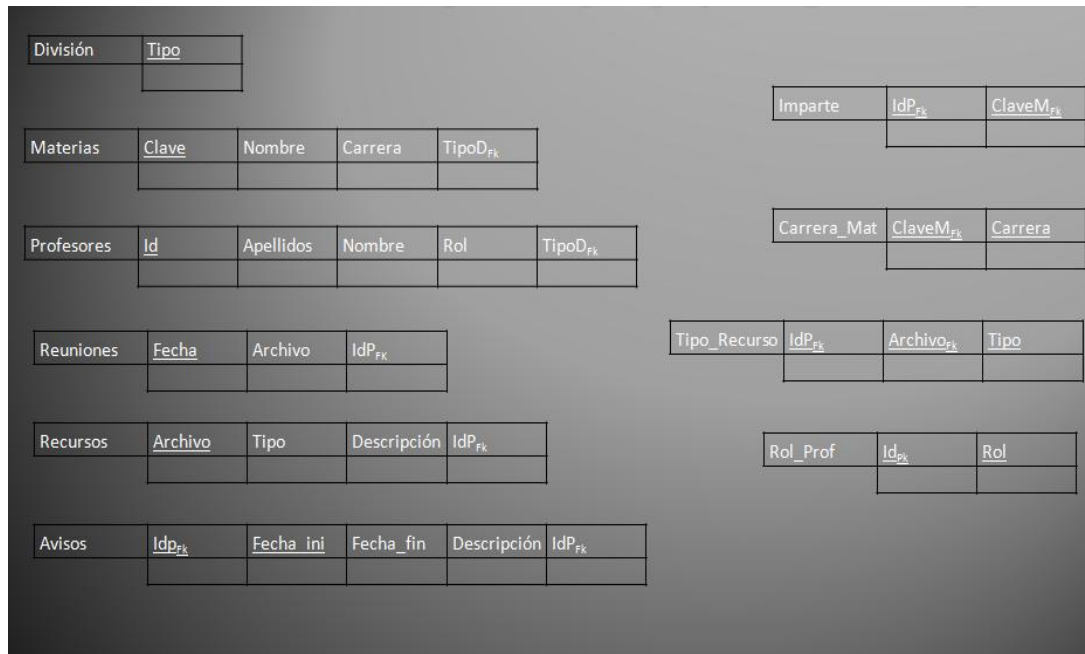


Figura 2.11 Esquema Relacional de la Base de Datos

Gracias a este procedimiento, podemos asegurar que la base de datos planteada cumple con las características de normalización. En este caso, la base está normalizada hasta la tercera forma normal (3FN)[6].

## CAPÍTULO III. IMPLEMENTACIÓN Y PRUEBAS

En este sistema podemos distinguir tres tipos de actores, un usuario general que llamaremos Alumno, donde éste podrá tener acceso a la información de la plataforma a modo de consulta así como también poder realizar comentarios. Por otro lado, el usuario que llamaremos Profesor, quien además de las posibilidades de un alumno, tendrá la posibilidad de subir recursos, emitir avisos e inscribirse a materias para saber qué materias está impartiendo. Por último, el usuario Coordinador, en donde también se incluirán funciones de Alumno y de Profesor, y además las de subir el documento de una reunión y editar la información de los profesores del área.

### 3.1 Alumno

#### 3.1.1 Bienvenida y Menús



Figura 3.1 Bienvenida y Menús

En la *Figura 3.1* podemos observar nuestra ventana de bienvenida así como el menú superior y el lateral a la derecha.

En el menú superior podemos ingresar a la información que haya sido dada de alta tanto por los profesores como por el coordinador del área, y podemos ver que se divide en materias, profesores, licenciaturas, recursos reuniones y avisos.

En menú de la derecha está dividido en cuatro partes: Menú principal, en donde encontramos la opción para realizar las consultas; Relevantes, donde se muestran los avisos de los profesores, se muestra información de la fecha de su aviso, el mensaje y quién lo realizó; Contacto, encontramos las opciones Acerca de que muestra información acerca del sistema, y la de Comentarios, en donde vamos a poder ingresar para realizar esta operación; y por último Búsqueda que ayudará a buscar mediante palabras clave alguna información en esta plataforma.

### 3.1.2 Consultas

Nombre	Description	Profesor	Enlace
11.jpg	archivo de otro profesor	PEDRO BELLO LÓPEZ	Ver->
1967192718.jpg	mi nueva descripción	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
AlgebraRelacional.pptx	otro material pedro	PEDRO BELLO LÓPEZ	Ver->
descarga.jpg	mi nuevo mensaje	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
descarga1.jpg	mi nuevo mensaje	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
descarga2.jpg	mi nuevo mensaje	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
descarga3.jpg	mi nuevo mensaje con el mensaje3	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
MateriasAreaConocimiento.pdf	otro tipo de material	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
mustang.jpg	mi archivo 1	ALMA DELIA AMBROSIO VAZQUEZ	Ver->
winkey.txt	le falta descripción	PEDRO BELLO LÓPEZ	Ver->

Figura 3.2 Sección de Recursos

En la *Figura 3.2* vemos que como usuarios generales del sistema tenemos acceso a la información relacionada con los recursos didácticos que hayan facilitado los profesores, así como los avisos y reuniones de su área.

En el menú de la derecha (principal para el usuario) encontramos el apartado de Consultas, en donde se va a facilitar una interfaz para ingresar algún criterio de búsqueda dentro de la plataforma. La funcionalidad será la misma para la sección de Búsqueda, solo que ésta última brinda un acceso directo sin tener que entrar al menú. Adicionalmente, como trabajo a futuro, el menú de Consultas será capaz de ir añadiendo filtros de búsqueda más personalizados o con diferentes opciones para mostrar la información contenida en éste sistema, ejemplo de esto puede ser: ordenar por fecha, por nombre, más consultados (utilización de algún método de “ranking”, etc.). Veamos a continuación en la *Figura 3.3* el apartado de consultas.

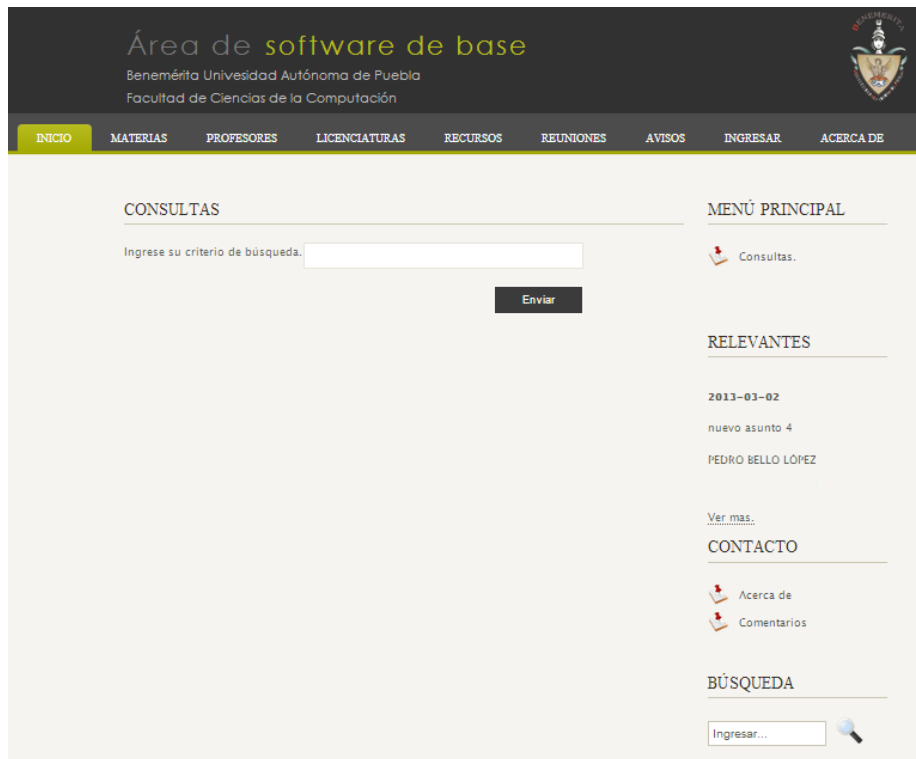


Figura 3.3 Consultas

Mostramos en la *Figura 3.3* la ventana de consultas. Una vez ingresado algún criterio (palabras clave), se mostrarán los resultados como a continuación vemos en la *Figura 3.4*.

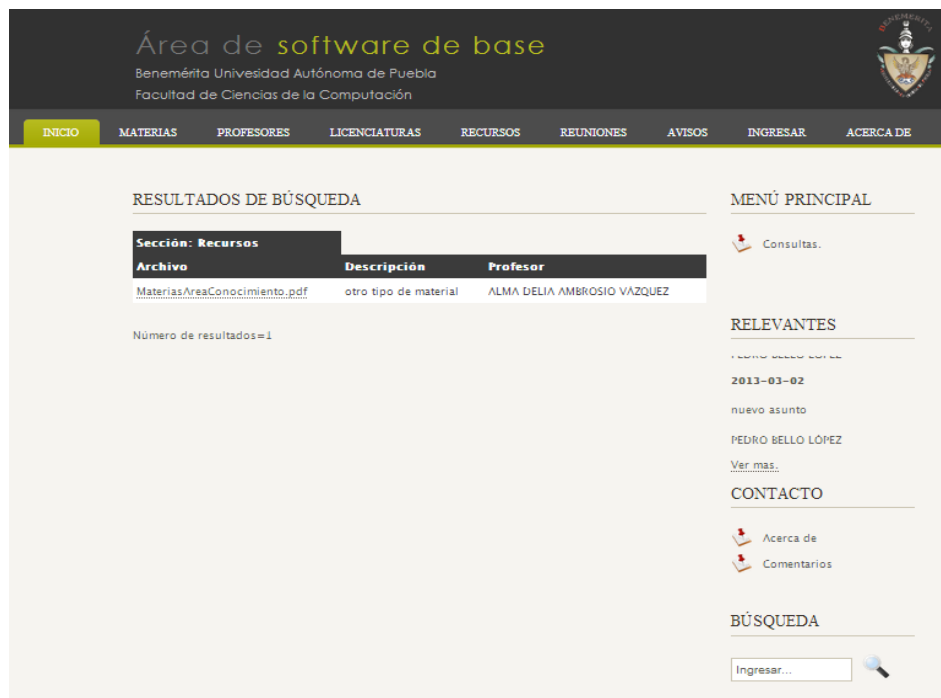


Figura 3.4 Resultados de Búsqueda

### 3.1.3 Comentarios

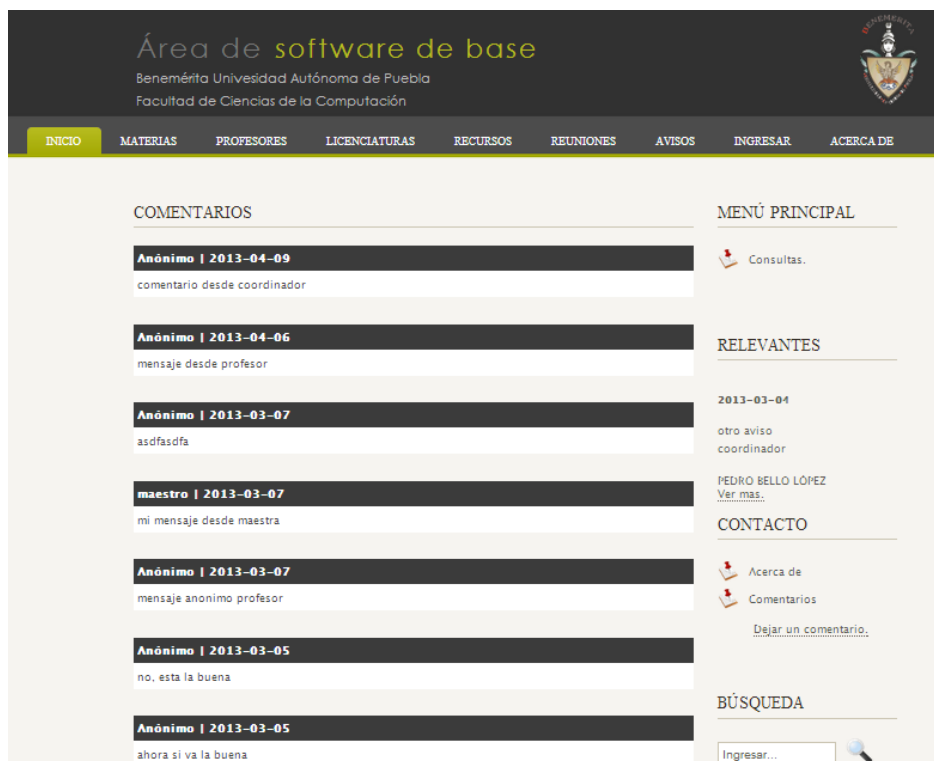


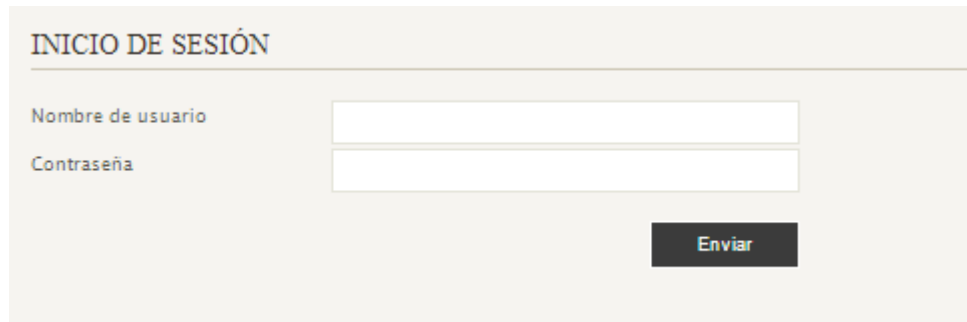
Figura 3.5 Comentarios

En la sección de comentarios (*Figura 3.5*) vemos que éstos se muestran mencionando quién realizó ese comentario (si es que el usuario ingreso ese dato, sino se imprimirá como anónimo), la fecha en que lo realizó y el mensaje. Para realizar un comentario, debemos seleccionar la opción “Dejar un comentario” de la parte derecha y mostrará un formulario como el que vemos a continuación en la *Figura 3.6*.

The image shows a form titled 'DÉJANOS UN COMENTARIO'. It contains three input fields: 'Nombre:', 'Email:', and 'Mensaje:'. Below the 'Mensaje:' field is a large text area. At the bottom left, there is a red arrow icon and the text 'Desliza para enviar'. At the bottom right, there is a black button labeled 'Enviar'. A note at the bottom left states '\*Campos obligatorios.'

Figura 3.6 Formulario para Comentarios

### 3.1.4 Validación e Ingreso al Sistema



INICIO DE SESIÓN

Nombre de usuario

Contraseña

Enviar

Figura 3.7 Inicio de Sesión

Al seleccionar la opción de Ingresar el menú superior, muestra la ventana para el inicio de sesión (*Figura 3.7*). Al ingresar los datos el sistema verificará el tipo de usuario y es como asignará el perfil correspondiente (Profesor o Coordinador).

### 3.2 Profesor

Cuando ingresamos como usuario profesor al sistema, muestra el menú de opciones que podemos ver en la *Figura 3.8*. Como primera revisión haremos el caso de Avisos; al seleccionar “Mis avisos”, de este menú mostrará las opciones para editar o eliminar avisos, a su vez mostrará los avisos que hemos realizado como se muestra en la *Figura 3.9*.

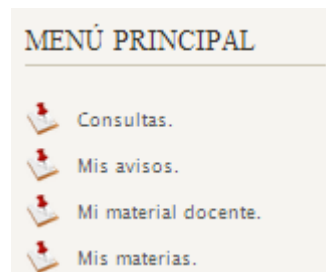


Figura 3.8 Menú Profesor

### 3.2.1 Avisos

The screenshot shows a web interface with two main sections. On the left, under the heading 'MIS AVISOS', there is a table with the following data:

Asunto	Fecha	Descripción	Profesor
aviso nuevo	2013-04-16	aviso nuevo profesor	ALMA DELIA AMBROSIO VAZQUEZ

On the right, under the heading 'MENÚ PRINCIPAL', there is a list of options: 'Consultas.', 'Mis avisos.', 'Crear un aviso', and 'Editar/Eliminar un aviso'. The last two options are underlined, suggesting they are clickable.

Figura 3.9 Mis Avisos

Para dar un aviso, tendremos que seleccionar la opción “Crear un aviso” en el menú principal, mostrará una serie de campos que ayudarán a realizarlo, podemos verlo en la *Figura 3.10*.

The screenshot shows a form titled 'CREA UN AVISO'. It contains the following fields and controls:

- Asunto:** A text input field.
- Descripción:** A large text area for entering details.
- Fecha de inicio:** A date picker showing 'dd/mm/aaaa'.
- Fecha de fin:** A date picker showing 'dd/mm/aaaa'.
- Enviar:** A dark button to submit the form.

Figura 3.10 Crear Aviso

Una vez creado el aviso, se mostrará en la sección de Avisos.

Por otra parte, “Editar/Eliminar un aviso” en el menú principal, ofrece las opciones para editar o eliminar un aviso, seleccionaremos la opción que se desea del menú desplegable, veamos inicialmente el caso para editar (*Figura 3.11*).

MIS AVISOS EDITAR/BORRAR

Seleccione el número de aviso y al final de la lista, la operación que desee realizar.

No.	Asunto	Fecha de inicio	Fecha de fin	Descripción
1	aviso nuevo	2013-04-16	2013-04-17	aviso nuevo profesor

Seleccione el número de aviso  Seleccione operación

Figura 3.11 Editar/Borrar Avisos

Como podemos observar, se muestra en forma de lista los avisos que han realizado (no todos, solo lo que ha realizado el usuario que ha ingresado con esa cuenta), muestra la información de cada uno de ellos, seleccionaremos el número de aviso que requerimos y ofrece las opciones de editar o borrar. Para editar el sistema muestra lo siguiente (Figura 3.12).

EDITANDO AVISO

Asunto:

Descripción:

Fecha de inicio:

Fecha de fin:

Figura 3.12 Editando aviso.

El sistema recupera la información del aviso en cuestión y la muestra en los campos respectivos, aquí, modificaremos la información que interesa. Cabe recordar que, como se mencionó en la sección de documentación y flujo de eventos, cada uno de estos formularios tiene campos requeridos, para éste en particular, la fecha de inicio seleccionada, no deberá ser menor a la de fin, para asegurar congruencia.

Para cuando se elige la opción de eliminar un aviso, nuevamente se recupera la información del aviso a ser afectado, y se emite un mensaje para confirmar la operación (Figura 3.13). Cuando así se realiza, el aviso es eliminado de la base de datos del sistema.



Figura 3.13 Eliminando un Aviso

### 3.2.2 Recursos Didácticos (Material Docente)

El usuario tendrá disponible la sección de “Mi material Docente” (Figura 3.14) del menú principal para visualizar, publicar, modificar, o eliminar algún material docente.

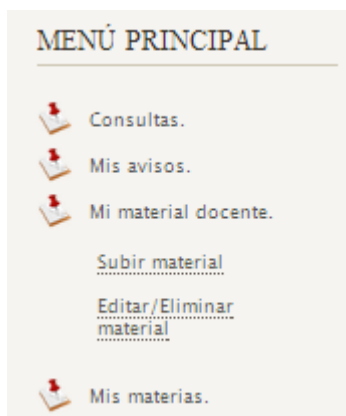


Figura 3.14 Menú Mi Material Docente

Al mostrar los elementos disponibles vemos información acerca del nombre del archivo así como su tipo (extensión), su descripción el nombre de quién lo subió, y el campo llamado “Enlace” que permite descargar o abrir el archivo en una ventana nueva.

**SUBIR MATERIAL**

Descripción:

Archivo\*:

\*Campo obligatorio

Figura 3.15 Subir Material

Al seleccionar la operación "Subir Material" muestra una ventana como vemos en la *Figura 3.15*, podemos ver los campos que necesitamos llenar para subir el recurso.

**MI MATERIAL DOCENTE EDITAR/BORRAR**

Seleccione el número de material y al final de la lista, la operación que desee realizar.

No.	Archivo	Descripción
1	1967192718.jpg	mi nueva descripción
2	descarga.jpg	mi nuevo mensaje
3	descarga1.jpg	mi nuevo mensaje
4	descarga2.jpg	mi nuevo mensaje
5	descarga3.jpg	mi nuevo mensaje con el mensaje3
6	Materias/AreaConocimiento.pdf	otro tipo de material
7	mustang.jpg	mi archivo 1

Seleccione el número de material didáctico  Seleccione operación

Figura 3.16 Editar o Eliminar Material

Como en el caso de los avisos, la opción "Editar/Eliminar material" mostrará en forma de lista la serie de recursos didácticos que el profesor haya dado de alta, así como también podremos seleccionar eliminar o editar el recurso seleccionado (*Figura 3.16*).

**EDITANDO ARCHIVO "MATERIASAREACONOCIMIENTO.PDF"**

Descripción:

Archivo\*:

\*Campo obligatorio

Figura 3.17 Editando un Recurso Didáctico

Para esta operación, editar un recurso (*Figura 3.17*), se puede cambiar la descripción del archivo, y de ser necesario, el origen del mismo, se debe tomar en cuenta que el archivo nuevo debe ser cambiado por uno diferente, ya que como se utiliza como parte de una llave de identificación (llave principal), no permitirá el reingreso de un archivo del mismo nombre.

Ahora, mostraremos el apartado para eliminar un archivo (*Figura 3.18*). De igual manera, se muestra la información acerca del archivo a eliminar y se pide la confirmación de la operación.

**CONFIRMACIÓN**

---

Realmente desea eliminar el siguiente recurso:

Archivo	Descripción
MateriasAreaConocimiento.pdf	otro tipo de material

Figura 3.18 Eliminar Recurso.

### 3.2.3 Materias

Entraremos al ámbito de las materias. Para éste se muestra el siguiente menú:

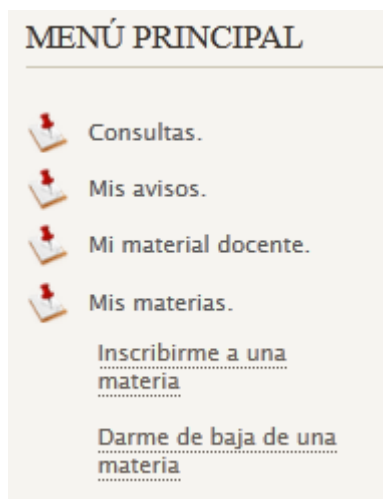


Figura 3.19 Menú Mis Materias

Podemos ver que muestran dos opciones: “Inscribirme a una materia” y “Dar de baja de una materia”. Analicemos cada una.

The image shows a page titled "INSCRIBIRME A UNA MATERIA" with a light beige background. It features a table with four columns: "No.", "CLAVE", "MATERIA", and "CARRERA". The table lists 14 courses. Below the table, there is a dropdown menu labeled "Seleccione el número de materia:" with the value "1" selected. At the bottom right, there is a dark grey button labeled "Continuar".

No.	CLAVE	MATERIA	CARRERA
1	CCO212	SISTEMAS OPERATIVOS	LCC
2	CCOM009	ENSAMBLADOR	LCC
3	CCOM014	SISTEMAS OPERATIVOS I	LCC
4	CCOM253	SISTEMAS OPERATIVOS II	LCC
5	CCOM257	COMPILADORES	LCC
6	ICCM603	SISTEMAS DE TIEMPO REAL	LCC
7	ICCM618	INTRODUCCION A LOS COMPILADORES	LCC
8	LCC106	ENSAMBLADOR	LCC
9	LCC210	PROGRAMACION DE SISTEMAS	LCC
10	LCC226	SISTEMAS OPERATIVOS CENTRALIZADOS Y DISTRIB	LCC
11	LCC320	SISTEMAS DISTRIBUIDOS	LCC
12	LCC322	COMPILADORES	LCC
13	LCC590	APLICACIÓN DE REDES Y SIST. DISTRIB	LCC
14	LIC312	SISTEMAS DE TIEMPO REAL	LCC

Figura 3.20 Inscribirse a una Materia

Seleccionando la opción “Inscribirme a una materia” podemos ver que se muestra el listado de materias que están disponibles en los diferentes planes de las carreras de la Facultad (Figura 3.20). El usuario deberá seleccionar la materia a la que desea inscribirse y después se mostrará un mensaje de confirmación en el que se mostrará la información de la materia seleccionada como se muestra en la (Figura 3.21). Para este caso, se debe tomar en cuenta de que si la materia seleccionada, ya fue elegida con anterioridad, se mostrará un mensaje que informe de esta situación, es decir, no permitirá inscribirse dos veces a la misma materia.

CONFIRMACIÓN

Está a punto de inscribirse a la siguiente materia:

Clave	Nombre	Carrera
CCO212	SISTEMAS OPERATIVOS	LCC

Inscribir

Figura 3.21 Confirmación de Inscripción

Para el caso de “Dar de baja una materia” (Figura 3.22) el sistema permite dar de baja de nuestra cuenta una materia en la cual ya estemos inscritos. Se muestra una lista de las materias ya dadas de alta en nuestra cuenta, tendremos que seleccionar una del listado y continuar con el procedimiento. De no tener ninguna, la aplicación lo informará y deshabilitará las opciones para continuar.

DAR DE BAJA UNA MATERIA

No.	CLAVE	MATERIA	CARRERA
1	LCC322	COMPILADORES	LCC
2	LCC590	APLICACIÓN DE REDES Y SIST. DISTRIB	LCC

Seleccione el número de materia: 1 ▼

Continuar

Figura 3.22 Dar de Baja una Materia

Cuando se continúa con la baja de alguna materia, se muestra un mensaje de confirmación para advertir el procedimiento en curso y ratificar la operación (Figura 3.23).

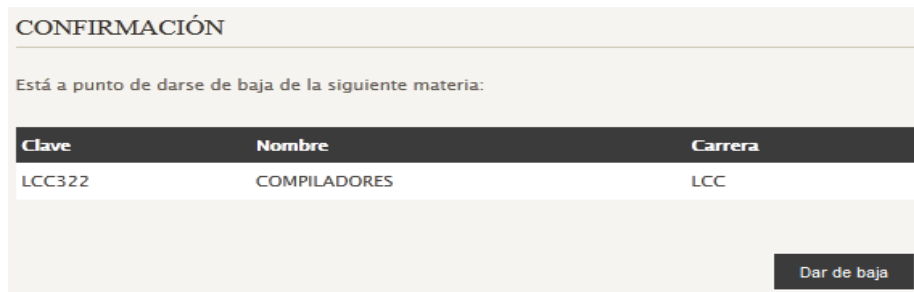


Figura 3.23 Confirmación de Baja de Materia

### 3.3 Coordinador

Para el usuario con perfil de coordinador, tenemos el siguiente menú:



Figura 3.24 Menú Coordinador

Vamos a resaltar el hecho de que las funciones descritas en el perfil de alumno y profesor, son las mismas para el perfil de coordinador (Consultas, Comentarios, Avisos, Materias, Material Docente), con las mismas partes y procedimientos. Adicionalmente a estas, se describirán a continuación las secciones de “Actas de reuniones” y “Profesores”.

#### 3.3.1 Actas de Reuniones

Al seleccionar esta opción se muestran en pantalla los documentos de las reuniones que el coordinador del área ha subido. Se puede ver en la *Figura 3.25* que se describe información acerca de la fecha y el archivo del acta, este a su vez, al seleccionarlo, abrirá una ventana nueva mostrándonos el documento.

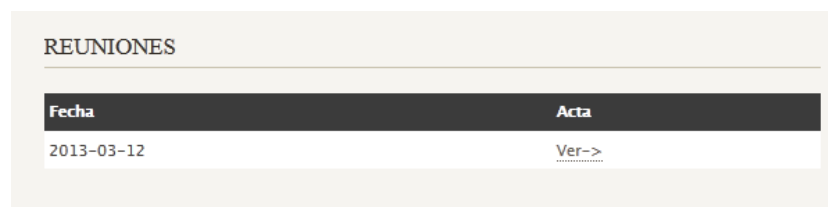


Figura 3.25 Actas de Reuniones

En el menú principal se muestran dos operaciones: “Subir acta” y “Eliminar acta” (Figura 3.26), cada una de las cuales se van a detallar a continuación.

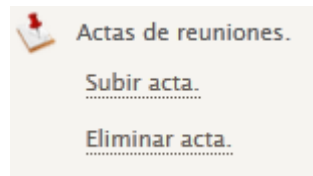


Figura 3.26 Menú Actas de Reuniones

Cuando elegimos la operación “Subir acta”, muestra nuevamente un formulario con los campos necesarios para subir el documento (Figura 3.27). Al llenar los campos y aviándolos, el sistema se ocupa de validarlos, de ser correctos, muestra un mensaje en donde menciona que la operación fue realizada, si no, indica los campos inválidos o incorrectos.

A screenshot of a web form titled "REUNIONES SUBIR ACTA". It contains two input fields: "Fecha:" with a placeholder "YYYY/MM/DD" and "Archivo:" with a file upload icon and a button labeled "Examinar...". At the bottom right, there is a dark button labeled "Enviar".

Figura 3.27 Subir Acta

En el caso de eliminar un acta, se muestra el listado de las actas que han sido dadas de alta en el sistema (Figura 3.28), se deberá seleccionar el acta que se desea eliminar y continuar el procedimiento. Una vez seleccionada el acta muestra la información sobre ésta y pide la confirmación para proceder a la eliminación de la misma.

### 3.3.2 Profesores

Para el usuario con perfil de coordinador, se le darán opciones para el alta, modificación y eliminación de información de los profesores que conforma el área de software de base. En la Figura 3.28 se muestra el menú principal de este procedimiento.

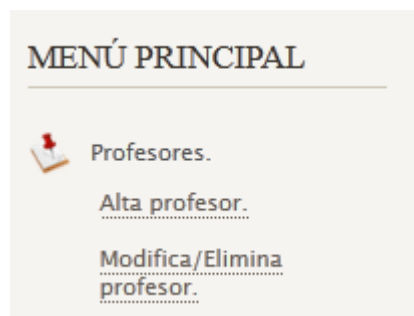


Figura 3.28 Menú Profesores

Cuando el coordinador del área requiera dar de alta a un nuevo profesor, se mostrará un formulario con los campos necesarios para el perfil del mismo. A continuación se muestra el listado de los mismos.



ALTA PROFESOR

Nombre\*:

Apellidos\*:

División\*: PRIMARIA ▼

Rol\*: PROFESOR ▼

Nombre de usuario\*:

Contraseña\*:

Repita la contraseña\*:

Sitio web:

\*Campos obligatorios.

Enviar

Figura 3.27 Alta Profesor

Al enviar el formulario, el sistema validará los campos. Mostrará mensaje de advertencia en campos obligatorios, validación de campos ya existentes (Nombre y/o Nombre de usuario) y el ingreso de los campos Contraseña y Repita la contraseña. Una vez que las validaciones sean aceptadas, el nuevo profesor será dado de alta en el sistema.

Para las opción “Modifica/Elimina profesor” en ambos casos, se muestra el listado de los profesores dados de alta en el sistema. Se deberá seleccionar el número de profesor que se desea afectar y elegir la operación deseada (Figura 3.28). Veamos los casos particulares.

PROFESORES EDITAR/ELIMINAR				
Seleccione el número de profesor y al final de la lista, la operación que desee realizar.				
No.	NOMBRE	APELLIDOS	DIVISION	WEB
1	JOSE MARTIN	ORATO RAMIREZ	PRIMARIA	Ir
2	HILARIO	SALAZAR MARTINEZ	PRIMARIA	
3	LETICIA	MENDOZA ALONSO	PRIMARIA	
4	MARCOS	GONZÁLEZ FLORES	PRIMARIA	
5	LAURA	CUAYAHUITL ROMERO	PRIMARIA	
6	LUIS ENRIQUE	COLMENARES GUILLEN	PRIMARIA	
7	HILDA	CASTILLO ZACATELCO	PRIMARIA	
8	ALMA DELIA	AMBROSIO VÁZQUEZ	PRIMARIA	
9	MIGUEL ANGEL	LEON CHAVEZ	SECUNDARIA	
10	YOLANDA	MOYAO MARTINEZ	SECUNDARIA	

Figura 3.28 Editar/Eliminar Profesores

En el caso cuya opción seleccionada haya sido la de editar a un profesor, se recuperará la información del profesor que haya sido elegido (*Figura 3.29*), autocompletando los campos de su información. El usuario podrá modificar cualquiera de ellos. Una vez cumplido esto, el sistema nuevamente validará que los campos sean correctos y de ser así la información será actualizada.

EDITANDO PROFESOR	
Nombre*:	<input type="text" value="JOSE MARTIN"/>
Apellidos*:	<input type="text" value="ORATO RAMIREZ"/>
División*:	<input type="text" value="PRIMARIA"/>
Rol*:	<input type="text" value="PROFESOR"/>
Nombre de usuario*:	<input type="text"/>
Contraseña*:	<input type="password" value="••••••"/>
Repita la contraseña*:	<input type="password" value="••••••"/>
Sitio web:	<input type="text" value="www.cs.buap.mx"/>
*Campos obligatorios.	
<input type="button" value="Enviar"/>	

Figura 3.29 Editando Profesor

Por otro lado, para el caso de dar de baja a un profesor, el sistema nuevamente recuperará la información del profesor a modificar en pantalla (*Figura 3.30*). En este caso, se enviará un mensaje de alerta al usuario, pidiendo confirmar la operación, de continuar, el perfil del profesor será eliminado de la base de datos.

**CONFIRMACIÓN**

---

Realmente desea eliminar al siguiente profesor de la base de datos:

Nombre	Apellidos	División	Rol	web
JOSE MARTIN	ORATO RAMIREZ	PRIMARIA	PROFESOR	Ir

[Confirmar](#)

Figura 3.30 Eliminando Profesor

## CONCLUSIÓN Y TRABAJO A FUTURO

A nivel general se reconoce la gran ventaja de la Ingeniería de Software como proceso estructural y organizado para el diseño de aplicaciones ya que nos permite una forma de llevar a cabo los objetivos que se tienen en el momento del desarrollo de un sistema, que va desde recabar la información necesaria por parte del cliente o interesado en dicho sistema hasta su implementación, desarrollo y mantenimiento.

Dentro de algunas de las ventajas que ofrece el tener este tipo de procedimientos, es la de corregir errores de manera oportuna y sistemática, dimensionar las posibilidades del sistema, mejorar el rendimiento de los recursos disponibles (teniendo bases teóricas como al momento de diseñar nuestra base de datos), dar un interfaz más amigable y fácil de usar al interesado, ofrecer alternativas de solución en materia de seguridad y confiabilidad, y tener una mejor forma de ofrecer un producto final de calidad y productividad relativo a su diseño y uso. Todo esto sin dejar en ningún momento de lado el objetivo de satisfacer las necesidades de nuestro contratista haciendo que el sistema cumpla las funciones para lo cual fue requerido.

En lo particular, al desarrollar el presente sistema, se ha logrado obtener un medio más eficaz de comunicación entre la comunidad estudiantil de la Facultad de Ciencias de la Computación, específicamente en el área de software de base, facilitando un medio más adecuado y concerniente al estudio científico desarrollado en esta institución.

Tanto profesores como alumnos tendrán la posibilidad de extender sus conocimientos y habilidades más allá del aula de clases, haciendo uso de un medio tan importante hoy en día como lo es el internet mediante este sistema web.

Para los mismos integrantes de esta área, les será posible el intercambio de información, teniendo disponible en todo momento conclusiones importantes sustentadas en las actas de reuniones. Así como también la posibilidad de tener un medio organizado e interactivo para el uso de su material docente.

Los alumnos serán quizá la población más beneficiada, pues la disponibilidad de apuntes, programas, presentaciones, documentos y demás recursos didácticos, estará presente en todo momento.

Como a trabajo a futuro se plantean los siguientes puntos:

- Hacer un método de búsqueda con más herramientas como el uso de filtros más especializados.
- Tener un mejor analizador léxico en el momento de revisar el contenido de los comentarios haciendo uso de teorías como la del procesamiento de lenguaje natural.
- La extensión de este sistema para las demás áreas que conforman a la facultad.

## BIBLIOGRAFÍA

- [1] Pastor Joan Antoni, "CONCEPTO DE SISTEMA DE INFORMACIÓN EN LA ORGANIZACIÓN". Editorial UOC, Barcelona España 2002.
- [2] J. Braude, Eric. "INGENIERIA DE SOFTWARE, una perspectiva orientada a objetos", Primera edición. México Febrero 2003.
- [3] Pressman, Roger S. "INGENIERIA DE SOFTWARE, un enfoque práctico". Quinta Edición, España 2002.
- [4] Jacobson, Booch, Rumbaugh. "EL LENGUAJE UNIFICADO DE MODELADO. Manual de Referencia". Pearson Education Madrid 2002.
- [5] Jacobson, Booch, Rumbaugh., "EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE". Addison Wesley. Segunda Edición. Madrid 2000.
- [6] R. Elmasri, S.B. Navathe, "SISTEMAS DE BASES DE DATOS. Conceptos fundamentales". Quinta Edición, Addison-Wesley 2007.
- [7] Ullman, Jeffrey D. Widom, Jennifer. "INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS". Primera edición. México 1999.
- [8] Silberschatz, Abraham. Korth, Henry F. Sudarshan, S. "FUNDAMENTOS DE BASES DE DATOS". Cuarta Edición, Madrid 2002.
- [9] Gutiérrez Rodríguez, Abraham; Bravo García, Ginés. "PHP A TRAVÉS DE EJEMPLOS" Primera Edición Ra-Ma 2004.
- [10] Gilfillan, Ian. "LA BIBLIA MySQL". Anaya Multimedia 2003.
- [11] Kabir, Mohammed J. "SERVIDOR APACHE 2 LA BIBLIA". Anaya multimedia Madrid 2002.
- [12] Pascual González, Francisco. "Macromedia Dreamweaver 8", Alfaomega, México 2006.
- [13] <http://www.visual-paradigm.com/>
- [14] <http://www.wampserver.com/en/>
- [15] <http://php.net/manual/es/index.php>
- [16] <http://dev.mysql.com/doc/refman/5.0/es/index.html>